

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA‘LIM
VAZIRLIGI**

NAVOIY DAVLAT PEDAGOGIKA INSTITUTI

FIZIKA - MATEMATIKA FAKULTETI

"INFORMATIKA O‘QITISH METODIKASI" KAFEDRASI

ALGORITMLAR FANIDAN

O‘QUV-USLUBIY MAJMUA

BILIM	100000 - Gumanitar
TA‘LIM SOHASI	110000 - Pedagogika
TA‘LIM YO‘NALISHI	5110700 - Informatika o‘qitish metodikasi

Navoiy - 2019

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligi tomonidan 201__ yil _____ tasdiqlangan o‘quv rejasi va O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligining va 201____-yil _____-avgustdagi _____-sonli buyrug‘i bilan fan dasturi tasdiqlangan va Oliy va o‘rta maxsus, kasb-hunar ta’limi yo‘nalishlari bo‘yicha O‘quv-uslubiy birlashmalar faoliyatini Muvofiqlashtiruvchi Kengashning 201__-yil _____-avgustdagi ____-sonli bayonnomasi bilan ma’qullangan. “Algoritmlar” fanining namunaviy fan dasturi va namunaviy o‘quv rejaga muvofiq ushbu o‘quv uslubiy majmua ishlab chiqildi.

Tuzuvchilar:

Ibragimov Alimjan Artikbayevich	- Navoiy davlat pedagogika instituti «Informatika o‘qitish metodikasi» kafedrasida dotsenti, fizika-matematika fanlari nomzodi;
Xamroyeva Dilafro‘z Namozovna	- Navoiy davlat pedagogika instituti «Informatika o‘qitish metodikasi» kafedrasida katta o‘qituvchisi.
Absobirov Samariddin Qurbonovich	Navoiy davlat pedagogika instituti «Informatika o‘qitish metodikasi» kafedrasida o‘qituvchisi.

Taqrizchilar:

Yodgorov G‘.R.	- NavDPI “Informatika o‘qitish metodikasi” kafedrasida mudiri, fizika-matematika fanlari nomzodi;
Ro‘ziyev R.A.	- NavDPI “Informatika o‘qitish metodikasi” kafedrasida dotsenti, fizika-matematika fanlari nomzodi.

Fanning o‘quv uslubiy majmuasi “Informatika o‘qitish metodikasi” kafedrasining 2019-yil _____-avgustdagi 1-son yig‘ilishidan muhokamadan o‘tgan va Fakultet kengashida muhokama qilish uchun tavsiya etilgan.

Kafedra mudiri:_____ f.-m.f.n.Yodgorov G‘.R.

Fanning o‘quv uslubiy majmuasi Fizika–matematika fakultet kengashida muhokama etilgan va foydalanishga tavsiya qilingan (2019-yil __avgustdagi 1-sonli bayonnoma).

Fakultet kengashi raisi:_____ dots.Kamolov I.R.

Navoiy davlat pedagogika institutining 2019-yil _____avgustdagi 1-sonli Ilmiy uslubiy kengashida muhokama qilinib tasdiqlangan.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

MUNDARIJA

I. KIRISH.....	4
II. O‘QUV MATERIALLARI.....	5
2.1.Ma’ruza matn.....	6
2.2.Amaliy mashg‘ulot ishlanmalari.....	121
2.3.Laboratoriya mashg‘ulot ishlanmalari.....	201
III. MUSTAQIL TA’LIM MAVZULARI.....	268
IV.GLOSSARIY.....	271
V. ILOVALAR.....	281
5.1.Fan dasturi.....	282
5.2.Ishchi o‘quv dasturi.....	292
5.3.Tarqatma material.....	317
5.4.Test.....	319
5.5. Nazorat topshiriqlari.....	324
5.6. Baholash mezoni.....	331
VI. FOYDALANILGAN ADABIYOTLAR.....	339

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

KIRISH

Ta’lim sohasidagi tub islohotlarning asosiy maqsadi jahon andozalari asosida bilimlar berish va raqobatdosh kadrlar tayyorlashdir. Shuning uchun ta’lim tizimidagi 5110700 – Informatika o‘qitish metodikasi yo‘nalishida o‘qitiladigan fanlar ham zamonaviy fanlardan hisoblanadi. Ushbu namunaviy dastur bugungi kunning zamonaviy bilimlari bilan yangilangan va qayta ishlangan dastur bo‘lib, unda fanning nazariy va amaliy jihatlariga alohida e’tibor qaratilgan. Mazkur fan dasturi bakalavriat yo‘nalishi: 5110700 – Informatika o‘qitish metodikasi ta’lim yo‘nalishida tahsil olayotgan talabalarning o‘zlashtirishi lozim bo‘lgan bilimlari va unga qo‘yiladigan talablar asosida tuzilgan bo‘lib, bo‘lajak fan o‘qituvchisi egallashi kerak bo‘lgan bilimlar va ko‘nikmalar majmuini o‘z ichiga oladi:

- algoritmik tillar, algoritm va uning xossalari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari;

- steklar, navbatlar, daraxtlar, algoritmlarning tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, saralashning Sheyker, Shella usullari, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, kattaliklar va ularning tiplari, buruqlar tizimi va operatorlari, massivlar, chiziqli, tarmoqlanuvchi va takrorlanuvchi operatorlar, funksiya va protseduralar, fayllar bilan ishlash, tilning grafik imkoniyatlari;

“Algoritmlar” fani insonlarda axborot muhitida ma’lum bir dunyoqarashni shakllantirishga xizmat qilishi bilan bir qatorda, uning axboriy madaniyatni egallashida asosiy rol o‘ynaydi. Bugungi “Axborot” asrida yoshlarning kompyuter savodxonligini oshiribgina qolmay, balki ma’lumotlar bazalari bilan ishlash imkoniyatlarini oshiradi. Umumiy o‘rta ta’lim maktablari, akademik litsey va kasb–hunar kollejlarda “Informatika” mutahassislaridagi fanlarni o‘qitish uchun kadrlarni tayyorlab beradi.

Fanni o‘qitishdan maqsad - bo‘lajak informatika o‘qituvchisining kasbiy sohasida egallashi lozim bo‘lgan bilimlar va amalda qo‘llash uchun ko‘nikma va malakalarni shakllantirish va rivojlantirishdan iborat. Ushbu dasturda har bir kasb egasi uning faoliyat ko‘rsatish turidan qat’iy nazar egallashi kerak bo‘lgan tayanch nazariy va amaliy ma’lumotlarni o‘z ichiga oladi.

Fanning o‘qitishning vazifasi - algoritm tushunchasi va uning xossalari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar bilan tanishtirish;

- algoritmik tillarning asosiy tushunchalari: steklar, navbatlar, daraxtlar, algoritmlar tahlili kabilar haqida ma’lumotlar berish.

- saralash algoritmlari va ulardan foydalanish usullari haqida tasavvurlar hosil qilish borasidagi bilimlar va amalda qo‘llash uchun ko‘nikma va makalalarni shakllantirish va rivojlantirishdan iborat.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

II. O‘QUV MATERIALLARI

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA‘LIM
VAZIRLIGI**

Navoiy davlat pedagogika instituti

“Informatika o‘qitish metodikasi” kafedrası

ALGORITMLAR FANIDAN

MA‘RUZA MATNLARI

Navoiy-2019

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Ushbu ma’ruza matnlari to‘plami Informatika o‘qitish metodikasi yo‘nalishidagi bakalavrlar uchun mo‘ljallangan “Algoritmlar” fani bo‘yicha fan dastur asosida tuzilgan bo‘lib, bo‘lajak mutaxassis egallashi kerak bo‘lgan bilim va ko‘nikmalarni o‘z ichiga oladi.

Tuzuvchilar: dots.A.A.Ibragimov
k.o‘q.D.N.Xamroyeva
o‘q. S.Q.Absobirov

Taqrizchi: dots.Ro‘ziyev R.A.

Ma’ruza matnlari to‘plami Navoiy davlat pedagogika instituti “Informatika o‘qitish metodikasi” kafedrası yig‘ilishida ko‘rib chiqilgan va tavsiya qilingan. 2019-yil____ - avgustdagi 1-sonli yig‘ilish bayonnomasi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

1-Modul. Algoritm tushunchasi va mohiyati.

MA’RUZA №1

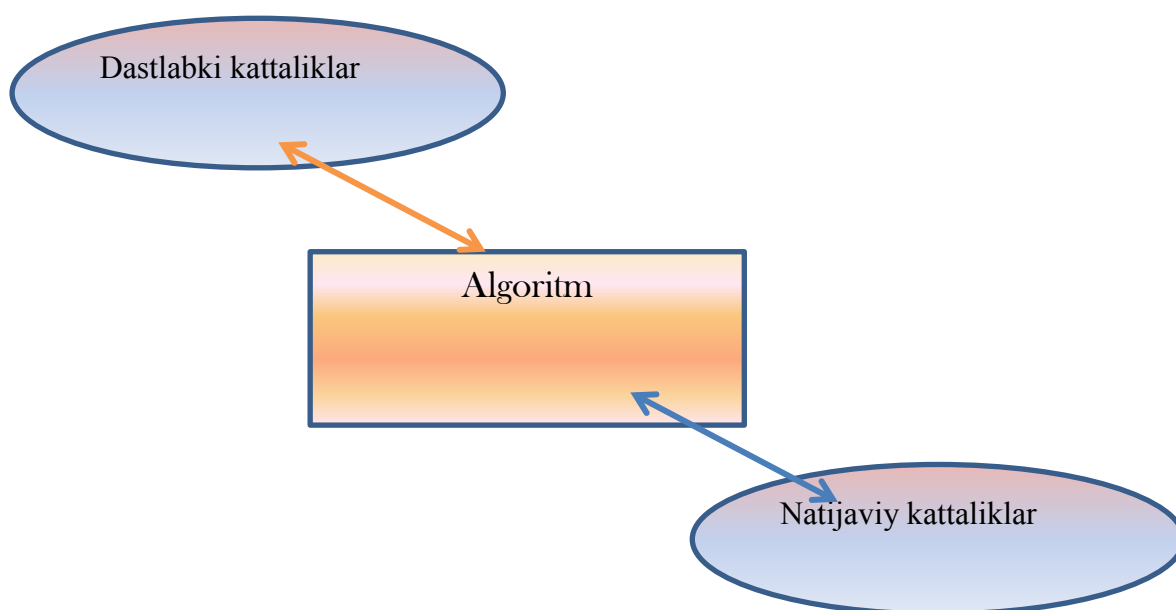
MAVZU: ALGORITM TUSHUNCHASI VA ULARDAN FOYDALANISH.

REJA:

1. Algoritm tushunchasi. Algoritm xossalari.
2. Algoritm turlari va ularni tasvirlash usullari.
3. Algoritmik tillar.

Tayanch so‘z va iboralar: algoritm, xossa, tasvirlash usullari, blok-sxema, dastur.

Algoritm bu aniq hisoblashlarni bajaruvchi protsedura bo‘lib unga kirish qismida kattalik yoki kattaliklar berilib chiqishda natijaviy kattalik yoki kattaliklar olinadi. Demak algoritm hisoblovchi qadamlardan tashkil topgan bo‘lib dastlabki qiymatlarga ko‘ra natijaviy kattaliklar qiymatini beradi. Bu holatni sxematik tarzda quyidagicha tasvirlash mumkin.



Algoritmni qo‘yilgan hisoblash masalani (computational problem) aniq bajaruvchi uskuna sifatida ham qaralishi mumkin. Algoritmarda keltirilgan protseduralar yordamida kattaliklar bilan amallar bajarilib natijalar olinadi. Masalan, biror sonlar ketma-ketligini orta boorish tartibida saralash. Saralash masalasi (sorting problem) ga misol keltiramiz:

Kirish: n -ta sondan iborat sonlar ketma-ketligi (a_1, a_2, \dots, a_N) .

Chiqish: n -ta sondan iborat sonlar ketma-ketligi $(b_1 \leq b_2 \leq \dots \leq b_N)$.

Misol, $(31, 41, 59, 26, 41, 56)$ kiruvchi ketma-ketlik bo‘lsa, chiquvchi ketma-ketlik $(26, 31, 41, 41, 56, 59)$ bo‘lishi lozim. Bunga o‘xshash kiruvchi ketma-ketlik saralash ekzemplari (instance) deb yuritiladi. Agar algoritm har qanday kiruvchi qiymatlar uchun aniq va mos chiquvchi qiymatlarni bera olsa u aniq (correct) deb yuritiladi. Algorimlardan amaliyotda foydalanishga ayrim misollarni keltiramiz:

- Odam DNK si tarkibidagi 100 ming gen identifikatsiyasi, DNK-ni tashkil etuvchi 3 milliard asosiy juftlikni saralashva tahlili masalasi;
- Internetda ma’lumotlar olish masalasi: kata hajmdagi ma’lumotlarni olish, jo‘natish, qidiruv va optimal marshrut tanlash;
- Electron kommertsiya masalalarida (kredit karta nomerlari , parollar, bank xisob-kitob raqamlari himoyasi, raqamli imzo va b);

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Algoritmlarni ishlab chiqishda masalani yechimi uchun zarur bo‘lgan vaqt va xotira hajmi muhim ko‘rsatgichlar hisoblanib algoritmlarni yaratishda ularni samarali foydalanishni hisobga olish zarur. Aynan bir masalani yechish uchun turli algoritmlar tuzilishi mumkin. Ular bir-biridan samardorlik darajasi bilan farqlanadilar. Bu farq turli texnik va dasturiy ta‘minotlarda har xil bo‘lishi mumkin.

Misol uchun ikkita saralash algoritmlari farqini ko‘rib chiqamiz:

Saralash algoritmi	Sarflanadigan vaqt	Izoh
Joylashtirish usuli	$C_1 n^2$ bu N^2 -ga proporsional	C_1 -n ga bog‘liq bo‘lmagan doimiylik n-saralanadigan elementlar soni
Qo‘shish usuli	$C_2 n \lg n$	$\lg n = \log_2 n$, C_2 -n ga bog‘liq bo‘lmagan doimiylik

Qo‘shish usuli joylashtirish usulidan samaraliroq ekanligini quyida keltirilgan jadval ma‘lumotlarini tahlili orqali keltiramiz.

komputerlar	Saralanadigan sonlar soni	Saralovchi algoritm	Talab qilinadigan vaqt
A(tez ishlovchi 1sekundda 10mlrd amal bajaradi)	10 mlnta(taqruban 80 mb)	Joylashtirish usuli (tajribali dasturchi tomonidan yaratilgan algoritm saralash uchun $2n^2$ amal bajariladi)	$\frac{2 * (10^7)^2 \text{ buyruqlar}}{10^{10} \text{ buyruq/sec}}$ $= 20000 \text{ sec}$ <i>(5,5 soatdan ko‘proq)</i>
B(sekin ishlovchi 1sekundda 10 mln amal bajaradi)		Qo‘shish usuli (o‘rta darajali dasturchi tomonidan yaratilgan algoritm saralash uchun $50n \lg n$ amal bajariladi))	$\frac{50 * 10^7 \lg 10^7}{10^7} \approx 1163 \text{ sekund}$ <i>(20 min dan kam)</i>

Umuman olganda algoritm - bu quyilgan masalaning echimiga olib keladigan, ma‘lum qoidaga binoan bajariladigan amallarning chekli qadamlar ketma-ketligidir. Boshqacha qilib aytganda algoritm boshlang‘ish ma‘lumotlardan natijagacha olib keluvchi jarayonning aniq yozilishidir.

Algoritm tushunshasining turli ta‘riflari bir qator talablarga javob berishi kerak:

- algoritm chekli sondagi elementar bajariluvshi ko‘rsatmalardan iborat bo‘lishi kerak;
- algoritm chekli sondagi qadamlardan iborat bo‘lishi kerak;
- algoritm barsha boshlang‘ish berilganlar ush un umumiy bo‘lishi kerak;
- algoritm to‘g‘ri echimga olib kelishi kerak.

Har qanday algoritm ma‘lum ko‘rsatmalarga binoan bajariladi va bu ko‘rsatmalarga buyruq deyiladi. Yuqoridagi fikrga ko‘ra algoritm asosan masalani eshimini toppish ush un tuziladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Bitta masalani eshishning bir neshaalgoritmi mavjud bo‘lishi mumkin. Ular orasida eng samaralisini, bajarilishi ushuncha kam amallar, mashina vaqti, xotira va h.k.ni talab qiluvchi algoritmlarni tanlash lozim. Samarali algoritmlar mavjud bo‘lishi shartlari va ularni qurish (ishlab shiqish)ni o‘rganish algoritmlar nazariyasi asosini tashkil etadi.

Algoritm kibernetika va matematikaning asosiy tushunchalaridan biri bo‘lib bu atama o‘rta asrlarda yashab ijod etgan buyuk o‘zbek matematigi Al-Xorazmiy nomidan kelib shiqqan. U IX asrning 825 yilidayoq o‘zi kashf etgan o‘nli sanoq tizimida to‘rt arifmetikaamallarini bajarish qoidalarini bergan. Arifmetikaamallarini bajarish jarayoni esaalxorazm deb atalgan. Bu atama 1747 yildan boshlab algorismus, 1950 yilga kelib algoritm deb ham ataldi. Fanda "Yevklid algoritmi", "G‘iyosiddin Koshiy algoritmi", "Laure algoritmi", "Markov algoritmi" deb ataluvshi algoritmlar ma‘lum algoritm tushunchasi tobora kengayib borib, kibernetikaning nazariy va mantiqiy asosi hisoblangan algoritmlar nazariyasi paydo bo‘lgan. Kompyuterlar paydo bo‘lishi bilan algoritm atamasi hozirgi ma‘nosi bilan axborot texnologiyalari sohasida eng asosiy atamalardan biri bo‘lib qoldi. Odatda algoritmlar u yoki bu hisoblashga doir masalalarni (computational problems) eshish ushuncha tuziladi.

Qo‘yilgan masala ushuncha yaratiladigan algoritmda kiruvshi va shiquvshi ma‘lumotlar muhim ahamiyatga ega, agar algoritm to‘g‘ri tuzilgan bo‘lsa, ijroshi (kompyuter) aniq natijalar beradi.

Algoritm quyidagi xossalarga ega: aniqlik, tushunarlilik, ommaviylik, natijaviylik va diskretlik.

Aniqlik va tushunarlilik - degandaalgoritmda ijroshiga berilayotgan ko‘rsatmalar aniq mazmunda bo‘lishi tushuniladi. Shunki ko‘rsatmalardagi noaniqliklar mo‘ljallangan maqsadga erishishgaolimb kelmaydi. Ijroshiga tavsiya etiladigan ko‘rsatmalar tushunarli mazmunda bo‘lishi shart, aks holda ijroshi uni bajaraolmaydi.

Ommaviylik - deganda har bir algoritm mazmuniga ko‘ra bir turdagi masalalarning barshasi ushuncha ham o‘rinli bo‘lishi, ya‘ni umumiy bo‘lishi tushuniladi.

Natijaviylik - deganda algoritmda chekli qadamlardan so‘ng albatta natija bo‘lishi tushuniladi. Shuni ta‘kidlash joizki, algoritm avvalfdan ko‘zlangan maqsadga erishishga olib kelmasligi ham mumkin. Bunga ba‘zan algoritmnining noto‘g‘ri tuzilgani yoki boshqa xatolik sabab bo‘lishi mumkin, ikkinshi tomondan, qo‘yilgan masala ijodiy yeshimga ega bo‘lmasligi ham mumkin. Lekin salbiy natija ham deb qabul qilinadi.

Diskretlik - deganda algoritmlarni chekli qadamlardan tashkil qilib bo‘laklash imkoniyati tushuniladi.

Algoritmga doir quyidagi masalalarni misol sifatida keltirish mumkin:

- Talabani kundalik ishlarni tashkil etish;
- To‘rtburshak perimetri va yuzasini hisoblash;
- R radiusli doirani yuzasini va aylana uzunligini topish;
- $A_1, A_2, A_3, \dots, A_n$ sonlarni toq elementlarini yig‘indisini topish;
- Berilgan ketma-ketlik sonlarni o‘shish (kamayish) tartibda joylashtirish va h.k.

Algoritmnining ushta turi mavjud: shiziqli, tarmoqlanuvshi va takrorlanuvshi (tsiklik).

SHiziqli algoritmlar - hesh qanday shartsiz faqat ketma-ket bajariladigan jarayonlardir.

Tarmoqlanuvshi algoritmlar - ma‘lum shartlarga muvofiq bajariladigan jarayonlardir.

Takrorlanuvshi algoritmlar - biron bir shart tekshirilishi yoki biron parametrning har xil qiymatlari asosidachekli ravishda takrorlanish yuz beradigan jarayonlardir.

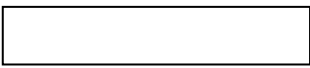
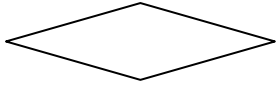

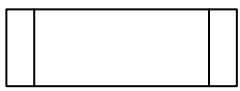
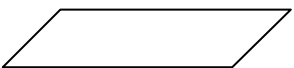
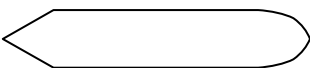
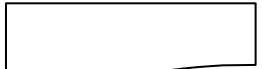


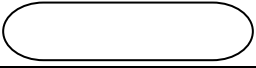
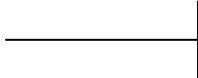
Algoritmni turli usullarda tasvirlash mumkin.

- so‘z bilan ifodalash;
- formulalarda berish;
- blok-sxemalarda tasvirlash;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

- dastur shaklida ifodalash va boshqalar.

Algoritmni blok-sxema ko‘rinishda tasvirlash qulay va tushunarli bo‘lgani ushuncha ko‘p ishlatiladi. Bunda algoritmdagi har bir ko‘rsatma o‘z shakliga ega. Masalan: parallelogramm ko‘rinishdagi belgi ma’lumotlarni kiritish va shiqarish; to‘g‘ri to‘rtburshak belgisi hisoblash jarayonini; romb belgisi shartlarning tekshirilishini bildiradi. Algoritmni blok-sxema shaklida tasvirlashda quyidagi geometrik figuralardan foydalaniladi:

Nomi	Belgilanishi	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarni bajarilishi natijasida ma’lumotlarning uzgarishi
Karor		Biror shartga bog‘liq ravishda algoritmning bajarilish yunalishini tanlash
SHakl uzgartirish		Dasturni uzgartiruvchi buyruk yoki buyruklar turkumini uzgartirish amalini bajarish
Avval aniklangan jarayon		Oldindan ishlab chikilgan dastur yoki algoritmdan foydalanish
Kiritish CHikarish		Axborotlarni kayta ishlash mumkin bulgan shaklga utkazish yoki olingan natijani tasvirlash
Displey		EXMga ulangan displeydan axborotlarni kiritish yoki chikarish
Xujjat		Axborotlarni kogoza chikarish yoki kogoздan kiritish
Axborotlar okimi chizigi		Bloklar orasidagi boglanishlarni tasvirlash
Boglagich		Uzilib kolgan axborot okimlarini ulash belgisi
Boshlash Tugatish		Axborotni kayta ishlashni boshlash, vaktincha yoki butunlay tuxtatish
Izoh		Bloklarga tegishli turli xildagi tushuntirishlar

Algoritmni tasvirlash usullariga misollar keltirib o‘tamiz:

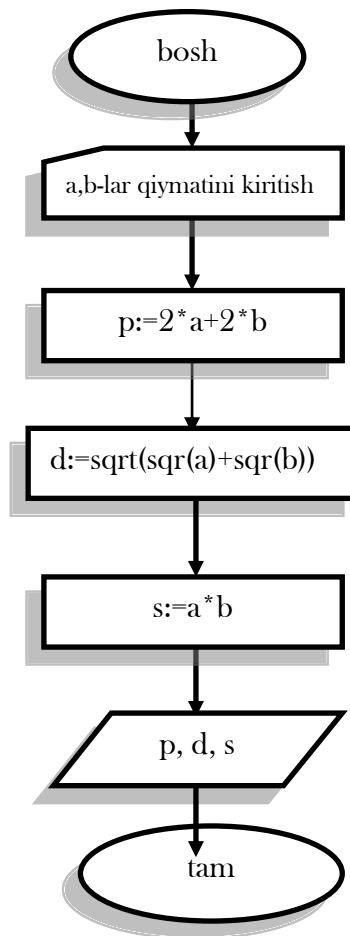
Masala: to‘g‘ri to‘rtburshakning tomonlariga ko‘ra uning perimetri, diagonali va yuzasini hisoblash.

1. So‘z bilan ifodalash:

- 1.1. boshlash;
- 1.2. tomonlar qiymatini kiritish (a, b);

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

- 1.3. perimetr qiymatini hisoblash (p);
 - 1.4. diagonal qiymatini hisoblash (d);
 - 1.5. yuzasini hisoblash (s);
 - 1.6. perimetr, diagonal va yuzasini qiymatini shop etish.
2. Formulalarda berish:
- 2.1. A va B to‘rtburshak tomonlari qiymatlari;
 - 2.2. $P=2*a+2*b$;
 - 2.3. $D = \sqrt{a^2 + b^2}$;
 - 2.4. $S=a*b$;
 - 2.5. P, D va S qiymatlarini shop etish
3. Blok-sxemalarda tasvirlash:



4. Dastur shaklida ifodalash: (Pascal dasturlash tili misolida)

```
Program to‘rtburshak yuzi;  
Var a, b: Integer;  
P, d, s: real;  
Begin  
Write ('a,b tomonlarni qiymatlari kiritilsin');  
ReadLn(a,b);  
P:=2*a+2*b;  
D:=sqrt(sqr(a)+sqr(b));  
S:=a*b;  
WriteLn('to‘rtburshak perimetri= ',p);  
WriteLn('to‘rtburshak dioganperli= ',d);  
WriteLn('to‘rtburshak yuzasi= ',S);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

End.

Hozirgi kunda juda ko‘p algoritmik tillar mavjud bo‘lib, ularni dasturlash tillari deb ataymiz. Algoritmik til - algoritmlarni bir xil vaaniq yozish ushuni ishlatiladigan belgilashlar va qoidalar tizimidir. Algoritmik til oddiy tilga yaqin bo‘lib u matematik belgilarni (yuqorida aytilganidek) o‘z ishigaoladi. Qo‘yilgan masalalarni eshishga tuzilgan algoritmlarni to‘g‘ridan-to‘g‘ri mashinaga berib, eshib bo‘lmaydi, shu sababli yozilgan algoritmnini biror bir algoritmik tilga o‘tkazish zarur. Har qanday algoritmik til o‘z qo‘llanilish sohasiga ega. Masalan, o‘quv jarayonlari ushuni Pascal, Delphi, VBA, java, C++dasturlash tillari va boshqalar.

1-misol: Kiritilgan n-natural sonni tub ko‘paytuvchilarga ajratuvchi algoritmnini Pascal dasturlash tilida ifodalanishini ko‘rib chiqamiz:

```
var i,k:integer; n:integer;
a:array[byte] of integer; label qq;
procedure opr(nn:integer);
begin i:=2;
while(nn>0) do
begin
if nn mod i=0 then write(i,' ');
i:=i+1;
nn:=nn div i;
end;
end;
begin
readln(n);
k:=1;
for i:=2 to n do
while (n mod i=0) do
begin a[k]:=i; write(a[k],' '); n:=n div i
; k:=k+1; end;
writeln;
readln;
end.
```

2-misol: $\int_0^5 x^2 dx$ — qiymatini hisoblovchi dastur tuzing.

```
procedure
TForm1.Button1Click(Sender: TObject);
var
h,a,x,d,b,s:real;
n:integer;
begin
a:=0; s:=0;
b:=5;
n:=10000;
h:=(b-a)/n;
x:=a;
while (x<b) do
begin
d:=sqr(x);
s:=s+d*h;
x:=x+h;
end;
end;
end.
```

Chapter 24), and using a search engine to quickly find pages on which particular information resides (related techniques are in Chapters 11 and 32).

- * Electronic commerce enables goods and services to be negotiated and exchanged electronically, and it depends on the privacy of personal information such as credit card numbers, passwords, and bank statements. The core technologies used in electronic commerce include public-key cryptography and digital signatures (covered in Chapter 31), which are based on numerical algorithms and number theory.
- * Manufacturing and other commercial enterprises often need to allocate scarce resources in the most beneficial way. An oil company may wish to know where to place its wells in order to maximize its expected profit. A political candidate may want to determine where to spend money buying campaign advertising in order to maximize the chances of winning an election. An airline may wish to assign crews to flights in the least expensive way possible, making sure that each flight is covered and that government regulations regarding crew scheduling are met. An Internet service provider may wish to determine where to place additional resources in order to serve its customers more effectively. All of these are examples of problems that can be solved using linear programming, which we shall study in Chapter 29.

Although some of the details of these examples are beyond the scope of this book, we do give underlying techniques that apply to these problems and problem areas. We also show how to solve many specific problems, including the following:

- * We are given a road map on which the distance between each pair of adjacent intersections is marked, and we wish to determine the shortest route from one intersection to another. The number of possible routes can be huge, even if we disallow routes that cross over themselves. How do we choose which of all possible routes is the shortest? Here, we model the road map (which is itself a model of the actual roads) as a graph (which we will meet in Part VI and Appendix B), and we wish to find the shortest path from one vertex to another in the graph. We shall see how to solve this problem efficiently in Chapter 24.
- * We are given two ordered sequences of symbols, $X = \{x_1, x_2, \dots, x_m\}$ and $Y = \{y_1, y_2, \dots, y_n\}$, and we wish to find a longest common subsequence of X and Y . A subsequence of X is just X with some (or possibly all or none) of its elements removed. For example, one subsequence of $\{A, B, C, D, E, F, G\}$ would be $\{B, C, E, G\}$. The length of a longest common subsequence of X and Y gives one measure of how similar these two sequences are. For example, if the two sequences are base pairs in DNA strands, then we might consider them similar if they have a long common subsequence. If X has m symbols and Y has n symbols, then X and Y have 2^m and 2^n possible subsequences,

respectively. Selecting all possible subsequences of X and Y and matching them up could take a prohibitively long time unless m and n are very small. We shall see in Chapter 15 how to use a general technique known as dynamic programming to solve this problem much more efficiently.

- We are given a mechanical design in terms of a library of parts, where each part may include instances of other parts, and we need to list the parts in order so that each part appears before any part that uses it. If the design comprises n parts, then there are $n!$ possible orders, where $n!$ denotes the factorial function. Because the factorial function grows faster than even an exponential function, we cannot feasibly generate each possible order and then verify that, within that order, each part appears before the parts using it (unless we have only a few parts). This problem is an instance of topological sorting, and we shall see in Chapter 22 how to solve this problem efficiently.
- We are given n points in the plane, and we wish to find the convex hull of these points. The convex hull is the smallest convex polygon containing the points. Intuitively, we can think of each point as being represented by a nail sticking out from a board. The convex hull would be represented by a tight rubber band that surrounds all the nails. Each nail around which the rubber band makes a turn is a vertex of the convex hull. (See Figure 33.6 on page 1029 for an example.) Any of the 2^n subsets of the points might be the vertices of the convex hull. Knowing which points are vertices of the convex hull is not quite enough, either, since we also need to know the order in which they appear. There are many choices, therefore, for the vertices of the convex hull. Chapter 33 gives two good methods for finding the convex hull.

These lists are far from exhaustive (as you again have probably surmised from this book’s heft), but exhibit two characteristics that are common to many interesting algorithmic problems:

1. They have many candidate solutions, the overwhelming majority of which do not solve the problem at hand. Finding one that does, or one that is “best,” can present quite a challenge.
2. They have practical applications. Of the problems in the above list, finding the shortest path provides the easiest examples. A transportation firm, such as a trucking or railroad company, has a financial interest in finding shortest paths through a road or rail network because taking shorter paths results in lower labor and fuel costs. Or a routing node on the Internet may need to find the shortest path through the network in order to route a message quickly. Or a person wishing to drive from New York to Boston may want to find driving directions from an appropriate Web site, or she may use her GPS while driving.

Not every problem solved by algorithms has an easily identified set of candidate solutions. For example, suppose we are given a set of numerical values representing samples of a signal, and we want to compute the discrete Fourier transform of these samples. The discrete Fourier transform converts the time domain to the frequency domain, producing a set of numerical coefficients, so that we can determine the strength of various frequencies in the sampled signal. In addition to lying at the heart of signal processing, discrete Fourier transforms have applications in data compression and multiplying large polynomials and integers. Chapter 30 gives an efficient algorithm, the fast Fourier transform (commonly called the FFT), for this problem, and the chapter also sketches out the design of a hardware circuit to compute the FFT.

Data structures

This book also contains several data structures. A *data structure* is a way to store and organize data in order to facilitate access and modifications. No single data structure works well for all purposes, and so it is important to know the strengths and limitations of several of them.

Technique

Although you can use this book as a “cookbook” for algorithms, you may someday encounter a problem for which you cannot readily find a published algorithm (many of the exercises and problems in this book, for example). This book will teach you techniques of algorithm design and analysis so that you can develop algorithms on your own, show that they give the correct answer, and understand their efficiency. Different chapters address different aspects of algorithmic problem solving. Some chapters address specific problems, such as finding medians and order statistics in Chapter 9, computing minimum spanning trees in Chapter 23, and determining a maximum flow in a network in Chapter 26. Other chapters address techniques, such as divide-and-conquer in Chapter 4, dynamic programming in Chapter 15, and amortized analysis in Chapter 17.

Hard problems

Most of this book is about efficient algorithms. Our usual measure of efficiency is speed, i.e., how long an algorithm takes to produce its result. There are some problems, however, for which no efficient solution is known. Chapter 34 studies an interesting subset of these problems, which are known as NP-complete.

Why are NP-complete problems interesting? First, although no efficient algorithm for an NP-complete problem has ever been found, nobody has ever proven

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Chapter 1 The Role of Algorithms in Computing

that an efficient algorithm for one cannot exist. In other words, no one knows whether or not efficient algorithms exist for NP-complete problems. Second, the set of NP-complete problems has the remarkable property that if an efficient algorithm exists for any one of them, then efficient algorithms exist for all of them. This relationship among the NP-complete problems makes the lack of efficient solutions all the more tantalizing. Third, several NP-complete problems are similar, but not identical, to problems for which we do know of efficient algorithms. Computer scientists are intrigued by how a small change to the problem statement can cause a big change to the efficiency of the best known algorithm.

You should know about NP-complete problems because some of them arise surprisingly often in real applications. If you are called upon to produce an efficient algorithm for an NP-complete problem, you are likely to spend a lot of time in a fruitless search. If you can show that the problem is NP-complete, you can instead spend your time developing an efficient algorithm that gives a good, but not the best possible, solution.

As a concrete example, consider a delivery company with a central depot. Each day, it loads up each delivery truck at the depot and sends it around to deliver goods to several addresses. At the end of the day, each truck must end up back at the depot so that it is ready to be loaded for the next day. To reduce costs, the company wants to select an order of delivery stops that yields the lowest overall distance traveled by each truck. This problem is the well-known “traveling-salesman problem,” and it is NP-complete. It has no known efficient algorithm. Under certain assumptions, however, we know of efficient algorithms that give an overall distance which is not too far above the smallest possible. Chapter 35 discusses such “approximation algorithms.”

Parallelism

For many years, we could count on processor clock speeds increasing at a steady rate. Physical limitations present a fundamental roadblock to ever-increasing clock speeds, however: because power density increases superlinearly with clock speed, chips run the risk of melting once their clock speeds become high enough. In order to perform more computations per second, therefore, chips are being designed to contain not just one but several processing “cores.” We can liken these multicore computers to several sequential computers on a single chip; in other words, they are a type of “parallel computer.” In order to elicit the best performance from multicore computers, we need to design algorithms with parallelism in mind. Chapter 27

FOYDALANILGAN ADABIYOTLAR RO‘YHATI:

1. Thomas H. Cormen va b. Intruduction to algorithms. Massachusetts Institute of Technology. London 2009.(5-10pp)
2. Thomas H. Cormen va b. Intruduction to algorithms. Massachusetts Institute of Technology. London 2009. (11-13pp)
3. Слинкин Д.А.Основы программирования на Турбо-Паскале: Учебно-методическое пособие для студентов вузов. Шадринск: Изд-во Шадринского пединститута, 2003. - 244 с. (10 -p)
4. M.U.Ashurov, N.D.Mirzaxmedova .Turbo Pascal dasturlash tili.(Uslubiy qo‘llanma),Toshkent TDPU – 2011 (3-10pp)

¹ Thomas H. Cormen va b. Intruduction to algorithms. Massachusetts Institute of Technology. London 2009.(5-10pp)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

MA‘RUZA №2

MAVZU: ALGORITMLAR SAMARADORLIGINI BAHOLASH.

Reja:

1. Algoritmlar samaradorligini baholash.
2. Algoritmlar va boshqa texnologiyalar.
3. Tarixiy ma'lumotlar.
4. Algoritmlar nazariyasi fanining predmeti, maqsadi va vazifalari.
5. Algoritmlar nazariyasining nazariy va amaliy ahamiyati.

Kalit so‘zlar: Alan Tyuring, Aloiz Chyorch, Emil Post, Tyuring mashinasi, Post mashinasi, Chyorchning lyamda- hisoblanuvchanlik usuli, $P=NP$ muammosi.

1. Algoritmlar samaradorligini baholash.

Kompyuteringizning tezligi va xotira miqdorini abadiy oshirish mumkin, deylik. Bu holatda algoritmlar o‘rganish kerakmi? Bor bo‘lishi mumkin, lekin faqat namoyish etish uchun, echim usulini cheklangan vaqti bor va u to‘g‘ri javob beradi.

Kompyuterlar juda tez bo‘lganda, masalani echishga har qanday konkret usul mos kelarmidi.

Albatta, bugungi kunda juda samarali kompyuterlar, lekin ularning ishlashi juda katta bo‘lishi mumkin emas. Xotira ham arzon, lekin bepul bo‘lishi mumkin emas. Shunday qilib, hisob-vaqti - cheklangan resurs, shuningdek xotira miqdori ham. Siz donolik bilan bu resurslarini boshqarishingiz kerak, bunga algoritmlardan, vaqt va xotira xarajatlaridan samarali foydalanish kerak.

Har xil masalalarni yechish uchun mo‘ljallangan, turli xil algoritmlar, samaradorligi bo‘yicha sezilarli darajada farq qiladi. Bu farqlar juda katta bo‘lishi mumkin ekan. Masalan, ikki saralash algoritmlar, 2-darsta muhokama qilinadi. Birinchisini bajarish uchun, saralashni joylashtirish, bunga vaqt kerak bo‘ladi, shunday baholanmoqda $c_1 n^2$, n - bu saralash elementlarning soni, c_1 bo‘lsa bu – doimiy, n ga bog‘liq emas. Shunday qilib, bu algoritmni vaqti taxminan n^2 proportsional.

Ikkinchi algoritm amalga oshirish uchun, saralash birlashtirishi, vaqt talab etadi, taxminan $c_2 n \lg n$ ga teng, $\lg n$ - bu $\log_2 n$ qisqa yozuvi, c_2 bu - boshqa doimiy n ga bog‘liq emas. Odatda doimiy usul qo‘shimchalar doimiy birlashtirish usulidan kichikroq, $c_1 < c_2$. Doimiy omillar algoritmni ish vaqtiga juda kam ta‘sir qiladigan bog‘liq omillardan ko‘ra, shunga ishonch hosil qilaylik. Saralashni joylashtirish algoritmni ish vaqtini shunday yozaylik $c_1 n$, birlashtirish saralashini esa $c_2 n \lg n$.

Joylashtirish saralashi n omilga ega, birlashtirish saralashi esa $\lg n$ ga ega bu esa sezilarli darajada kamligini ko‘rishimiz mumkin. Kiritish hajmi n etarlicha katta bo‘lganda qo‘shish saralashi odatda tezroq bo‘ladi, saralash ob'ektlar kichik hajmdagi birlashtirishda, katta n uchun ahamiyatsiz qiymati $\lg n$ nisbatan n to‘liq doimiy farqi qadriyatlar o‘rnini qoplash, aslida birlashtirish yanada sezilarli namoyon bo‘ladi, saralash afzalligi ziyoda. Bu doimiy c_1 , c_2 dan necha marta kam muhim emas. Saralash elementlarini sono ishshi bilan burilish nuqtasi hosil bo‘ladi, shunda birlashtirish saralashi yanada samarali bo‘ladi.

Misol tarzida ikkita A va B kompyuterlarni ko‘rib chiqamiz. A kompyuteri ancha tezroq, va unda joylashtirish saralashi algoritmi ishlaydi, B kompyuter esa sekin va unda saralash algoritmi birlashtirish usuli bilan ishlaydi. Har ikkita kompyuterlar bir nechta saralashni bajarishi kerak. Kompyuter A sekundiga o‘n milliard ko‘rsatmalar bajaradi, B kompyuter sekundiga faqat o‘n million ko‘rsatmalar bajaradi, shunday qilib A kompyuteri ming marta B kompyuterdan tez. Saralash birlashtirish yuqori darajadagi til yordamida bir programct tomonidan amalga oshirilgan.

Bu kompilyator juda samarali emas edi, va natija $50n \lg n$ ko‘rsatmalarga bajaradigan kod paydo bo‘ldi.

O‘n million raqamlarini tartiblashtirish uchun A kompyuterga kerak bo‘ladi:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$$\frac{2 * (10^7)^2}{10^{10}} = 20000$$

B kompyuterga kerak bo‘ladi

$$\frac{50 * 10^7 \lg 10^7}{10^7} \approx 1163$$

Ko‘rib turganingizdek, kod bilan foydalanish, ish vaqti sekin ko‘tarilganda, yomon komilyator bilan ham eng sekin kompyuterda ham 17 marta kam vaqt talab qiladi.

Qo‘shish usuli joylashtirish usulidan samaraliroq ekanligini quyida keltirilgan jadral ma’lumotlarini tahlili orqali keltiramiz.

Kompyuterlar	Saralanadigan sonlar soni	Saralovchi algoritmi	Talab qilinadigan vaqt
A (tez ishlovchi 1 sekunda 10 mlrd amal bajaradi)	10 mlnta (taqriban 80 mb)	Joylashtirish usuli (tajribali dasturchi tomonidan yaratilgan algoritm saralash uchun $2n^2$ amal bajariladi)	$\frac{2 * (10^7)^2 \text{ buyruqlar}}{10^{10} \text{ buyruq/sec}}$ <p>=20000 sec (5,5 soatdan ko‘proq)</p>
B (sekin ishlovchi 1 sekunda 10 mln amal bajaradi)		Qo‘shish usuli (o‘rta darajali dasturchi tomonidan yaratilgan algoritm saralash uchun 50n lgn amal bajariladi))	$\frac{50 * 10^7 \lg 10^7}{10^7} \approx 1163 \text{ sekund}$ <p>(20 min dan kam)</p>

2. Algoritmlar va boshqa texnologiyalar

Yuqoridagi misol shuni ko‘rsatadiki, kompyuter apparat kabi algoritmlarni ham, texnologiya sifatida hisobga olinishimiz kerak.

Umumiy tizim ish faoliyatini algoritm samaradorligiga ham bog‘liq, va apparat kuchiga ham. Algoritm rivojlantirish sohasida jadal rivojlantirish bo‘lyapti, boshqa kompyuter texnologiyalaridek.

Savol tug‘iladi, algoritmlar shunchalik muhimi, zamonaviy kompyuterlarda ishlaydigan bo‘lsin, agar shunday kabi yuqori texnologiyalar boshqa sohalarda ulkan yutuqlarga erishilgan bo‘lsa

- zamonaviy kompyuter mimarileri va ularning ishlab chiqarish texnologiyalari;
- osonlik bilan erishish, intuitiv grafik foydalanuvchi interfeysi (GUI);
- Ob'ektga yo‘naltirilgan tizimlar;
- Integratsiyalashgan veb texnologiyasi;
- tezroq tarmoqlari, simli va simsiz.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Misol uchun, bir joydan boshqasiga olish uchun qanday belgilaydigan Web xizmat. Uni amalga oshirish bir yuqori samarali apparat, grafik foydalanuvchi interfeysi, bir global tarmoq va, ehtimol, bir ob'ekt yo‘naltirilgan yondashuv yotadi.

Bundan tashqari, bunday yo‘nalishlarini topish kabi bir berilgan veb-xizmati tomonidan amalga muayyan operatsiyalar uchun zarur algoritmlarni foydalanish, ko‘rish va enterpolasyon manzilini, xaritalar bilan foydalaniladi. Bundan tashqari, dastur, yuqori saviyada algoritmik mazmunini talab qilmaydi, kuchli algoritmlarga bog‘liq. Bu dastur ishlash apparat ishiga bog‘liq ekanligi ma‘lum, va amaliy rivojlanishida turli algoritmlardan foydalaniladi.

Biz hammamiz bilamizki, ilova yaqindan grafik foydalanuvchi interfeysi bilan bog‘liq, va har qanday grafik foydalanuvchi interfeysini ishlab chiqish uchun talab algoritmlari kerak bo‘ladi. Tarmoq ustida ishlaydigan ilovalarni eslatib o‘tamiz.

Ular faoliyat olib borishlari uchun, algoritmlarga asoslarga yo‘nalishni olib borishlari kerak bo‘ladi. Eng keng tarqalgan dasturlar tilda tuziladi, mashinadan farqli. Ularning kodi turli kompilyator va interpretatorlar bilan ishlov beriladi, turli algoritmlardan keng foydalanadi. Bundan tashqari, kompyuterlar kuchini doimiy o‘sishi, ular tobora murakkab vazifalarni hal qilish uchun qo‘llaniladi. Biz muammoni murakkabligini ortishimiz bilan, ikki saralash usullari qiyosiy tahlili misolida ko‘rib turganimizdek eng muhim farqlar algoritmlarini samaradorligini ko‘rinadi oshirilmoqda. Asosiy algoritmlar va ularni rivojlantirish usullari-asosiy xususiyatlatdan biri. Zamonaviy kompyuter texnologiyalari bilan, ayrim vazifalarni algoritmlarni bilmagan holda ham qilinishi mumkin, lekin bu sohada kop narsaga erishish mumkin.

Mashqlar

1.2.1 Dastur darajasida zarur bo‘lgan algoritmik content dasturini misol qilib keltiring va bu algoritmlarni funktsiyasini muhokama qiling

1.2.2 Deylik, bitta mashinada ikkita saralash algoritmni qiyosiy tahlil amalga oshirilmoqda. N elementlarni joylashtirish saralashi uchun $8n^2$ kerak bo‘ladi, birlashtirish saralashi uchun $64n \lg n$ qadamlar kerak bo‘ladi. Joylashtirish saralashi birlashtirish saralashidan qiymati oshsa, n ni qiymati qancha bo‘lishi kerak?

1.2.3 Ikkita algoritm bitta mashinada amalda bo‘lsa, n algoritmni qaysi minimal qiymati, ish vaqti $100n^2$ formula bilan aniqlansa, ish vaqti 2^n formula bilan aniqlangan qaysi biri tezroq ishlaydi. Kimning yugurib vaqt $100p^2$ bilan belgilanadi, uning ish vaqti, ham algoritmlar Shu mashina amalga bo‘lsa, sifatida ifodalangan bir algoritm tezroq n algoritm qaysi minimal qiymati da?

Masala

1.1. Algoritmlarni ish vaqtini solishtirish

Quyida bir jadval bo‘lib, satrlari turli vazifalarga mos $f(n)$, ustunlaru esa- vaqt qiymatiga t. N ni maksimal qiymatlari bilar jadvalni toldiring, masala t vaqti bilan yechilishi mumkin, agar masalani yechish uchun algoritmni ish vaqti $f(n)$ mikro sekundga teng bo‘lsa.

	1 sekund	1 minut	1 soat	1 kun	1 month	1 yil	1 asr
$\lg n$							
\sqrt{n}							
n							
$n \lg n$							
n^2							
n^2							
n^2							
$n!$							

1.2 Algorithms as a technology

Suppose computers were infinitely fast and computer memory was free. Would you have any reason to study algorithms? The answer is yes, if for no other reason than that you would still like to demonstrate that your solution method terminates and does so with the correct answer.

If computers were infinitely fast, any correct method for solving a problem would do. You would probably want your implementation to be within the bounds of good software engineering practice (for example, your implementation should be well designed and documented), but you would most often use whichever method was the easiest to implement.

Of course, computers may be fast, but they are not infinitely fast. And memory may be inexpensive, but it is not free. Computing time is therefore a bounded resource, and so is space in memory. You should use these resources wisely, and algorithms that are efficient in terms of time or space will help you do so.

Efficiency

Different algorithms devised to solve the same problem often differ dramatically in their efficiency. These differences can be much more significant than differences due to hardware and software.

As an example, in Chapter 2, we will see two algorithms for sorting. The first, known as *insertion sort*, takes time roughly equal to $c_1 n^2$ to sort n items, where c_1 is a constant that does not depend on n . That is, it takes time roughly proportional to n^2 . The second, *merge sort*, takes time roughly equal to $c_2 n \lg n$, where $\lg n$ stands for $\log_2 n$ and c_2 is another constant that also does not depend on n . Insertion sort typically has a smaller constant factor than merge sort, so that $c_1 < c_2$. We shall see that the constant factors can have far less of an impact on the running time than the dependence on the input size n . Let's write insertion sort's running time as $c_1 n \cdot n$ and merge sort's running time as $c_2 n \cdot \lg n$. Then we see that where insertion sort has a factor of n in its running time, merge sort has a factor of $\lg n$, which is much smaller. (For example, when $n = 1000$, $\lg n$ is approximately 10, and when n equals one million, $\lg n$ is approximately only 20.) Although insertion sort usually runs faster than merge sort for small input sizes, once the input size n becomes large enough, merge sort's advantage of $\lg n$ vs. n will more than compensate for the difference in constant factors. No matter how much smaller c_1 is than c_2 , there will always be a crossover point beyond which merge sort is faster.

For a concrete example, let us pit a faster computer (computer A) running insertion sort against a slower computer (computer B) running merge sort. They each must sort an array of 10 million numbers. (Although 10 million numbers might seem like a lot, if the numbers are eight-byte integers, then the input occupies about 80 megabytes, which fits in the memory of even an inexpensive laptop computer many times over.) Suppose that computer A executes 10 billion instructions per second (faster than any single sequential computer at the time of this writing) and computer B executes only 10 million instructions per second, so that computer A is 1000 times faster than computer B in raw computing power. To make the difference even more dramatic, suppose that the world's craftiest programmer codes insertion sort in machine language for computer A, and the resulting code requires $2n^2$ instructions to sort n numbers. Suppose further that just an average programmer implements merge sort, using a high-level language with an inefficient compiler, with the resulting code taking $50n \lg n$ instructions. To sort 10 million numbers, computer A takes

$$\frac{2 \cdot (10^7)^2 \text{ instructions}}{10^{10} \text{ instructions/second}} = 20,000 \text{ seconds (more than 5.5 hours) ,}$$

while computer B takes

$$\frac{50 \cdot 10^7 \lg 10^7 \text{ instructions}}{10^7 \text{ instructions/second}} \approx 1163 \text{ seconds (less than 20 minutes) .}$$

By using an algorithm whose running time grows more slowly, even with a poor compiler, computer B runs more than 17 times faster than computer A! The advantage of merge sort is even more pronounced when we sort 100 million numbers: where insertion sort takes more than 23 days, merge sort takes under four hours. In general, as the problem size increases, so does the relative advantage of merge sort.

Algorithms and other technologies

The example above shows that we should consider algorithms, like computer hardware, as a *technology*. Total system performance depends on choosing efficient algorithms as much as on choosing fast hardware. Just as rapid advances are being made in other computer technologies, they are being made in algorithms as well.

You might wonder whether algorithms are truly that important on contemporary computers in light of other advanced technologies, such as

- * advanced computer architectures and fabrication technologies,
- * easy-to-use, intuitive, graphical user interfaces (GUIs),
- * object-oriented systems,
- * integrated Web technologies, and
- * fast networking, both wired and wireless.

The answer is yes. Although some applications do not explicitly require algorithmic content at the application level (such as some simple, Web-based applications), many do. For example, consider a Web-based service that determines how to travel from one location to another. Its implementation would rely on fast hardware, a graphical user interface, wide-area networking, and also possibly on object orientation. However, it would also require algorithms for certain operations, such as finding routes (probably using a shortest-path algorithm), rendering maps, and interpolating addresses.

Moreover, even an application that does not require algorithmic content at the application level relies heavily upon algorithms. Does the application rely on fast hardware? The hardware design used algorithms. Does the application rely on graphical user interfaces? The design of any GUI relies on algorithms. Does the application rely on networking? Routing in networks relies heavily on algorithms. Was the application written in a language other than machine code? Then it was processed by a compiler, interpreter, or assembler, all of which make extensive use

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

of algorithms. Algorithms are at the core of most technologies used in contemporary computers.

Furthermore, with the ever-increasing capacities of computers, we use them to solve larger problems than ever before. As we saw in the above comparison between insertion sort and merge sort, it is at larger problem sizes that the differences in efficiency between algorithms become particularly prominent.

Having a solid base of algorithmic knowledge and technique is one characteristic that separates the truly skilled programmers from the novices. With modern computing technology, you can accomplish some tasks without knowing much about algorithms, but with a good background in algorithms, you can do much, much more.

Exercises

1.2-1

Give an example of an application that requires algorithmic content at the application level, and discuss the function of the algorithms involved.

1.2-2

Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n , insertion sort runs in $8n^2$ steps, while merge sort runs in $64n \lg n$ steps. For which values of n does insertion sort beat merge sort?

1.2-3

What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
\sqrt{n}							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

² Thomas H. Cormen va b. Intruduction to algorithms. Massachusetts Institute of Technology. London 2009. (11-13pp)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3. Tarixiy ma’lumotlar.

Bizgacha yetib kelgan intuitiv ma’nodagi algoritm eramizdan avvalgi III- asrda Yevklid tomonidan taklif qilingan. Ushbu algoritm juda mashhur bo‘lib, XX-asr boshlarigacha «algoritm» so‘zining o‘zi «Yevklid algoritmi» ma’nosida ishlatilib kelindi. Boshqa matematik masalalarni bosqichli yechishni tasvirlash uchun esa «usul» so‘zidan foydalanilgan.

Zamonaviy algoritmlar nazariyasi rivojidadagi boshlang‘ich nuqta deb, nemis matematigi Kurt Gyodelning ilmiy ishini ko‘rsatib o‘tish mumkin (1931 y. Simvolik mantiqlarning to‘lamlasligi to‘g‘risidagi teorema). Ushbu ishda ba’zi matematik muammolarni qaysidir sinfga taalluqli algoritmlar yordamida hal etib bo‘lmasligi ko‘rsatib berilgan.

1936 yilda Algoritmilar nazariyasi bo‘yicha birinchi fundamental ilmiy ishlar bir-biridan alohida tarzda Alan Tyuring, Aloiz Chyorch va Emil Postlar e’lon qildilar. Ular tomonidan taklif etilgan TM, Post mashinasi va Chyorchning lyamda-hisoblanuvchanlik usuli algoritm formalizmining ekvivalent shakllaridir. Ular tomonidan taklif etilgan tezislar algoritm intuitiv tushunchasi va formal tizimlarning ekvivalentligini ta’kidlab berdi. Algoritmik yechimsiz muammolarning formulirovkasi va isboti ushbu ishlarning muhim natijasi bo‘ldi.

1950 - yillarda Algoritmilar nazariyasi rivojlanishiga rus matematiklari Kolmogorov va Markovlari z hissalarini qo‘shdilar. 60-70-yillarga kelib Algoritmilar nazariyasi fanida quyidagi mustaqil yo‘nalishlar ajralib chiqdi:

- Klassik algoritmlar nazariyasi (formal tillar terminlarida masalalarni ifodalash, yechimli masala tushunchasi, 1965 yilda Edmonds tomonidan ta’riflangan $P=NP$ muammosi, NP to‘liq masalalar sinfining ochilishi va tekshirilishi);
- Algoritmilarning asimptotik tahlili nazariyasi (asimptotik baholash usullari, algoritmlarning murakkabligi, algoritmlarni baholash kriteriyalari va h.k.). Ushbu yo‘nalish rivojiga Knut, Axo, Xopkroft, Ulman, Karp kabi olimlar o‘z hissalarini qo‘shdilar;
- hisoblash algoritmlarining amaliy tahlili nazariyasi (algoritmilarning mehnattalabligi oshkor funksiyasini topish, funksiyalarning chegaraviy tahlili, ratsional algoritmlarni tanlash metodikasi). Ushbu yo‘nalish rivojlanishiga sabab bo‘lgan ilmiy ish D.Knutning “Искусство программирования для ЭВМ” kitobidan iborat.

4. Algoritmilar nazariyasi fanining maqsadi va vazifalari.

Algoritmilar nazariyasi – Informatika va tabiiy matematikaning fundamental qismiga oid fan bo‘lib, uning davomi bevosita samarali dasturlarni tuzish, sonli usullar, mukammallashtirish usullari va ob’ektga yo‘naltirilgan dasturlash sohalarida o‘z nazariy va amaliy tadbiqini topadi.

Fanning maqsadi – talabalarda umumiy(intuitiv) va formal ma’nodagi algoritm haqidagi tushunchalarni ajrata olish, muayyan masalani yechish uchun algoritm mavjudligi haqida tushunchalarni shakllantirish, masalani yechish uchun tatbiq etilishi mumkin bo‘lgan algoritmlar orasida eng samaralisini ajratib olish, yaratilgan yoki mavjud algoritmlarni murakkablik ko‘rsatkichlarini baholash kabi bir qator nazariy va amaliy muammolar bo‘yicha bilim va ko‘nikmalarni hosil qilishdan iborat.

Fanning vazifasi – algoritmning formal ta’riflari asosida yechimga ega masalalarning natijaviyligini o‘rnata bilishga erishish. Muayyan algoritmlarning murakkablik funksiyasi qanday bo‘lishini anglashga o‘rgatish va amalda sinash ko‘nikmasini hosil qilish.

Algoritmilar nazariyasi turli yo‘nalishlarining yutuqlarini umumlashtirib, uning maqsadi va vazifalarini quyidagicha ham ko‘rsatib o‘tish mumkin:

- Algoritm tushunchasini formallashtirish va formal algoritmik tizimlarni tekshirish;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

- Bir qator masalalarning algoritmik yechimsizligini formal isbotlash;
- Masalalar klassifikatsiyasi, murakkablik sinflarini aniqlash va tekshirish;
- Algoritm murakkabligining asimptotik tahlili;
- Rekursiv algoritmlarni tekshirish va tahlil qilish;
- Algoritm qiyosiy tahlili uchun mehnattalablik oshkor funksiyasini topish;
- Algoritm sifatini qiyosiy baholash kriteriyalarini ishlab chiqish.

5. Algoritm nazariyasining nazariy va amaliy ahamiyati.

Algoritm nazariyasidan olingan nazariy natijalardan amalda ancha keng foydalanilmoqda. Bunda ikki aspektni alohida ko‘rsatib o‘tish mumkin:

Nazariy aspekt: qandaydir masalani tekshirish natijasida “Ushbu masala prinsipial jihatdan algoritmik yechimli? - degan savolga javob berish imkoniyati mavjud. Algoritmik yechimsiz masalalar TM to‘xtashi masalasiga olib kelinishi mumkin. Algoritmik yechimli masalalar uchun esa, ushbu masalaning NP to‘liq masalalar sinfiga mansubligi muhim ikkinchi nazariy savol bo‘lib hisoblanadi.

Amaliy aspekt: Algoritm nazariyasi usullari quyidagi vazifalarni bajarishga imkon beradi:

- Berilgan masalani yechish algoritmlari to‘plamidan eng rasional algoritmni tanlash;
- Murakkab masalalarni yechish algoritmlarini vaqt jihatidan baholash;
- Kriptografik metodlar uchun muhim bo‘lgan masala yechimini ma’lum vaqt oralig‘ida olib bo‘lmasligini ishonchli baholash;
- Amaliy tahlil asosida axborotlarni qayta ishlash sohasidagi masalalarni yechish effektiv algoritmlarini ishlab chiqish va rivojlantirish.

Nazorat savollari:

1. Algoritm nazariyasi faniga hissa qo‘shgan olimlardan kimlarni bilasiz?
2. Algoritm nazariyasi fanining maqsadlari nimadan iborat?
3. Algoritm nazariyasi fanining vazifalari nimalardan iborat?
4. Algoritm nazariyasi fani qaysi yo‘nalishlar bo‘yicha rivojlanib keldi?
5. Algoritm nazariyasi fani yutuqlarining nazariy va amaliy ahamiyati nimadan iborat?

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

MA’RUZA №3

MAVZU: ALGORITMLAR TAHLILI.

REJA:

1. Algoritm sifatini baholashning mezonlari.
2. Algoritmni tahlil qilish usullari.
3. Algoritmni to‘g‘riligini tekshirish.

Dastur to‘g‘riligini isbotlashning eng keng tarqalgan turi – bu uni testlardan o‘tkazishdir.

Algoritmni tekshirishda nazoratchi boshlang‘ich ma‘lumotlarni majmui algoritmik test deb nomlanadi.

To‘g‘ri deb shunday algoritmga aytiladiki, u masalaning qo‘yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang‘ich ma‘lumotlar bilan ham shakllantirib biladi. Odatda, dastur bergan natijalar ma‘lum bo‘lgan yoki qo‘lda hisoblangan ma‘lumotlar bilan taqqoslanadi, va ular to‘g‘riligi aniqlansa dastur to‘g‘ri ishlaydi degan hulosaga kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo‘lmaydi, ya‘ni dastur ishlamaydigan hamma holatlarni hisobga olib bo‘lmaydi.

Gudman va Xidetniyemi [2] lar tomonidan algoritm to‘g‘riligini isbotlash uchun quyidagi uslubiyat taklif qilingan.

Algoritm 0 dan m gacha bo‘lgan qadamlar ketma-ketligi ko‘rinishida tavsiflangan deb tahmin qilaylik. Har bir qadam uchun qandaydir asoslanishni taklif etamiz. Xususan, qadamdan oldin va keyin ishlaydigan shartlar haqida lemma kerak bo‘lishi mumkin. Shu bilan birgalikda, algoritm chekliligining isbotini ham taklif etamiz, va hamma ruxsat etilgan kiritish ma‘lumotlarini tekshirib, hamma mumkin bo‘lgan chiqarish ma‘lumotlarni olamiz. Algoritmni to‘g‘riligi bilan samaradorligi o‘rtasida hech qanday aloqa yo‘qligini ta‘kidlab o‘tamiz. Aslida hamma talablarga bir xil yahshi javob beradigan algoritm kamdan-kam ishlab chiqiladi.

Algoritmni amalga oshirish

Algoritmni amalga oshirish deganda, EHM uchun dasturni yozish deb tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

- 5.1. Asosiy o‘zgaruvchilarni aniqlash.
- 5.2. O‘zgaruvchilarning turlarini aniqlash.
- 5.3. Nechta massiv yoki fayllar va qanday kattalikda ular kerak bo‘ladi?
- 5.4. Bog‘lanilgan ro‘yhatlardan foydalanish ma‘nolimi?
- 5.5. Qanday dasturiy qismlar kerak bo‘lishi mumkin (tayyor bo‘lsa ham)?
- 5.6. Qaysi dasturlash tilini tanlash?

Dastur yozish yoki tuzishning hilma-hil usullari va uslublari mavjud.

Algoritmni va uning murakkabligini tahlil qilish

Algoritmni tahlil qilishdan maqsad – algoritmga ma‘lumotlarni aniq muvaffaqiyatli qayta ishlash uchun kerak bo‘ladigan xotira hajmi va ishlash vaqtining baholari va chegaralarini olish. Bir masalani yechadigan ikki algoritmni taqqoslash uchun qandaydirsonli mezon topish kerak.

Faraz qilaylik, A – qandaydir bir turkumdagi masalalarni yechadigan algoritm, n – esa shu turkumdagi alohida bir masalaning kattaligi. Umumiy holda, n – oddiy skalyar yoki massiv yoki kiritiladigan ketma – ketlikning uzunligi bo‘lishi mumkin. $f_A(n)$ - n kattalikdagi

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

ixtiyoriy masalani yechadigan algoritm A bajarish kerak bo‘lgan asosiy amallarni (qo‘shish, ayirish, taqqoslash,...) yuqori chegarasini beradigan ishchi funksiya. Algoritmning sifatini baholash uchun quyidagi mezonni ishlatamiz.

Agar $f_A(n)$ o‘shish tartibi n dan bog‘liq bo‘lgan polinomdan katta bo‘lmasa, A algoritm polinomial deb aytiladi, aks holda algoritm A eksponensial hisoblanadi.

Shular bilan birgalikda tahlil jarayonida ko‘p matematik fanlarda standart bo‘lgan iboralar ishlatiladi.

$f_A(n)$ funksiya $O[g(n)]$ deb belgilanadi, va $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = const \neq 0$ bo‘lganda, uni tartibi

katta n lar uchun $g(n)$ deb qabul qilinadi. Demak $f(n) = O[g(n)]$.

$f_A(n)$ funksiyasi $o[z(n)]$ deb katta n lar uchun belgilanadi, va unda $\lim_{n \rightarrow \infty} \frac{h(n)}{z(n)} = 0$ sharti

bajariladi.

Bu begilar “katta O” va “kichik o” deb nomlanadi. Agar $f(n) = O[g(n)]$ bo‘lsa, ikkala funksiya ham $n \rightarrow \infty$ bo‘lganda bir xil tezlikda o‘sadi.

Agar $f(n) = O[g(n)]$ bo‘lsa, unda $g(n)$, $f(n)$ nisbatan ancha tez o‘sadi.

Demak, $P_k(n)$ - qandaydir n o‘zgaruvchidan bog‘liq va k darajadagi polinom uchun $f_A(n) = O[P_k(n)]$ yoki $f_A(n) = oP_k(n)$ bo‘lganda algoritm polinomial hisoblanadi, aks holda algoritm eksponensial.

Eksponensial algoritm yahshi ishlamaydigan deb hisoblanadi. Agar algoritmlar eksponensial bo‘lsa, ular orasida eng samaralisini topish kerak, n kattaligidagi masalani $O(2^n)$ qadamda yechadigan algoritm $O(n!)$ yoki $O(n^n)$ qadamda masalani yechadigan algoritmdan afzalroq.

Dasturni tekshirish

Biz dasturni har bir qismini tekshiradigan kirituvchi ma’lumotlar to‘plamini tanlashimiz kerak. Ko‘p murakkab algoritmlarni matematik tomondan tadqiq qilish yoki juda qiyin yoki mumkin emas. Bunday holatlarda algoritmnii faoliyat jarayonida va qiyinligi bo‘yicha tekshiradi. Bundan tashqari dasturlarni hisoblash imkoniyatlarini aniqlash uchun ham testlash maqsadga muvofiq. Ko‘p dasturlar qandaydir kiritiladigan ma’lumotlar bilan yahshi ishlasa, boshqalari bilan yomon ishlaydi. “Yahshi” lardan “yomon” larga o‘tish “mayin” bo‘lish kerak. Testlash uchun ma’lumotlar dasturning qiyinligiga, mavjud vaqt resurslariga, kiritish-chiqarishsoniga bog‘liq holda tanlanadi. Bu yerda analitik va eksperimental tahlil bir-birini to‘ldiradi.

Hujjatlashtirish

O‘zingiz yozmagan dastur kodini o‘qish juda qiyin. Bu muammoni hujjatlashtirish yordamida yechsa bo‘ladi. Hujjatlashtirish o‘z ichiga hamma yordamchi ma’lumotlarni oladi va dasturda nima bajarilishini tushuntirib beradi, xususan, blok-sxemalardagi boshqarishni uzatish, berilganlarni kiritish-chiqarish shaklini batafsil tavsif qilish, siklning parametrlari, yordamchi local va global proseduralarni bajarilishi va boshqalar.

Hujjatlashtirishning eng asosiy qoidasi bu “boshqalar yozgan dasturlarni qanday ko‘rishni istasangiz, o‘zingiz ham dasturni shunday ko‘rinishda rasmiylashtiring”.

Masalalar yechish.

1 - misol. Berilgan to‘rt xonali butun sonning raqamlari ko‘paytmasini toping.

Test

Test tartibi	Tekshirish	Son	Natija
1	Musbat son	2314	P = 24
2	Manfiy son	-1245	P = 40

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Algoritmi:

```

alg Butun_son (but Num, P)
arg Num
natija P
boshlbutun i, j, k, l
  Num := abs(Num)
  i := Num div 1000
  j := ((Num div 100) mod 10)
  k := ((Num div 10) mod 10)
  l := Num mod 10
  P := i * j * k * l;

```

Tamom

Turbo Pascaldagi dasturi:

```

Program Farrux;
Var Number, i, j, k, l, P : Integer;
BEGIN
  ReadLn(Number); Number:=Abs(Number);
  i := Number div 1000; Write(i:3);
  j := Number div 100 mod 10; Write(j:3);
  k := Number div 10 mod 10; Write(k:3);
  l := Number mod 10; WriteLn(l:3);
  P := i * j * k * l; WriteLn( P);
  ReadLn
END.

```

2 - misol. Butun qiymatli $A(N, M)$ matritsa berilgan. Agar matritsa satrining hech bo‘lmaganda biror elementi manfiy bo‘lsa, u holda bu satrning barcha elementlarini nollar bilan almashtiring

Test

Berilgan		Natija
N	A matritsa	A matritsa
3	$\begin{pmatrix} 1 & -2 & 1 \\ 1 & 2 & 1 \\ -1 & 2 & -2 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

Algoritmi

```

alg Modifikasiya(but N, haq jad A[1:N, 1:N])
boshl but i, j, lit Flag
kiritish N
sb iuchun 1 dan N gacha
  sbj uchun 1 dan N gacha
    kiritishA[i,j]
  so
so
sbi uchun 1 dan N gacha

```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```

j := 1; Flag := "Yuk"
sb toki (j<=N) va (Flag = "Yo‘q")
    agar A[i, j]<0 u holda Flag := "Ha"
    aks holda j:=j+1
hal bo‘ldi
so
agar Flag = "Ha" u holda
sbj uchun 1 dan N gacha A[i, j]:=0
    so
    hal bo‘ldi
so
sbi uchun 1 dan N gacha
    sbj uchun 1 dan N gacha
        chiqarishA[i, j]
    so
so
tamom.
    
```

Algoritmning bajarilishi

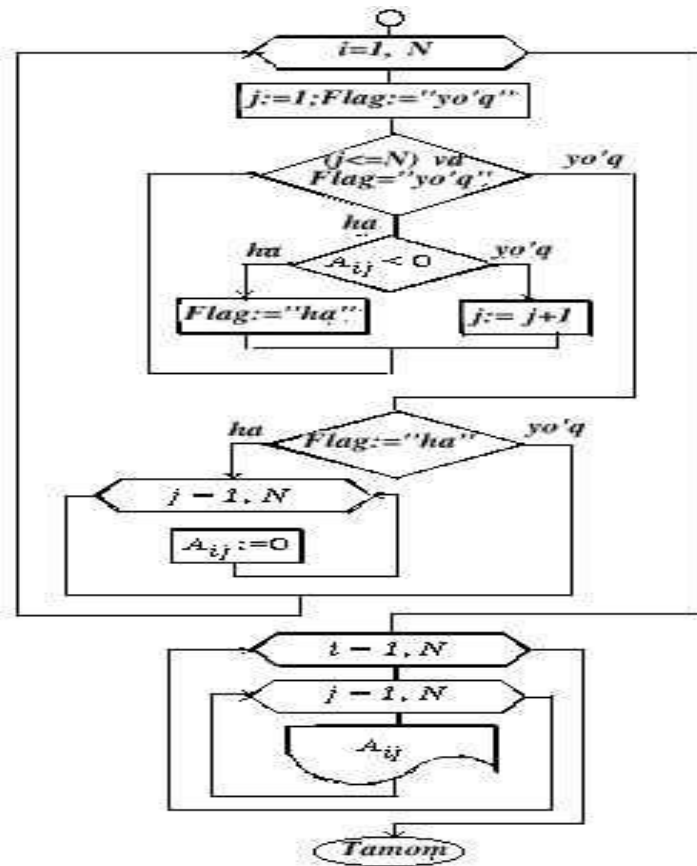
Tekshirilayotgan shartning belgilanishi:

(j<=N) va (Flag = "Yo‘q")=> (1)

i	Flag	j	(1)	A[i,j]<0	Flag="Ha"	A[i,j]
1	"Yo‘q" "Ha"	1	+	-	+	A[1,1]=0 A[1,2]=0 A[1,3]=0
		2	+	+		
		1	-(so)			
		2				
2	"Yo‘q"	1	+	-	-	
		2	+	-		
		3	+	-		
		4	-(so)			
3	"Yo‘q" "Ha"	1	+	+	+	A[3,1]=0 A[3,2]=0 A[3,3]=0
		1	-(so)			
		2				
		3				

Blok-sxemasi fragmenti:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



Turbo Pascaldagi dasturi:

```

Program Modify;
Var A : Array[1..10, 1..10] of Real;
    N, i, j : Integer;
Procedure InputOutput;
Begin
  ReadLn(N);
  For i := 1 to N do
    For j := 1 to N do
      begin Write('A[', i, ', ', j, '] = ');
            ReadLn(A[i, j])
          end;
  For i := 1 to N do
    begin
      For j := 1 to N do Write(A[i, j] : 5 : 1);
      WriteLn
    end;
End; { of InputOutput }
{-----}
Procedure Line(Var i : Integer);
Var Flag : Boolean;
Begin
  j := 1; Flag := FALSE;
  While (j<=N) and not Flag do
    If A[i, j]<0 then Flag:=TRUE else j:=j+1;
  If Flag then
  
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
For j := 1 to N do A[i, j] := 0
End;
{-----}
Procedure OutRes;
Begin
  WriteLn(' Natija- Matritsa:'); WriteLn;
  For i := 1 to N do
    begin
      For j := 1 to N do Write(A[i, j]:5:1);
      WriteLn
    end; ReadLn
End; { of OutRes }
BEGIN
  InputOutput;
  For i := 1 to N do Line(i);
  OutRes;
END.
```

MA'RUZA №4

MAVZU: ALGORITMNI ISHLAB CHIQISH METODLARI.

REJA:

1. Sxemalar va yangi algoritmlarni paydo qilishning usullari.
2. Algoritmik yechimga ega bo'lmagan masalalar.
3. O'z-o'ziga qo'llanuvchanlik muammosi.
4. Tyuring mashinasining o'z-o'ziga qo'llanuvchanligi.

Kalit so'zlar: Algoritmik yechimsizlik, diofant tenglama, hisoblanuvchi bo'lmagan funksiya, qo'llanuvchanlik, o'z-o'ziga qo'llanuvchanlik, kirish so'zi, chiqish so'zi.

1. Sxemalar va yangi algoritmlarni paydo qilishning usullari.

Algoritmlarni yaratish ijobiy ish, shuning uchun ixtiyoriy zarur algoritmlarni tuzish imkonini beradigan bir umumiy usul mavjud emas. Lekin algoritmlarni ishlab chiqishni asoslangan oddiy sxemalarini beradigan ko'pgina algoritmlashtirish nazariyalari bor. Bunday sxemalar va yangi algoritmlarni paydo qilishning o'rtasida bog'liqlik kuzatiladi. Tez uchraydigan va ko'p foydalaniladigan usullarni quyidagicha ajratib olish mumkin:

1. Algoritmlarni konstruksiyalash. Bu usulda yangi algoritm mavjud algoritmlardan tarkibiy qismlar sifatida foydalanib, bir-biriga moslab bir butunlik hosil qilish yo'li bilan ishlab chiqiladi.

2. Algoritmlarni ekvivalent qayta ishlash. Ikki algoritm ekvivalent hisoblanishi uchun quyidagi shartlar bajarilish kerak:

- Bittasi uchun mumkin bo'lgan dastlabki berilganlar varianti, ikkinchisi uchun ham mumkin bo'lishi kerak.

- Bir algoritmni qandaydir dastlabki ma'lumotga qo'llanilishi, ikkinchi algoritmni ham shu berilganga qo'llanilishiga kafolat beradi.

- Bir xil dastlabki berilgan ma'lumot uchun ikkala algoritm ham bir xil natija berishi. Lekin bu algoritmni ikki xil shakllarini ekvivalent deb nomlash noto'g'ridir.

Shunday qilib, algoritmni ekvivalent qayta ishlash deb, natijada dastlabki algoritmga ekvivalent algoritmni paydo qiladigan o'zgartirilishlarga aytiladi.

Misol tariqasida, algoritmni bir tildan boshqa tilga o'tkazishni keltirish mumkin. Shu bilan birgalikda algoritmni ekvivalent qayta ishlash usuli bilan keskin o'zgartirish mumkin, lekin bu holda asosiy e'tiborni dastlabki algoritmga nisbatan yahshi algoritmni yaratishga berish kerak.

3. Toraytiruvchi o'zgartirishlar. Bunday o'zgartirishlar natijasida dastlabki algoritmlar yechish kerak bo'lgan masalalarning xususiy holati yechimi algoritmlari ishlab

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

chiqiladi. Odatda, bu usulda ekvivalent qayta ishlash jarayonida algoritmi ixchamlashtirish maqsaddida foydalaniladi.

4. Formal usulni matematikaga bog‘liq bo‘lmagan muammoga qo‘llash. Buyerda matematik muammo matematik ko‘rinishga o‘tkazilib, uning algoritmini ishlab chiqishga uriniladi. Agar o‘xshash matematik masala yechimining algoritmi ma’lum bo‘lsa, undan foydalaniladi.

2. Algoritmik yechimga ega bo‘lmagan masalalar.

Algoritmik nazariyasida shunday masalalar mavjudki, ushbularni yechish algoritmlari mavjud emas. Bunday masalalar *algoritmik yechimsiz* deb ataladi. Odatda “*yangi*” masalalarning algoritmik yechimsizligi ularni oldindan ma’lum algoritmik yechimsiz masalalarga keltirish yo‘li bilan isbotlanadi. Shu bilan birga yangi masalaning yechimi mavjud bo‘lganda oldindan yechimsiz deb hisoblangan masalani ham yechish mumkinligi isbotlanadi. Bunday masalalar qatoriga *o‘z-o‘ziga qo‘llanuvchanlik* muammosi misol bo‘ladi.

Algoritmga aniq ta’rif berilganidan so‘ng berilgan ommaviy muammolar algoritmik yechimga ega bo‘lish yoki bo‘lmazlik masalasini hal etish imkoniyatlari paydo bo‘ldi. Algoritmik yechimga ega bo‘lmagan masalalar na’munalarini ko‘rib chiqamiz .

1936-yili A.Chyorch tomonidan predikatlar hisobi uchun formulalarning umumqiyamatli bo‘lish yoki bo‘lmazligini hal qiladigan algoritm mavjud emasligi isbotlandi.

1-ta’rif. Biror bir alifboning so‘zlar to‘plami o‘zining chekli sondagi o‘rniga qo‘yish qoidalari bilan birgalikda assotsiativ hisob deyiladi.

Assotsiativ hisobning ixtiyoriy ikkita so‘zi uchun bu ikkita so‘zning teng kuchli bo‘lish-bo‘lmazlik masalasi assotsiativ hisobda so‘zlarning *ekvivalentlik* muammosi deyiladi.

Bu masala 1911-yilda e’lon qilingan. 1946–47-yillarda rus matematigi A.A.Markov va amerikalik matematik E.Postlar ekvivalentlik muammosi algoritmik yechimga ega emasligini hal etganlar.

1955-yilda rus matematigi P.S.Novikov gruppalar nazariyasida so‘zlar ekvivalentligi muammosi algoritmik yechimga ega emasligini isbotladi.

1900-yilda matematiklarning Parijda bo‘lib o‘tgan ikkinchi halqaro kongressida yechilishi qiyin bo‘lgan 23 ta matematik muammolar e’lon qilindi. Shu muammolarning o‘ninchisida har qanday butun koeffitsientli n ta o‘zgaruvchili ko‘phad butun ildizlarga ega bo‘lish, bo‘lmazligini aniqlaydigan algoritm bor yoki yo‘qligini aniqlashdan iborat edi. Bunday ko‘phadlarga quyidagilar misol bo‘ladi :

$$f(x, y, z) = x^2 + y^2 + z^2 - 2xyz,$$

$$f(x) = 5x^3 - x^2 + x + 15.$$

Hususiy holda butun koeffitsientli bir noma’lumli

$$f(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n \quad (a_0 \neq 0)$$

ko‘rinishdagi n darajali ko‘phadning butun yechimlarini topish algoritmi mavjud ekanligi ma’lum.

1968-yili yuqorida keltirilgan masala umumiy holda algoritmik yechimga ega emasligi Yu.Matuyasevich tomonidan isbot qilindi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3. O‘z-o‘ziga qo‘llanuvchanlik muammosi.

Tyuring mashinasi haqida gapirganda, ixtiyoriy Tyuring mashinasi sxemasini kodlangan holda tasmaga yozish mumkinligini bilamiz. Xuddi shuningdek, ixtiyoriy Markovning normal algoritmining o‘rinlashtirish formulalarini ajratish uchun biror belgidan foydalanib kodlash mumkin. U holda Normal algoritmnining o‘zi so‘zga aylanadi va ixtiyoriy Normal algoritm uchun KIRISH so‘zi sifatida qo‘llanilishi mumkin. Xususiyl holda Normal algoritm o‘z-o‘ziga KIRISH so‘zi bo‘ladi.

Barcha algoritmlar ikki sinfga bo‘linadi: o‘z-o‘ziga qo‘llanuvchan va qo‘llanilmas:

O‘z-o‘ziga qo‘llanuvchan algoritmlar deb, o‘zining ifodasi ustida ishlab, ertami-kechmi to‘xtaydigan algoritmlarga aytiladi. Agar algoritm ishi cheksiz takrorlanuvchi bo‘lsa, bunday algoritmlar o‘z-o‘ziga qo‘llanilmas deyiladi.

Shunday qilib, haqli savol tug‘iladi: Qanday qilib u yoki bu algoritmnining o‘z-o‘ziga qo‘llanuvchanligini aniqlash mumkin? Ya‘ni, ixtiyoriy algoritm uchun yuqoridagi savolga javob beruvchi umumiy algoritm topilishi kerak.

3. Tyuring mashinasining o‘z-o‘ziga qo‘llanuvchanligi.

Ishni hech qaysi Tyuring mashinasi yordamida hisoblab bo‘lmaydigan funksiya qurishdan boshlaymiz.

Hisoblanuvchi bo‘lmagan funksiyaga misol. Buning uchun foydalanish mumkin bo‘lgan barcha Tyuring mashinalarini ifoda etamiz: Tyuring mashinasidagi ichki holatlarni cheksiz $q_0, q_1, q_2, \dots, q_s, \dots$ lar bilan belgilaymiz. Ushbu mashinalar majmui alfavitlari $a_0, a_1, a_2, \dots, a_s, \dots$ lar bilan belgilaymiz. Ushbu cheksiz ketma-ketliklarning barcha simvollarini standart $\{a_0, 1, q, U, CH\}$ alfavit so‘zlari bilan ifodalaymiz. Bunda quyidagi qoidalar qabul qilinadi:

q_0 q so‘zi bilan kodlansin;

q_1 qq so‘zi bilan kodlansin;

q_2 qqq so‘zi bilan kodlansin;

...

q_i qq...q (q lar $i+1$ ta) so‘zi bilan kodlansin;

va h.k.

a_1 1 so‘zi bilan kodlansin;

a_2 11 so‘zi bilan kodlansin;

...

a_i 11...1 (1 lar $i+1$ ta) so‘zi bilan kodlansin;

va h.k.

Standart alfavitda Tyuring mashinasi dasturini quyidagi qoidaga asosan SO‘Z ko‘rinishida ifodalash mumkin. Oldin dasturning barcha buyruqlari o‘chiriladi. Buning uchun $q_i a_i \rightarrow q_i a_m x, x \in \{e, Ch, O\}$ yozuvlarda « \rightarrow » belgisi tushirib qoldiriladi. q_i, a_i, a_1, a_m harflar standart alfavitning mos harflariga almashtiriladi. Bundan keyin buyruq-so‘zlar ketma-ketligi bitta so‘z shaklida yoziladi.

Shunday qilib, har bir Tyuring mashinasini qandaydir chekli standart alfavitdagi chekli so‘z bilan ifoda etish mumkin bo‘lar ekan.

Chekli alfavitdagi barcha chekli so‘zlar to‘plami sanoqli bo‘lgani uchun, Tyuring mashinalari soni ham shunga mos bo‘ladi.

Endi barcha Tyuring mashinalarini nomerlaymiz. Buning uchun turli xil Tyuring mashinalari dasturlarini ifoda etuvchi standart alfavitdagi barcha so‘zlarni fiksirlangan sanoqli cheksiz ketma-ketlik ko‘rinishida joylashtiramiz. Bunda quyidagi qoidaga rioya etiladi. Oldin barcha bir harfli so‘zlar yozib olinadi:

$\alpha_0, \alpha_1, \dots, \alpha_k$ (bu ketma-ketlik chekli buladi).

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

So‘ngra ikki harfli so‘zlar terib olinadi: $\alpha_{k+1}, \alpha_{k+2}, \dots, \alpha_1$ (bunday so‘zlar ketma-ketligi ham chekli bo‘ladi), keyin uch harfli so‘zlar keladi va h.k. Natijada barcha Tyuring mashinalari dasturlari ketma-ketligiga ega bo‘lamiz:

$$\alpha_0, \alpha_1, \dots, \alpha_1.$$

I sonini Tyuring mashinasi nomeri deb qabul qilamiz.

Endi $A=\{1\}$ alfavitda berilgan so‘zlar to‘plamidan qiymat qabul qiluvchi barcha funksiyalar to‘plamini ko‘rib chiqamiz. Boshqa tomondan, barcha mavjud bo‘lishi mumkin bo‘lgan Tyuring mashinalarini nomerlash yo‘li bilan ushbu mashinalar to‘plamini sanoqli ekanligini ko‘rsatdik. Bundan Tyuring buyicha hisoblanuvchi funksiyalar to‘plamining ham sanoqli ekanligi kelib chiqadi. Yuqorida ifodalangan funksiyalar to‘plami esa sanoqlidir. Bundan Tyuring buyicha hisoblanuvchi funksiyalarning mavjudligi kelib chiqadi. Hech bir Tyuring mashinasida hisoblanmaydigan aniq funksiya ko‘rsatsak, funksiyani hisoblovchi algoritm mavjud emasligini isbotlaydi.

$A=\{1\}$ alfavitdan olingan so‘zlar uchun berilgan φ funksiyani quramiz. Ixtiyoriy k uzunlikdagi $A=\{1\}$ alfavitdan olingan α So‘z uchun:

$$\varphi(\alpha) = \begin{cases} B_n 1, & \text{agar } A = \{1\} \text{ alfavitdagi } n \text{ nomerli TM} \\ \alpha \text{ so‘zni } B_n \text{ so‘zga aylantirsa,} \\ 1, & \text{aks holda.} \end{cases}$$

Teorema-1: $\varphi(\alpha)$ funksiya Tyuring mashinasi buyicha xisoblanuvchi emas.

Isbot: Aksini to‘g‘ri deb hisoblaylik. Ya‘ni T Tyuring mashinasi mavjud bo‘lib, uning standart alfaviti $\{a_0, 1, q, U, CH\}$ bo‘lsin va ushbu funksiyani hisoblasin. K - ushbu Tyuring mashinasining nomeri bo‘lsin. $\alpha=11\dots 1$ (1 lar soni k ta) bo‘lganda $\varphi(\alpha)=\varphi(11\dots 1)$ ga teng. Bunda so‘z nimaga teng bo‘lishini ko‘ramiz. Faraz qilaylik T mashina $11\dots 1$ so‘zni B_k so‘zga almashtirsin. Bu B_k ham $A=\{1\}$ dan olingan. Bundan $\varphi(11\dots 1)=B_k$ ekanligi kelib chiqadi.

Ikkinchi tomondan, $\varphi(\alpha)$ funksiyaning ifodasidan $\varphi(1\dots 1)=B_k 1$ ekanligi ma‘lum. Bu kelib chiqqan ziddiyat $\varphi(\alpha)$ ni hisoblovchi Tyuring mashinasi mavjud emasligini isbotlaydi.

Algoritmik yechimsizlik muammosiga yana bir misol – o‘z-o‘ziga qo‘llanuvchanlikni aniqlashdir.

Faraz qilaylik Tyuring mashinasi tasmaida uning o‘z funksional sxemasi yozilgan bo‘lsin. Agar mashina ushbu konfiguratsiyaga qo‘llanuvchan bo‘lsa, uni o‘z-o‘ziga qo‘llanuvchi deb ataymiz, aks holda esa qo‘llanilmas bo‘ladi.

Teorema-2: Tyuring mashinalari o‘z-o‘ziga qo‘llanuvchanligini aniqlash muammosi algoritmik yechimsizdir.

Isbot: Aksini faraz qilaylik. Tyuring tezisiga asoslanib, Bunday algoritmni hal qiluvchi Tyuring mashinasi mavjud deb hisoblaymiz. T – shunday Tyuring mashinasi bo‘lsin. Uning tasmaiga mos usulda kodlangan u yoki bu Tyuring mashinasining dasturi kiritiladi. Bunda agar mashina o‘z-o‘ziga qo‘llanuvchan bo‘lsa, kiritilgan so‘z mashina tomonidan o‘z-o‘ziga qo‘llanuvchanlik haqidagi savolga tasdiq javobini anglatuvchi S simvolga aylantiriladi. Mashina o‘z-o‘ziga qo‘llanilmas bo‘lsa, uning dasturini ifoda etuvchi KIRISH so‘zi yuqoridagi savolga inkor ma‘nosini anglatuvchi A simvolga aylantiriladi.

Endi T_1 Tyuring mashinasini ko‘rib o‘tsak, ushbu mashina o‘z-o‘ziga qo‘llanilmas kodlarni A ga aylantirsa, o‘z-o‘ziga qo‘llanuvchi kodlarga esa T_1 mashina qo‘llanilmas bo‘lsin. Bunday mashina T mashina yordamida quriladi, agar uning dasturi quyidagicha o‘zgartirilsa, ya‘ni S simvol paydo bo‘lgandan keyin, mashina to‘xtash o‘rniga, uni cheksiz marta takrorlasin. Demak, T_1 mashina har qanday o‘z-o‘ziga qo‘llanilmas kodga qo‘llanuvchan (A simvol xosil kilinadi), ammo o‘z-o‘ziga qo‘llanuvchan kodlarga qo‘llanilmasdir. Bu esa ziddiyatga olib keladi, chunki bunday mashina o‘z-o‘ziga qo‘llanuvchan ham, qo‘llanilmas ham bo‘la olmaydi. Ushbu ziddiyat *Teoremani isbotlaydi*.

Ushbu isbotlangan teorema ba‘zi boshqa umumiy muammolarning ham yechimsizligini isbotlaydi. Masalan, Tyuring mashinasi uchun qo‘llanuvchanlikni aniqlash

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

muammosi algoritmik yechimsizdir. U quyidagidan iborat: Qandaydir Tyuring mashinasi dasturi va konfiguratsiyasi berilgan. Ushbu konfiguratsiyaga berilgan mashina qo‘llanuvchanmi, yo‘qmi, aniqlash kerak. Bu masalani yechish algoritmi mavjud bo‘lganda, uning yordamida mashinaning o‘z dasturi kodiga qo‘llanuvchan ekanligini aniqlash mumkin bo‘lar edi. Ammo yuqoridagi teoremaga asosan, bunday algoritm mavjud emas.

Chapter 2 Getting Started

2.3.1 The divide-and-conquer approach

Many useful algorithms are *recursive* in structure: to solve a given problem, they call themselves recursively one or more times to deal with closely related subproblems. These algorithms typically follow a *divide-and-conquer* approach: they break the problem into several subproblems that are similar to the original problem but smaller in size, solve the subproblems recursively, and then combine these solutions to create a solution to the original problem.

The divide-and-conquer paradigm involves three steps at each level of the recursion:

Divide the problem into a number of subproblems that are smaller instances of the same problem.

Conquer the subproblems by solving them recursively. If the subproblem sizes are small enough, however, just solve the subproblems in a straightforward manner.

Combine the solutions to the subproblems into the solution for the original problem.

The *merge sort* algorithm closely follows the divide-and-conquer paradigm. Intuitively, it operates as follows.

Divide: Divide the n -element sequence to be sorted into two subsequences of $n/2$ elements each.

Conquer: Sort the two subsequences recursively using merge sort.

Combine: Merge the two sorted subsequences to produce the sorted answer.

The recursion “bottoms out” when the sequence to be sorted has length 1, in which case there is no work to be done, since every sequence of length 1 is already in sorted order.

The key operation of the merge sort algorithm is the merging of two sorted sequences in the “combine” step. We merge by calling an auxiliary procedure $\text{MERGE}(A, p, q, r)$, where A is an array and p , q , and r are indices into the array such that $p \leq q < r$. The procedure assumes that the subarrays $A[p..q]$ and $A[q+1..r]$ are in sorted order. It *merges* them to form a single sorted subarray that replaces the current subarray $A[p..r]$.

Our MERGE procedure takes time $\Theta(n)$, where $n = r - p + 1$ is the total number of elements being merged, and it works as follows. Returning to our card-playing motif, suppose we have two piles of cards face up on a table. Each pile is sorted, with the smallest cards on top. We wish to merge the two piles into a single sorted output pile, which is to be face down on the table. Our basic step consists of choosing the smaller of the two cards on top of the face-up piles, removing it from its pile (which exposes a new top card), and placing this card face down onto

NAZORAT UCHUN SAVOLLAR

1. Algoritmik yechimsizlik nima?
2. O‘z-o‘ziga qo‘llanuvchanlik nima?
3. Qanday algoritmik yechimsiz muammolarni bilasiz?
4. Har bir usul bo‘yicha algoritm tuzishga misol ko‘rsating.
5. Algoritmni ishlab chiqish uchun yana qanday usullarni bilasiz?

the output pile. We repeat this step until one input pile is empty, at which time we just take the remaining input pile and place it face down onto the output pile. Computationally, each basic step takes constant time, since we are comparing just the two top cards. Since we perform at most n basic steps, merging takes $\Theta(n)$ time.

The following pseudocode implements the above idea, but with an additional twist that avoids having to check whether either pile is empty in each basic step. We place on the bottom of each pile a *sentinel* card, which contains a special value that we use to simplify our code. Here, we use ∞ as the sentinel value, so that whenever a card with ∞ is exposed, it cannot be the smaller card unless both piles have their sentinel cards exposed. But once that happens, all the nonsentinel cards have already been placed onto the output pile. Since we know in advance that exactly $r - p + 1$ cards will be placed onto the output pile, we can stop once we have performed that many basic steps.

MERGE(A, p, q, r)

```

1   $n_1 = q - p + 1$ 
2   $n_2 = r - q$ 
3  let  $L[1..n_1 + 1]$  and  $R[1..n_2 + 1]$  be new arrays
4  for  $i = 1$  to  $n_1$ 
5       $L[i] = A[p + i - 1]$ 
6  for  $j = 1$  to  $n_2$ 
7       $R[j] = A[q + j]$ 
8   $L[n_1 + 1] = \infty$ 
9   $R[n_2 + 1] = \infty$ 
10  $i = 1$ 
11  $j = 1$ 
12 for  $k = p$  to  $r$ 
13     if  $L[i] \leq R[j]$ 
14          $A[k] = L[i]$ 
15          $i = i + 1$ 
16     else  $A[k] = R[j]$ 
17          $j = j + 1$ 

```

In detail, the MERGE procedure works as follows. Line 1 computes the length n_1 of the subarray $A[p..q]$, and line 2 computes the length n_2 of the subarray $A[q + 1..r]$. We create arrays L and R (“left” and “right”), of lengths $n_1 + 1$ and $n_2 + 1$, respectively, in line 3; the extra position in each array will hold the sentinel. The for loop of lines 4–5 copies the subarray $A[p..q]$ into $L[1..n_1]$, and the for loop of lines 6–7 copies the subarray $A[q + 1..r]$ into $R[1..n_2]$. Lines 8–9 put the sentinels at the ends of the arrays L and R . Lines 10–17, illus-

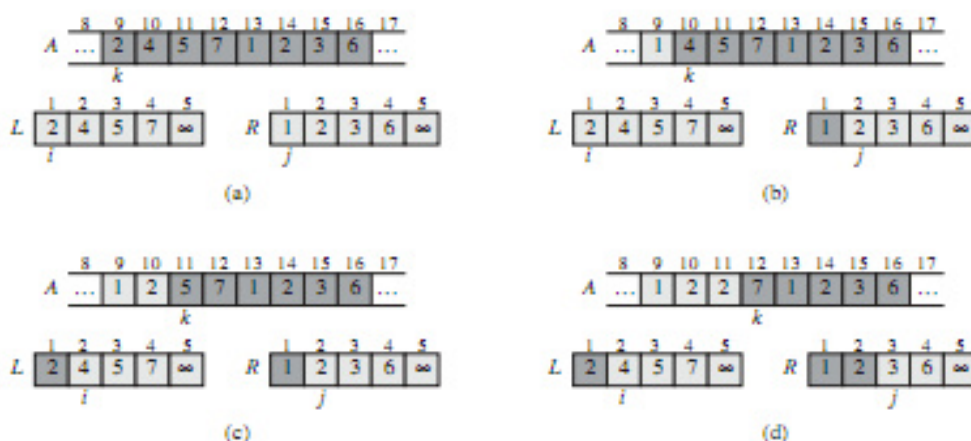


Figure 2.3 The operation of lines 10–17 in the call $\text{MERGE}(A, 9, 12, 16)$, when the subarray $A[9..16]$ contains the sequence (2, 4, 5, 7, 1, 2, 3, 6). After copying and inserting sentinels, the array L contains (2, 4, 5, 7, ∞), and the array R contains (1, 2, 3, 6, ∞). Lightly shaded positions in A contain their final values, and lightly shaded positions in L and R contain values that have yet to be copied back into A . Taken together, the lightly shaded positions always comprise the values originally in $A[9..16]$, along with the two sentinels. Heavily shaded positions in A contain values that will be copied over, and heavily shaded positions in L and R contain values that have already been copied back into A . (a)–(h) The arrays A , L , and R , and their respective indices k , i , and j prior to each iteration of the loop of lines 12–17.

trated in Figure 2.3, perform the $r - p + 1$ basic steps by maintaining the following loop invariant:

At the start of each iteration of the for loop of lines 12–17, the subarray $A[p..k-1]$ contains the $k - p$ smallest elements of $L[1..n_1 + 1]$ and $R[1..n_2 + 1]$, in sorted order. Moreover, $L[i]$ and $R[j]$ are the smallest elements of their arrays that have not been copied back into A .

We must show that this loop invariant holds prior to the first iteration of the for loop of lines 12–17, that each iteration of the loop maintains the invariant, and that the invariant provides a useful property to show correctness when the loop terminates.

Initialization: Prior to the first iteration of the loop, we have $k = p$, so that the subarray $A[p..k-1]$ is empty. This empty subarray contains the $k - p = 0$ smallest elements of L and R , and since $i = j = 1$, both $L[i]$ and $R[j]$ are the smallest elements of their arrays that have not been copied back into A .

FOYDALANILGAN ADABIYOTLAR RO‘YHATI:

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Сер: Классические учебники: COMPUTER SCIENCE. М.: МЦНМО, –960с., 2004.
2. Вирт Н. Алгоритмы и структуры данных. С примерами на Паскале. Санкт-Петербург, 352с., 2005.

³ Thomas H. Cormen. Intruduction to algorithms. Massachusetts Institute of Technology. London 2009. (29-31 pp)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA
2 MODUL. PASKAL DASTURLASH TILINING IMKONIYATLARI.
MA’RUZA № 5

MAVZU: PASKAL TILI DASTURLASH TILINING ALIFBOSI, BUYRUQLAR
TIZIMI VA OPERATORLARI.

REJA:

1. Dasturlash tillari va ularning klassifikatsiyasi, mashinaga mo‘ljallangan va proseduraga mo‘ljallangan dasturlash tillari, yuqori darajali dasturlash tillari.
2. Turbo Paskal va Paskal ABC dasturlash muhitini taqqoslash.
3. Paskal dasturlash tili strukturasi va alifbosi.
4. Paskal tilida o‘zgaruvchilarni tavsiflash.
5. Paskal tilining standart funksiyalari va kalit so‘zlari
6. Qo‘zg‘aluvchan va qo‘zg‘almas nuqtali o‘zgarmlar.
7. Foydalanuvchining kutubxona modullari.

Tayanch so‘z va iboralar: Pascal ABC dasturlash tili, o‘zgaruvchi, standart funksiyalar, o‘zgarmlar, biblioteka modullari.

1. Dasturlash tillari va ularning klassifikatsiyasi, mashinaga mo‘ljallangan va proseduraga mo‘ljallangan dasturlash tillari, yuqori darajali dasturlash tillari

Dasturlash tillari paydo bo‘lishidan oldin dasturlar *mashina kodlarida* (mashina kodi - protsessor tomonidan bajariladigan xotiradagi instruksiyalar ketma-ketligidir) tuzilar edi. Katta-katta dasturlarni tuzishda juda ko‘p vaqt talab qilinadi, ularni xatolarini tuzatish juda qiyin, modifikatsiyalash esa ko‘p hollarda ilojsiz edi. Shularni hisobga olib, inson uchun tushinarli bo‘lgan dasturlash tillarini tashkil qilish muammosi paydo bo‘ldi. Dasturlash tillari inson uchun tushinarli bo‘lgan dasturlarni tuzish imkonini beradi. Bunday dasturlarning matnini mashinalarlarda bajarish uchun, ularni mashina kodiga aylantirish zarur. Buning uchun *traslyator* deb ataluvchi maxsus dasturlardan foydalaniladi.

Traslyatorlar ikki xil ko‘rinishda bo‘ladi: *interpretator* va *kompilyator*.

Interpretator - dasturning har bir operatorini oraliq kodga tarjima qilib, mashina kodiga aylantiradi va uni bajarishga kirishadi. Kompilyator - dastur matnini to‘laligicha mashina kodiga aylantirib, uni bajarishga kirishadi.

Birinchi *dasturlash tili* - assembler bo‘lib, bu til past pogonadagi til turiga kiradi. Dasturning har bir qatori-bitta mashina komandasiga mos tushadi. Assembler tilida katta-katta dasturlarni yozish juda qiyin, shuning uchun keyinchalik yuqori pogonadagi dasturlar yaratildi. Bularga Beysik, Pascal, Fortran, Ci va hokozolar kiradi. Pascal - 1969 yilda Syurix texnika universiteti professori N.Virt tomonidan yozilgan bo‘lib, talabalarga dasturlar tuzishni o‘rgatishga mo‘ljallangan.

Satr holati aniq harakatlarni qanday tugmachalar yoki qatorlar yordamida bajarish mumkin ekanligini bildiradi. Ekrandagi kalit qatori sichqoncha yordamida boshqariladi. Berilgan komandani bajarish uchun sichqoncha shu satrga o‘rnatilib, chap tugmacha bosiladi. Masalan, asosiy menyuga o‘tish uchun F10 yoki “F10 Menu” qatoriga sichqoncha ko‘rsatgichi o‘rnatilib, tugmacha bosiladi.

Turbo-Pascal tahrirlagichi bir nechta fayllarni bir vaqtning o‘zida tahrirlash imkoniyatini beradi. Tahrirlagichga kirish uchun "File" menyusida "New" komandasini tanlash kifoya qiladi. Ekranida NONAME00.PAS nomli oyna paydo bo‘ladi. Kursor oynani chap tepa burchagida o‘rnatiladi. Bir vaqtning o‘zida bir nechta tahrirlagich oynasini ochish uchun "New" komandasi bir necha marotaba bajariladi.

Tahrirlagich komandalari WD matn tahrirlagichi komandalari kabidir.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

1-Jadval Kursorni siljitish komandalari

Tugmacha	Harakat
←, →	Bitta belgi chapga yoki o‘ngga
↓, ↑	Bitta satr tepaga yoki pastga
Home	Satr boshiga
End	Satr oxiriga
PgUp, PgDn	Bitta bet tepada yoki pastga
Ctrl- ←	Bitta so‘z chapga
Ctrl- →	Bitta so‘z o‘ngga
Ctrl-Home	Oyna boshiga
Ctrl-End	Oyna oxiriga
Ctrl-PgUp	Fayl boshiga
Ctrl-PgDn	Fayl oxiriga
Ctrl-Q-B	Blok boshiga

Tahrirlagichga matnni olib kirish ikki xil rejimlarda bajariladi: yozuvga qo‘shish (insert) va qayta yozish (overwrite). Rejimlar almashinishi INS tugmacha yordamida amalga oshiriladi. Rejim almashinuvi ro‘y bergan paytda kursor holati o‘zgaradi. Qolgan komandalar guruhi tayyor matnlarni korrektirovka qilish uchun mo‘ljallangan. Komandalar ro‘yxati 2- jadvalda berilgan.

2- jadval. Ko‘shish va o‘chirish komandalari

Tugmacha	Harakat
Ins	O‘chirish/yoqish qo‘shish rejimi
Del	Kursor holatida belgi o‘chirish
Backspace	Kursordan chapdagi belgini o‘chirish
Ctrl-Y	Qatorni o‘chirish
Ctrl-Q-Y	Kursor turgan joydan qator oxirigacha bo‘lgan belgilarni o‘chirish
Ctrl-N	Qatorni qo‘shish

Bloklar bilan ishlash.

Blok bilan ishlashdan oldin matnda blokni ajratib olish kerak. Bu quyidagicha amalga oshiriladi: Kursorni ajratish kerak bo‘lgan matni boshiga quyib CTRL-K-B ni birgalikda bosiladi, so‘ngra kursorni blok oxiriga o‘rnatib CTRL-K-K bosiladi. Keyin ekranda ajratilgan blok kerakli rangda paydo bo‘ladi. Bloklarni ajratish uchun SHIFT-↑, ↓, ←, → yo‘naltiruvchi tugmachalardan yoki bo‘lmasa sichqonchadan foydalanish mumkin. Bloklar bilan ishlash 3- jadvalda berilgan.

3- jadval. Bloklar bilan ishlash komandalari

Tugmacha	Harakat
Ctrl-K-C, Shift-Ins	Blokdan nusxa olish
Ctrl-K-Y, Ctrl-Del	Blokni o‘chirish
Ctrl-K-V, Shift-Del	Blokni surib quyish
Ctrl-K-H	Blokni rangini ajratish
Ctrl-K-P	Blokni chop etish
Ctrl-K-R	Blokni diskdan o‘qish
Ctrl-K-W	Blokni diskga yozish

Bloklarni tahrirlashda *Clipboard* nomli qo‘shimcha komanda mavjud.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Bu komanda tahrirlash rejimida bir nechta oynalar bilan ishlash imkoniyatini beradi.

Masalan, 1- oynadan 2- oynaga blok matnidani nusxa ko‘chirish uchun quyidagi harakatlar ketma-ketligi bajariladi: 1- oynada blokni ajratib, CTRL-INS bosiladi, so‘ngra ALT-2 yordamida 2- oynaga o‘tib, blokdan nusxa olish uchun SHIFT-INS bosiladi.

Bundan tashqari bloklarni tahrirlashda bosh menyudagi "Edit" maxsus komandasidan foydalanish mumkin. Bu quyidagicha bajariladi: "Edit" komandasi tanlangandan so‘ng ekranda, misolda ko‘rsatilganlarni bajarish uchun maxsus menyu paydo bo‘ladi.

"Cut" qirqib olib tashlash komandasi o‘zining harakatlari bo‘yicha "Copy" komandasi kabidir, lekin farqi shundaki Clipboard ga blokdan nusxa olinganda blok joriy oynadan o‘chib ketadi. Blokni joriy oynada asl holatiga takrorlash uchun SHIFT-INS bosiladi. Standart qadamlar ketma-ketligi: kiritish, saqlash, bajarish, tahrirlash - har qanday dastur tuzishni umumiy ssenariysini tashkil qiladi.

Tashkil qilish:

Kompilyator ishga tushgandan so‘ng ekranda tahrirlagichni standart oynasi paydo bo‘lmasa, yangi fayl tashkil qilish uchun "File" menyusidagi "New" komandasi yoki ALT-F-N bajarilishi kerak. So‘ngra tanish komandalar orqali joriy matnni dasturga kiritish lozim.

Saqlash:

Joriy matnni saqlash uchun "File" menyusidagi "Save" komandasidan foydalaniladi yoki F2 bosiladi. Ekranda fayl nomini so‘rovchi oyna hosil bo‘ladi. Nom kiritilgandan so‘ng komandani bajarish uchun ENTER bosiladi va joriy katalogda shu nomli yangi fayl paydo bo‘ladi.

Kompilyatsiya:

Dasturni saqlab quyilgandan so‘ng kompilyatsiyalashga o‘tish mumkin. Kompilyatsiyalashda matndagi sintaksis xatolar va boshqa xatolar tekshiriladi.

Buning uchun ALT-F9 bosiladi yoki asosiy menyuning ("Compile") opsiyasiga o‘tib, "Compile" komandasi bajariladi. So‘ngra kompilyatsiyani bajarilishi haqida qo‘shimcha ma’lumotni o‘z ichiga olgan ("Compiling") oynasi paydo bo‘ladi.

Agarda dastur matnida xatolik topilsa, u holda birinchi qator oynasida xatolik va xatolikni kelib chiqish sabablari haqida ma’lumot paydo bo‘ladi. Bu holatda kursor avtomatik ravishda xatolik bor qatorga yoki undan keyingi qatorga o‘tib qoladi.

So‘ngra dastur matnidagi xatoliklarni tuzatib, yana bir marta "Compile" komandasini bajarish lozim. Agarda boshqa xatolik topilmasa, u holda kompilyator bajariluvchi (executable) fayl obrazini yaratadi. "Compile" menyusining "Destination" komandasi bajariluvchi faylni qayerda saqlashni: xotirada (memory) yoki diskda (disk) EXE fayli ko‘rinishida saqlashni aniqlashga yordam beradi. O‘zgartirilgan dastur matnini F2 bilan saqlab, so‘ngra dasturni bajarishga o‘tiladi.

Bajarish:

Dasturni bajarish uchun CTRL-F9 bosiladi yoki asosiy menyuning ("Run") opsiyasiga o‘tib, ("Run") komandasi chaqiriladi. "Run" komandasi bajarilishidan oldin avtomatik ravishda dastur kompilyatsiyalanganmi yoki yo‘qligi tekshiriladi, agarda kompilyatsiyalanmagan bo‘lsa kompilyatsiyalanib, so‘nga dastur bajariladi.

Dastur ishlash paytida IDE tasviri, kompilyator ishga tushmasdan oldingi DOS ekraniga o‘rnini bo‘shatib beradi. Dastur ishi tugagandan so‘ng boshqarish yana IDE ga uzatiladi.

Natijalarni ko‘rib chiqish:

Dastur natijalarini ko‘rib chiqish uchun "Window" menyusidagi ("Output") komandasidan foydalaniladi. ("Output")ga kursorni olib kelib ENTER ni bosamiz, ekranda mos tartib raqamli oyna paydo bo‘ladi. Ekranni to‘la ko‘rib chiqish uchun "Output" oynasini ekranga to‘la yoyib chiqishni o‘zi yetarli bo‘ladi. Xuddi shu imkoniyatni qo‘shimcha "Window" menyusining ("User screen") komandasi ham bajarishi mumkin.

Dasturga qayta murojat qilish:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Diskga yozilgan biror bir dasturni ochish uchun "File" (F3) menyusidagi ("Open") komandasidan foydalaniladi. Komanda bajarilganda joriy katalogdagi barcha fayllar ro‘yxati ekranga chiqadi va oddiy klaviatura yoki sichqoncha yordamida kerakli faylni tanlab olish mumkin. Qo‘shimcha ravishda ish tugagandan so‘ng, IDE da avtomatlashtirilgan holatda diskda barcha tahrirlanuvchi fayllar ro‘yhatini saqlab quyish imkoniyati mavjud. Agada shu narsa sodir bo‘lgan taqdirda keyingi ishdan oldin sizning oxirgi dasturingiz avtomatik tarzda xotiraga yozilib qoladi.

Integrallashgan muhit bilan tanishuv

Integrallashgan muhit, ko‘poynali matnli redaktor, bosh menyu, sistemali menyu, fayllar bilan ishlash, ko‘poynali redaktorlash, matnda izlash va almashshtirish, dasturlarni bajarish, dasturni kompilyatsiyalash, dastur xatolarini tuzatish, rejimlarni o‘rnatish, qatorlar holati, "tezkor" tugmachalar, redaktorlash komandalari, Clipboard.

O‘zidan oldigi komilyatorlardan farqliroq Turbo Pascal kopilyatori intellektual (the integrated development environment - IDE) *integrallashgan muhitni* o‘z ichiga oladi. Kompilyatorning asosiy xususiyatlaridan biri, ko‘p oynali sistema, *ko‘p oynali matnli redaktor*, dialoglar oynasi (dialogs) va turli menyular va xokozolardir.

Kompilyator ishga tushgandan so‘ng ekranda asosiy oyna paydo bo‘ladi. Oynaning yuqori qatori gorizontol yo‘laklar (the menu bar) ko‘rinishidagi menyuga ajratiladi. Bunga Turbo Pascal kompilyatorining *bosh menyusi* deyiladi. Uning yordamida quyidagi komandalarga osongina murojat etish mumkin: *sistemali menyu(YO)*, *fayllar bilan ishlash ("File")*, *ko‘p oynali redaktorlash ("Edit")*, *matnda izlash va almashshtirish ("Search")*, *dasturlarni bajarish ("Run")*, *dasturni kompilyatsiyalash ("Compile")*, *dastur xatolarini to‘g‘rilash ("Debug")*, *rejimlarni o‘rnatish ("Options")*, oynalar bilan manipulyatsiyalar ("Window") va yordam ("Help"). Oynaning pastki qatori (the status line) *holatlar qatoriga* ajratilgan bo‘lib, u yerda funksional tugmachalarning vazifalari ko‘rsatilgan: F1 - Help, F2 - Save, F3 - Open, ALT-F9 - Compile, F9 - Make, F10 - Menu. Oynaning qolgan qismi maxsus belgilar bilan to‘ldirilgan va IDE (the desktop) ishchi oblastini tashkil etadi.

Asosiy menyuning komandalariga uchta usuldan birortasi yordamida murojat etish mumkin: F10 ni bosib, menyu ichiga kirib, kerakli komandani tanlash mumkin. Tanlangan komanda ekranda mos rangda paydo bo‘ladi. Uni bajarish uchun esa ENTER ni bosish kerak. Bundan tashqari menyuni boshqarish uchun sichqonchadan foydalanish mumkin. Va oxirida, asosiy menyudan biror-bir komandani tanlash uchun *"tezkor" tugmacha* (hotkeys) lardan foydalanish mumkin.

Har bir komandaning kalit so‘zlariga qoida bo‘yicha, bosh, bitta liter ajratilgan. Ajratilgan literlarning birortasi bilan ALT ni birgalikda bosib, mos komandani bajarishga kirishish mumkin. Masalan, "Compile" komandasini tanlash uchun ALT-C ni bosish kifoya, sistemali menyuga o‘tish uchun esa, ALT-SPACEBAR bosiladi. Siz o‘zingiz uchun qulay bo‘lgan variantni tanlab olishingiz mumkin.

2. Turbo Paskal va Paskal ABC dasturlash muhitini taqqoslash

Paskal dasturlash tili o‘zining soddaligi, mantiqiyiligi va samaraligi tufayli bu til butun dunyoga tez tarqaldi. Hozirgi paytda barcha hisoblash mashinalari, hususan mikroEHMLar ham shu tilda ishlash imkoniga ega. Dasturlar matnining to‘g‘riligini osonlik bilan tekshirish mumkinligini, ularning ma‘nosi yaqqol ko‘zga tashlanishi va oddiyligi bilan ajralib turadi.

Paskal tili ancha murakkab va ko‘p vaqt oladigan hisoblash ishlarini bajarishga mo‘ljallangan tarkiblashtirilgan dasturlar tuzishga imkon beradi. Yana bir afzalligi shundan iboratki foydalanuvchi xattolikka yo‘l quymasligi uchun yoki xatto yozib quygan bo‘lsa, tez tuzatib olishi uchun dasturda ishlatilgan o‘garuvchilar oldindan qaysi turga mansub ekanligi belgilab qo‘yilgan bo‘ladi. Shu bilan birga dasturning barcha elementlari haqida ma‘lumot tavsiflash bo‘limida mujassamlashgan bo‘ladi. Operatorlar soni esa, minimal darajada kamaytirilgandir.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Algoritmik tillarning mashina tillaridan asosiy farqlari sifatida quyidagilarni ko‘rsatish mumkin:

- mashina tili alifbosidan algoritmik til alifbosining o‘ta kengligi;
- tuzilgan dastur matinning ko‘rinish sifatini keskin oshiradi;
- ishlatilishi mumkin bo‘lgan amallar majmui mashina amallari majmuiga bog‘liq emas;
- bajariladigan amallar odam uchun qulay ko‘rinishda, ya‘ni amalda qabul qilingan matematik belgilashlarda beriladi;
- amallar operandlari uchun dasturchi tomonidan beriladigan shaxsiy ismlar qo‘yish mumkinligi;
- mashina uchun ko‘zda tutilgan ma’lumot tiplaridan tashqari yangi tiplar kiritish imkoniyati yaratilganligi.

Shunday qilib, ma’lum ma’noda aytish mumkinki, algoritmik tillar mashina tiliga bog‘liq emas.

Yo‘qorida aytilganlardan kelib chiqqan holda ma’lum bo‘ldiki algoritmik tilda yozilgan masala yechimining algoritmi to‘g‘ridan to‘g‘ri EHM da bajarilishi mumkin emas ekan. Buning uchun esa algoritm oldindan ishlayotgan EHM ning mashina tiliga translyator (kompilyator yoki interpretator) yordamida o‘g‘irilishi lozim.

Til alifbosi shu tilgagina tegishli bo‘lgan chekli sondagi belbilardan tashkil topgan. Dastur matnini yozishda faqat shu belgilardan foydalanish mumkin, boshqa belgilarni esa til tanimaydi yani ularda foydalanish mumkin emas.

Til sintaksisi alfovit harflaridan tashkil topgan bo‘lib, mumkin bo‘lgan konstruksiyalarni aniqlovchi qoidalar tizimidir. Mazkur tilda ifoda etilgan to‘la algoritm va uning alohida hadlari shu konstruksiyalar orqali ifoda qilinadi. Shunday qilib, belgilarning har qanday ketma-ketligini, hamda mazkur tilning matni to‘g‘riligi yoki noto‘g‘riligi til sintaksisi orqali bilib olamiz.

Til semantikasi algoritmik tilning ayrim konstruksiyalari uchun qoidalar tizimini tushuntirishga xizmat qiladi.

1981 – yilda Paskal tilining xalqaro standarti taklif etildi. Paskal tili Borland firmasi tomonidan yaratilgan.

Integrallashgan muhit - dasturlashga yordamlashuvchi dastur bo‘lib, quyidagi vazifalarni o‘z ichiga oladi:

- vazifalari dastur matnini kiritish imkonini beradi;
- kiritilayotgan dastur matnini tashqi xotirada saqlab turadi;
- uni ishga tushirishi uchun translyatori bor;
- sintaktik xatolarni momentalna aniqlaydi.

Mazkur katalogda TURBO.EXE fayliga murojaat qilingandan so‘ng ekranda Turbo-Paskal muhitining o‘z menyusu satriga ega bo‘lgan tahrir qilish sahifasi ochiladi.

Turbo – Paskal integrallashgan muhiti interfeysida *menyular satri*, *ishchi maydoni*, *ma’lumot satri* mavjud.



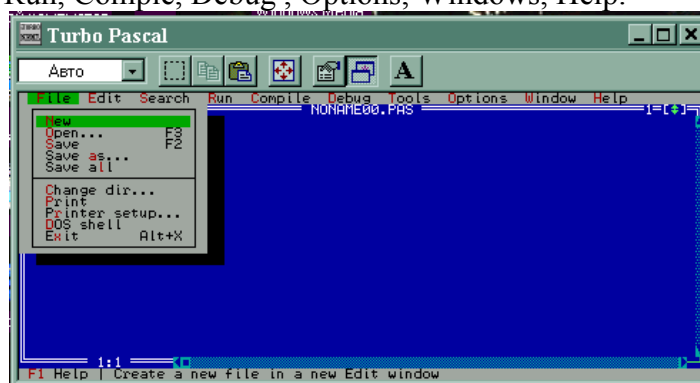
1 – rasm.

1 – rasmda Paskal dasturining oynasi ko‘rinib turibdi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Menyu satrida alohida vazifalariga ega bo‘lgan 9 ta bo‘limlar mavjud.

File, Edit, Search, Run, Comple, Debug , Options, Windows, Help.



2 – rasm.

Har bo‘lim o‘z bandlariga ega bo‘lib, ularning ichida ... belgi bilan tugaganlari alohida muloqot darchalariga ega bo‘ladilar.

File bo‘limiga murojaat etilganda hosil bo‘lgan majmuada Open... F3(2 - rasm) bandi kompyuter xotirasidan Paskal fayllarini ekranga chaqirish uchun mo‘ljallangan. Mazkur band faollashtirilganda muloqot darchasi hosil bo‘lib, u yerda kerakli fayl katalog ichidan axtariladi.

New yordamida yangi dastur matnini kiritish uchun oyna ochiladi.

Save F2 dasturni xotiraga kiritadi.

Save as... dasturni biror nom ostida xotiraga kiritadi.

Save all barcha fayllarni xotiraga kiritadi.

Change dir... yangi katalog hosil qiladi.

Print dastur matnini chop etadi.

Printer setup... dastur hisoblashi davomida kompyuter imkoniyatlaridan foydalanish darajasi haqida ma’lumot beradi.

Dos shell dasturdan vaqtinchalik operatsion tizimga chiqib turish imkoniyatini yaratadi.

Exit (Alt + x) NC ga chiqiladi.

Edit Paskal dasturlarini tahrir qilish vazifasini bajaradi. Tahrir qilish davrida belgilangan bo‘laklar ustida amal bajarish uchun klaviaturadagi tugmalarning quyidagi majmuasidan foydalanishimiz mumkin:

Ctrl+K+B -ajratiluvchi bo‘lakning boshini belgilash;

Ctrl+K+K - ajratiluvchi bo‘lakning oxirini belgilash;

Ctrl+K+C – belgilangan bo‘lakning nusxasini olish;

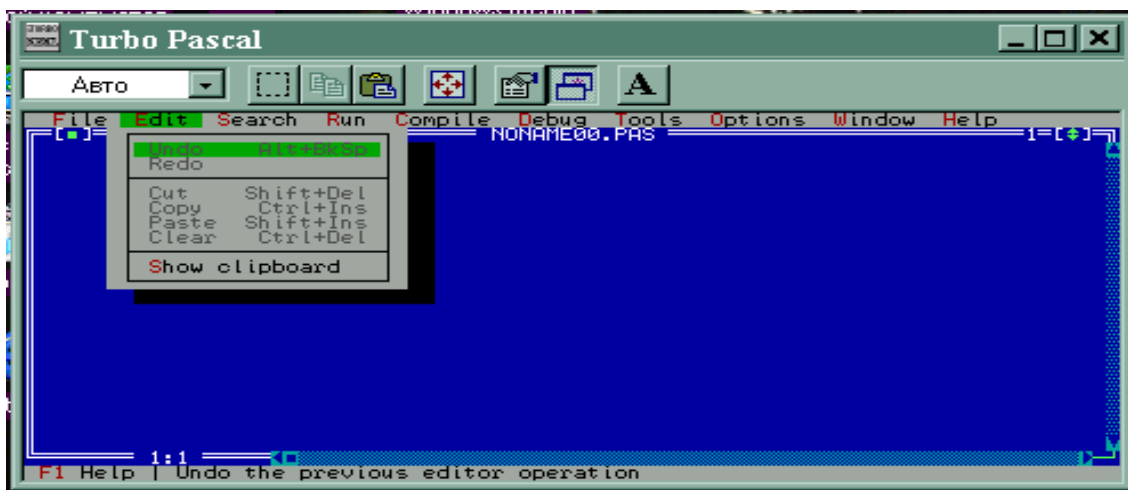
Ctrl+K+V – belgilangan bo‘lakni boshqa joyga ko‘chirish;

Ctrl+K+Y – belgilangan bo‘lakni o‘chirish;

Ctrl+K+P – belgilangan bo‘lakni chop etish;

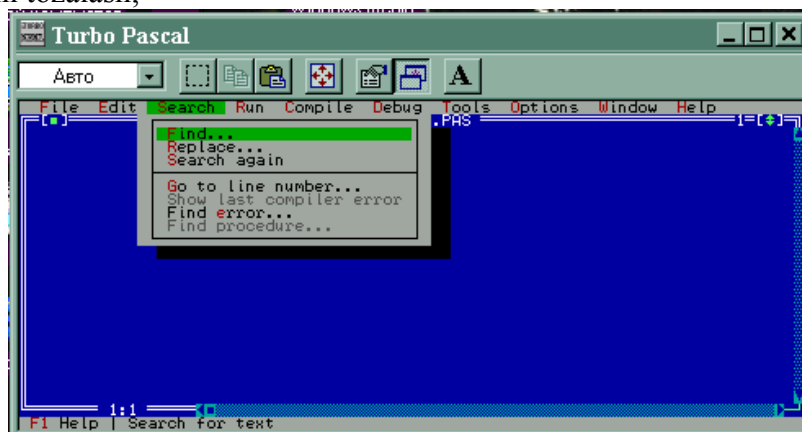
Ctrl+K+H – belgilash amalini bekor qilish;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



3 – rasm.

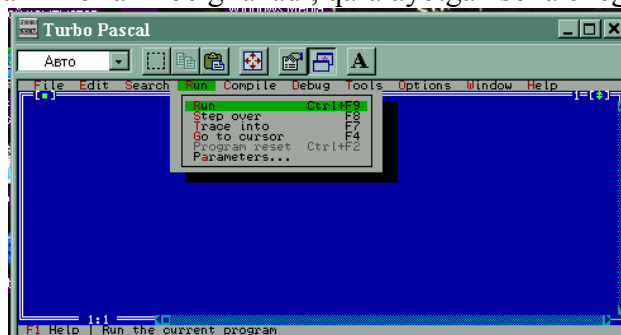
Undo - belgilangan bo‘lakni buferda saqlanishi;
Cut (Shift +del)belgilangan bo‘lakni olib tashlash;
Copy - xotiraga bo‘lakning nusxasini o‘tkazish;
Paste - bo‘lak nusxasini dasturda hosil qilish;
Show clipboard - almashish buferi mazmunini ko‘rish;
Clean - sahifani tozalash;



4 – rasm.

Search bo‘limi belgi va so‘zlarni axtarish va almashtirish vazifalarini bajaradi:
Find - dasturda belgi va so‘zni axtarish;
Replace - topilgan belgini o‘zgartirish;
Search again - amalni yangidan bajarish;
Goto line number - raqami ko‘rsatilgan qatorga o‘tish;
Find protsedure - kichik dasturni axtarish;
Find error - hisoblash xatoliklarini aniqlash.

Mazkur bo‘limning bandlariga murojaat qilinganda muloqot darchasi hosil bo‘lib, u yerda bajarilayotgan vazifalarni ko‘lami belgilanadi, qaralayotgan soha chegaralanadi.



“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Run bo‘limida tahrir qilingan dasturni hisobga o‘tkazish bandlari jamlangan:

Run - dasturni hisobga o‘tkazish;

Program reset – tahrir qilishni to‘xtatish;

Goto cursor - kursor turgan joygacha hisoblash;

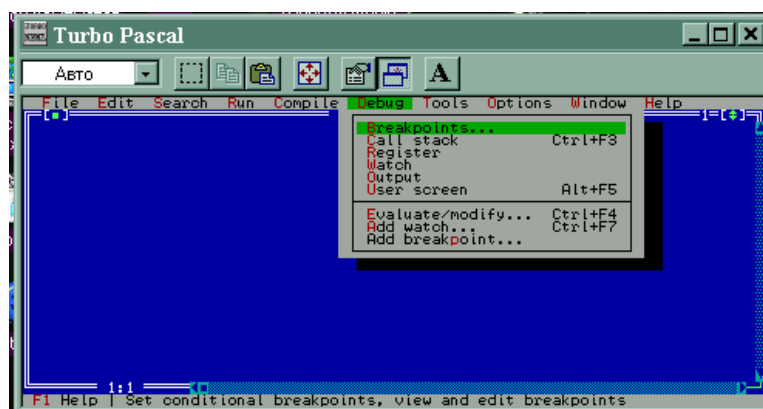
Trace into - hisoblash algoritmini ko‘rish;

Step over - satrlab hisoblash;

Params - dastur parametrlarini aniqlash.

Trace unto bandi dasturni belgilangan algoritm bo‘yicha qadamlab hisoblaydi, natijada mavjud kamchiliklarni aniqlash osonlashadi.

Step over bandi yuqoridagi bandga o‘xshash vazifani amalga oshirsa-da, hisoblash davomida



Debug bo‘limida 4 ta band bo‘lib, ularning har biri dastur hisoblashida yuzaga keluvchi xatoliklarni aniqlashni osonlashtirish vazifasini bajaradi.

Evaluate/modify-o‘zgaruvchi qiymatlarini baholash;

Watches - to‘xtash joyi va qiymatni ko‘rish;

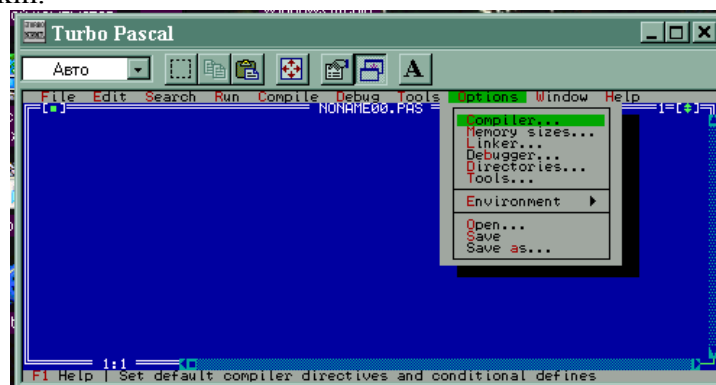
Toggle breakpoint - to‘xtash satrini tanlash;

Breakpoints - to‘xtash nuqtasi amallari.

Evaluate/modify ... bandi dastur hisoblashida oraliq o‘zgaruvchilar qabul qilgan qiymatlarni ko‘rish uchun mo‘ljallangan bo‘lib, murojaat etilganda ekranda muloqot darchasi hosil bo‘ladi. Mazkur darchaning birinchi satrida qaralayotgan o‘zgaruvchi yoziladi va keyingi qatorda uning joriy qiymati hosil bo‘ladi.

Watches bandi muloqotli darchasi quyidagi ko‘rinishda bo‘ladi:

Mazkur bandning satrlaridan foydalanib satrlab hisoblash usulida kerakli o‘zgaruvchining qabul qilayotgan qiymatlari uzluksiz kuzatib turiladi. Zarur bo‘lganda ifodalar sohasini tahrir qilish mumkin.



Options bo‘lim Turbo-Paskal muhiti ayrim xossalarini boshqarish uchun mo‘ljallangan:
Compiler - kompilyator;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Memory sizes - xotira hajmi;

Linker - moslashtiruvchi (kompanovshik);

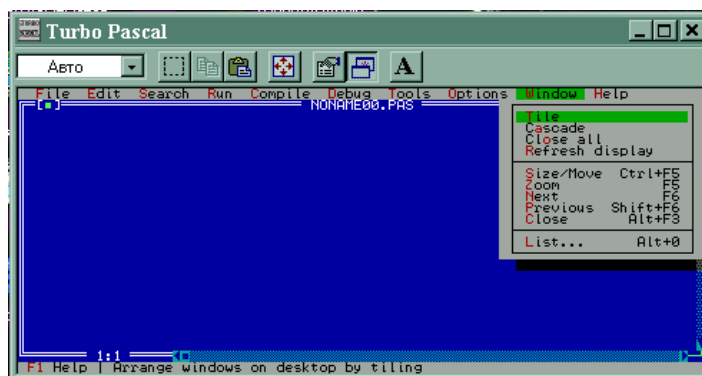
Directories - jadvallar;

Environment - faoliyat sharti;

Save options - opsiyalarni diskka yozish;

Retrieve options - opsiyalarni diskdan o‘qish.

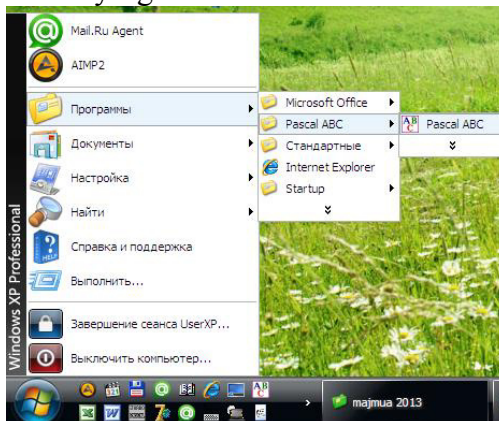
Compiler bandi muloqot darchasi yordamida hisoblash paytida qiymatlar o‘zgarishi oraliqlari, kiritish va chiqarish nazorati, matematik soprotsessorni qo‘shish va shu kabi vazifalarni amalga oshirish mumkin. Bunda [] ichida bo‘lishi kerak.



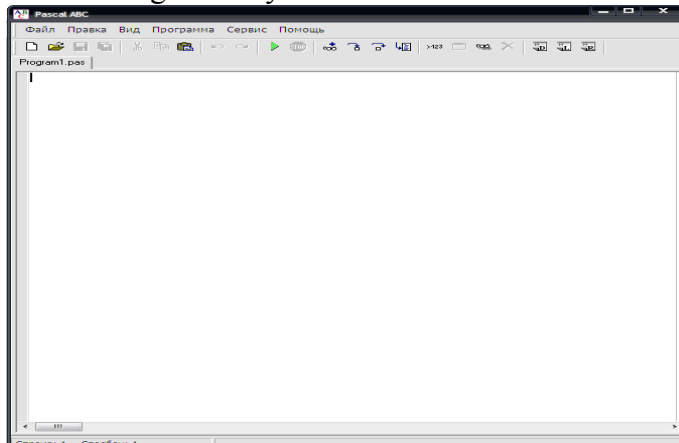
Turbo-Paskal menyusining navbatdagi bo‘limlari kopyuter ekranidan natija olishni maqbullashtirish va tizim haqida kerakli ma’lumotlarni tavsiya qilish vazifalarini bajaradi.

PaskalABC muhitiga kirish uchun quyidagi bosqichlarni amalga oshiramiz

Пуск-программы-PascalABC buyrug‘i tanlanadi

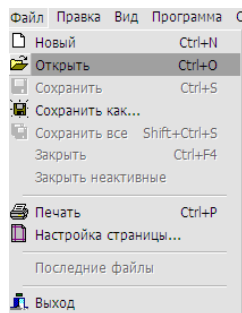


Paskal dasturlash tili muhitining umumiy ko‘rinishi

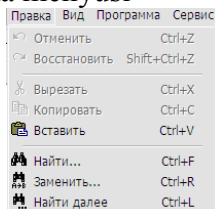


1. Paskal ABC muhitining fayl menyusi

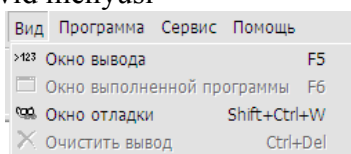
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



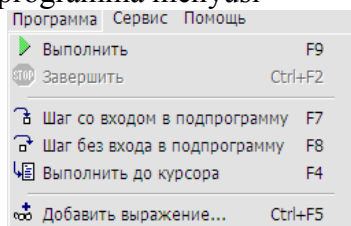
2. Paskal ABC muhitining pravka menyusi



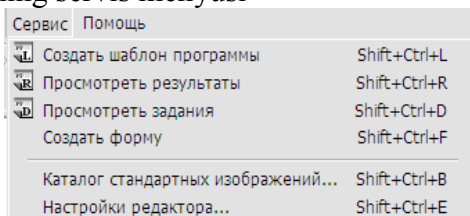
3. Paskal ABC muhitining vid menyusi



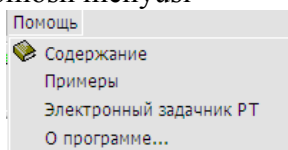
4. Paskal ABC muhitining programma menyusi



5. Paskal ABC muhitining servis menyusi



6. Paskal ABC muhitining pomosh menyusi



3. Paskal dasturlash tili strukturasi va alifbosi

Paskal tili alfaviti

Tanlab olingan algoritm asosida qo‘yilgan masalani kompyuterda yechish uchun qanday algoritmik tilning o‘z alifbosi, buyruqlar majmuasi va maxsus dasturlar jamlangan kutubxonasi bo‘lishi zarur. Paskal tili alifbosini uch qismga bo‘lish mumkin:

Paskal tili alifbosiga quyidagilar kiradi:

1. 26 ta lotin alifbosi harflari: A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,S,T, U,V,W,X,Y,Z va rus alifbosi harflari.

2. Arab raqamlari: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0. Nol soni O harfidan farq qilishi uchun dastur tuzishda uning ustiga chizib yoziladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

3. Arifmetik amallar. Paskal tilida quyidagi arifmetik amal belgilari mavjud: ko‘paytirish (*), masalan: A*B; bo‘lish (/), masalan: A/B; qo‘shish (+), masalan: A+B; ayirish (-), masalan: A-B. Paskal tilida darajaga ko‘tarish amali yo‘q. Shuning uchun ham sonlarni butun darajaga ko‘tarish (daraja ko‘rsatkichi katta son bo‘lmasa) ularni bir necha marotaba ko‘paytirish yo‘li bilan amalgam oshirish mumkin. Haqiqiy darajaga ko‘tarish (agar asos musbat son bo‘lsa) logarifmlash yo‘li bilan amalgam oshiriladi.

$$x^n = e^{n \ln x} \quad \text{yoki} \quad x^n = 10^{n \lg x}$$

4. Munosabat amal belgilari:

Paskal belgisi	Matematik ko‘rinishi	Ma‘nosi
=	=	Teng
<>	≠	Teng emas
<	<	Kichik
<=	≤	Kichik yoki teng
>	>	Katta
>=	≥	Katta yoki teng

5. Maxsus belgilar: . (nuqta); , (vergul); ; (nuqtali vergul); : (ikki nuqta); oddiy, kvadrat va figurali qavslar: (), [], { }; probel yoki bo‘sh joy tashlash, ‘ (apostrof); “ (qo‘sh tirnoq) va hokazo.

6. Xizmatchi so‘zlar: AND-va, ARRAY-massiv, BEGIN-boshlamoq, CASE-variant, CONST-o‘zgarimas, DIV-butunga bo‘lish, DO-bajarmoq, DOWNTO- gacha, ELSE-aks holda, END-tamom, FILE-fayl, FOR-uchun, FUNCTION-funksiya, GOTO-ga o‘tish, IF-agar, IN-ga, LABEL-belgi, MOD-modul, NOT-yo‘q, OF-dan, OR-yoki, PROGRAM-dastur, RECORD-yozuv, REPEAT-takrorlamoq, SET-to‘plam, THEN-u holda, TO-gacha, TYPE-turi, UNTIL-gacha, VAR-o‘zgaruvchi, WHILE-hozircha.

Foydalanuvchi tomonidan bajarilishi lozim bo‘lgan ma‘lum xarakterni elektron hisoblash mashinalariga maxsus so‘zlardan tashkil topgan operatorlar `rdamida yetkazib amalga oshirish mumkin. Demak, kompyuter uchun operator bajarilishi so‘zsiz shart bo‘lgan buyruqdir.

Operatorlar algoritmik tillarda asosiy tushuncha bo‘lib, o‘z navbatida ikki guruxga bo‘linadi: oddiy va murakkab operatorlar. Oddiy operatorlar jumlasiga begin, end, Uses, const, label kabi operatorlar kiritilishi mumkin.

Murakkab operatorlar bir necha asosiy operatorlarni o‘z ichiga oladi.

Kompyuterda biror masalani yechish uchun boshqa dasturlarga, tashqi qurilmalarga murojaat qilish mumkin, o‘zgarimas yoki yangi o‘zgaruvchilarning ko‘rinishini e‘lon qilish mumkin va xakozo.

Shunday qilib **Paskal tilidagi dastur strukturasi** quyidagi ko‘rinishda bo‘ladi

```

Program   Programma mavzusini berish
Uses     Ishlatilayotgan kutubxona bo‘limlari (modullari)
Label    Dasturning asosiy qismida ishlatilayotgan belgi (metka) larni e‘lon qilish
Const    O‘zgarimaslarni e‘lon qilish
Type     Yangi o‘zgaruvchilarning turini muomalaga kiritish
Var      Asosiy dasturda muomalada bo‘ladigan o‘zgaruvchilarni e‘lon qilish
Procedure, Function – Protsedura va Funksiyalarni e‘lon qilish.
Begin
    Dasturning asosiy qismi
End.
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Demak, har qanday dastur yuqorida berilgan asosiy tuzilmaning xususiy xoli bo‘lishi mumkin va ular o‘z navbatida Paskal tiliga xos bo‘lgan asosiy tushunchalar asosida xosil qilinadi.

Paskal tilida o‘zgaruvchilarni tavsiflash

Ma’lumki, har qanday qiymat yoki belgi bilan ish ko‘rish uchun eng avvalo ularga xotirada joy ajratish zurrur bo‘ladi. Buning uchun ishlatilishi zarur bo‘lgan o‘zgaruvchi yoki o‘zgarmaslar Paskal tilida e‘lon qilinishi kerak. Ko‘pchilik xollarda dasturlarda o‘zgarmas qiymatlar bilan ish ko‘rishga to‘g‘ri keladi. Masalan, $n=20$, $e=2.71$ kabi sonlar Paskal tilida quyidagicha e‘lon qilinadi.

Const $\pi=3.14$; $n=20$; $e=2.71$;

Umumiy holda o‘zgaruvchilar var (variable) operatori orqali qabul qilishi mumkin bo‘lgan qiymatiga qarab turlarga bo‘linadi. Butun sonlar ishlatilishi chegarasiga qarab har xil e‘lon qilinishi mumkin.

Butun sonli tipda berilganlar arifmetik ifodalarda qo‘llaniladigan qiymatlardan iborat bo‘lib, 1 dan 4 baytgacha bo‘lgan xotirani egallaydi.

Tipi	Diapazoni	Kerak bo‘lgan xotira (bayt)
Byte	0..255	1
Shortint	-128..127	1
Integer	-32768..32767	2
Word	0..65535	2
Longint	-2147483648..2147483647	4

Haqiqiy tiplar matematik ifodalarda qo‘llaniladigan haqiqiy qiymatlardan iborat bo‘lib, 4 dan 6 baytgacha xotirani egallaydi.

Haqiqiy sonlar uchun qo‘yilgan masalada yechimning aniqlik darajasiga qarab quyidagi operatorlar yordamida identifikatorlar e‘lon qilinadi:

Identifikator turi	qiymatlar oralig‘i	Aniqlik darajasi	Egallagan shajmi
Real	$2.9e^{-39} \dots 1.7e^{38}$	11 – 12	6 bayt
Single	$1.5e^{-45} \dots 3.4e^{38}$	7 – 8	4 bayt
Double	$5.0e^{-324} \dots 1.7e^{308}$	15 – 16	8 bayt
Extended	$3.4e^{-4932} \dots 1.1e^{4932}$	19 – 20	10 bayt
Comp	$-2e^{+63} \dots 2e^{+63}$	10 – 20	8 bayt

Belgili tiplar

Literli (belgili) tip ShK ni kod jadvalining qiymatlarini aniqlaydi. Literli tip o‘zgaruvchisi uchun 1 bayt kerak bo‘ladi.

Misol.

VAR

Ch: char;

Dasturda char tipidagi o‘zgaruvchilar va konstantalar apostrof ichiga olib yoziladi. Misol uchun, 'A' A harfini beradi, ' ' - joy tashash, ';' - nuqta vergul.

Bulev tipi

Bulev tipi ikki xil qiymat bilan beriladi: True (rost) va False (yolg‘on). Bu qiymatlar mantiqiy ifodalarda va munosabat ifodalarida keng qo‘llaniladi.

Jadval. Bulev tipi

Tipi	Diapazoni	Kerak bo‘lgan xotira
Boolean	True, False	1

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Foydalanuvchining tiplari

Pascal tilida standart tiplardan tashqari foydalanuvchi tomonidan aniqlangan skalyar tiplar mavjud. Bularga sanab o‘tiladigan va interval tiplar kiradi va xotirada 1 bayt joy egalaydi. Shuning uchun foydalanuvchining tiplari 256 belgidan oshmasligi kerak. Ularni qo‘llash dastur ko‘rinishini ancha o‘zgartiradi, xatolarni topish osonlashadi va xotira tejaladi.

Sanab chiqiladigan tiplar shu tipdagi berilganlar qanday qiymatlarni qabul qilsa, shu qiymatlarni sanab chiqish orqali beriladi. Alohida qiymatlar vergul orqali ajratiladi, hamma ro‘yxat esa qavs ichiga olib ko‘rsatiladi.

Yozilishi: TYPE

<tip nomi> = (<1-qiymat, 2- qiymat,..., n - qiymat>);

VAR

<identifikatori,...> : <tip nomi>;

Misol.

TYPE

Gaz = (C, O, N, F);

Metall = (Fe, Co, Na, Cu, Zn);

VAR G1, G2, G3 : Gaz;

Met1, Met2 : Metall;

Season: (Winter, Spring, Summer, Autumn);

Bu misolda foydalanuvchi tipining ikkita Gaz va Metall ko‘rinishidagi yozuvlari berilgan. Ularni qiymatini aniqlash - Mendeleyev D.I davriy sistemasidagi gaz va metallarning belgilanishini beradi. G1, G2, G3 va Met1, Met2 o‘zgaruvchilari yuqorida keltirilgan qiymatlarning bittasini qabul qilishi mumkin. Boshqa qiymatlarni qabul qilish dasturni uzilishiga olib keladi. Uchinchi tipdagi sanab o‘tiladiganlar ananim (nomsiz) va ular VAR da qiymatlarni sanab o‘tish orqali beriladi.

Season shu tipdagi o‘zgaruvchi bo‘lib, Winter, Spring, Summer va Autumn qiymatlarini qabul qilishi mumkin.

Bir xil tipdagi sanab o‘tiladigan qiymatlar uchun munosabat va mantiqiy operatsiyalarni qo‘llash mumkin. Tartiblash yozuv tipi elementinining tartib nomeri orqali amalga oshiriladi. Masalan, Winter < Spring ifodasi rost bo‘ladi chunki, yozuv tipida Spring Winter ga nisbatan katta tartib nomeriga ega.

Pascal boshqa tiplarga nisbatan farqliroq, foydalanuvchining sanab o‘tiladigan tiplarida kiritish-chiqarish operatsiyalarini qo‘llamaydi. Kerak bo‘lgan paytda foydalanuvchining o‘zi kiritish-chiqarishni tashkil qiladi. Sanab o‘tilgan tiplar bilan ishlash uchun Pascal tilida Succ, Pred, Ord standart quyi dasturlaridan foydalaniladi.

Interval tip, berilgan o‘zgaruvchi uchun qiymatlar chegarasi diapozonini aniqlovchi ikkita konsantani berish imkonini beradi.

Kompilyator interval tipdagi o‘zgaruvchilarda har bir operatsiyadan keyin o‘rnatilgan ichki diapazonda o‘zgaruvchining qiymati qoladimi yoki yo‘qmi, tekshirish qism dasturini generatsiya qiladi. Ikkala konstanta ham standart tiplarning birortasiga (real dan tashqari) tegishli bo‘lishi shart. Birinchi konstantaning qiymati albatta ikkinchi konstanta qiymatidan kichik bo‘lishi shart.

Yozilishi: TYPE

<tipning nomi> = <1-konstanta> .. <2-konstanta>;

VAR

<identifikatori,...> : <tip nomi>;

Misol.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

TYPE

Days = 1 .. 31;

VAR

RabDay, BolnDay : Days;

Bu misolda RabDay va BolnDay o‘zgaruvchilari Days tipida bo‘lib, ular 1...31 diapazonda har qanday qiymatlarni qabul qilishi mumkin. Diapazondan chiqish dastur o‘zilishiga olib keladi. Interval tipni boshqacharoq, universal usul bilan ham aniqlash mumkin. Bu usulda diapazon chegarasini konstanta qiymatlari bilan emas, nomi orqali aniqlash mumkin:

CONST

Min = 1; Max = 31;

TYPE

Days = Min .. Max; VAR

RabDay, BolnDay : Days;

Paskal tilining standart funksiyalari va kalit so‘zlari

Har bir algoritmik tilning dastur matnini yozish qoidalari turlicha bo‘ladi. Dasturlash tillaridan eng soddasi Beysik tilining ma‘lum versiyalarida dasturning bar bir operatori qat‘iy aniqlangan qator raqamlari orqali yoziladi. Paskal tilida esa operatorlar ketma-ket yozilib, o‘zaro «;» belgisi bilan ajratib boriladi. Bundan tashqari, yozilgan dasturning o‘qishga oson va undan foydalanish qulay bo‘lishi uchun dasturda «matnni ajratish» tushunchasi (bo‘sh joy, qatorning tugashi va izohlar) dan foydalaniladi. Bo‘sh joy (probel) grafik tasvirga ega bo‘lmagan belgi bo‘lib, qatordagi bo‘sh joyni anglatadi. Lekin, bo‘sh joy belgisi o‘zining sonli kodiga ega va dastur matnidagi boshqa belgilar kabi komputerga kiritiladi. Qator oxiri (tugashi) boshqaruvchi belgi bo‘lib, u ham grafik tasvirga ega emas. Ma‘lumki, dastur matnini yozish davomida uni tabiiy ravishda yangi qatorlarga ajratilib yoziladi. Chunki, shu matn yozilmoqchi bo‘lgan qog‘ozning ham, komputer ekranining ham o‘lchamlari cheklangan. Dastur matnini alohida qatorlarga ajratmay yozish ham mumkin, lekin bir satrga 256 tadan ortiq belgi sig‘maydi. Dastur matnini alohida qatorlarga ajratish dastur tuzuvchining xohishiga qarab bajariladi. Ma‘lum bir qator tugamay turib, yangi qatorga o‘tish uchun «qator oxiri» tugmachasi bosiladi. Bu tugmacha ham o‘zining maxsus sonli kodiga ega.

Izohlar dasturni o‘qishga oson bo‘lishi, uni qiynalmay tekshirib, yo‘l qo‘yilgan xatolarni to‘g‘rilash va dasturda bajarilayotgan ishlarni tushuntirib borish uchun qo‘yiladi. Izohsiz yozilgan dasturni hujjat sifatida qabul qilinmaydi. Muvaffaqiyatli qo‘yilgan izoh dasturning va dasturchining katta yutug‘i hisoblanadi. Izohlar ixtiyoriy vaqtda dastur matniga kiritilishi yoki olib tashlanishi mumkin. Bu bilan dasturning ishi o‘zgarib qolmaydi. Izohlarni «{» va «}» qavslari ichiga olinib yoziladi. Dastur «matn ajratgich»laridan foydalanishning quyidagi qoidalarga amal qilish lozim: tilning ketma-ket yozilgan ikkita konstruksiyasi orasiga albatta bo‘sh joy yozilishi kerak; ajratgichlarni xizmatchi so‘zlar, sonlar va ismlar orasiga qo‘yish maqsadga muvofiq emas.

Paskal tilida standart funksiyalarning berilishi quyidagi jadvalda keltirilgan

<i>Matematikada</i>	<i>Paskalda</i>	<i>Izoh</i>
+ , - , · , ÷	+ , - , * , /	qo‘shish, ayirish, ko‘paytirish, bo‘lish
Sin x	Sin(x)	Sinus
Cos x	Cos(x)	Kosinus
tg x	Sin(x)/ Cos(x)	Tangens
Ctg x	Cos(x)/ Sin(x)	Kotangens
arctg x	arctan(x)	Tangens

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$ x $	Abs(x)	X ning absolyut qiymati
$[x]$	Int(x)	int(5.2)=5 butun qismi
lnx	Ln(x)	Natural logarifm
e^x	exp(x)	E= 2,71 exponent son
x ni kasr qismi	Frac(x)	Frac(5.2)=0.2 kasr qismi
x ni yaxlitlash	Round(x)	Round(3.245)=3.25
x ning butun qismi	trunc(x)	trunc(3.2)=3
$\frac{x}{y}$ dagi qoldiq	x mod y	10 mod 3=1
$\left[\frac{x}{y} \right]$	x div y	10 div 3=3
x^2	Sqr(x), x*x	x ni kvadrati
\sqrt{x}	Sqrt(x)	X ni kvadrat ildizi
$x < y, x > y$	$x < y, x > y$	x kichik y dan, x katta y dan
$x \geq y$	$x \geq y$	X katta yoki teng y ga
$x < y < z$	$x < y$ and $y < z$	x kichik y dan va y kichik z dan
$4 \cdot 10^9$	4e9	O‘n darajasi

Darajaga ko‘tarish amali bo‘lmaganligi uchun x^y ni paskalda tasvirlash uchun quyidagi shakl almashtiramiz $x^y = e^{\ln x^y} = e^{y \ln x}$. Demak, $x^y = e^{y \ln x}$ Paskalda $\exp(y * \ln(x))$. Bu erda $x > 0$. agar $x < 0$ bo‘lsa $-\exp(y * \ln(x))$.

Shu formuladan foydalanib $\sqrt[y]{x}$ ni paskal ko‘rinishida yozaylik. Buning uchun shakl almashtirishni amalga oshiramiz:

$\sqrt[y]{x} = x^{\frac{1}{y}}$ ko‘rinishiga ega. Demak, $\exp((1/y) * \ln(x))$ ko‘rinishida yoziladi.

Simvollar uchun quyidagi funksiyalar ishlatiladi:

Chr(n) - n tartib nomeriga mos keluvchi belgini aniqlaydi

Ord(x) - x belgining tartib nomerini aniqlaydi

Pred(x) - x dan oldingi belgining tartib nomerini aniqlaydi

Succ(x) - x dan keyingi belgining tartib nomerini aniqlaydi

Keltirilgan funksiyalar Paskal tilida maxsus funksiyalar deb ataladi.

Paskal tilining kalit so‘zlari

And-va

Array –massiv

Begin-boshlash

Case-variant

Const-o‘zgarmas

Div –qoldiqsiz butun bo‘lish

Do –bajarish

Downto –gacha kamaytirish

Else –aks holda

End –tamom

File –fayl

For –uchun

Function –funktsiya

Procedure –prosedura

Goto –ga o‘tish

If –agar

In –ga tegishli

Label-belgi

Mod-butun qoldiqli bo‘lishdagi qoldiq.

To –gacha ko‘paytirish

So‘zlar rezervlashgan so‘zlarga, standart indentifikatorlarga va foydalanivchining indentifikatorlariga bo‘linadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Rezervlashgan so‘zlar tilning tashkiliy qismi hisoblanib, aniq ma’noga ega bo‘ladi. Quyida Pascal versiyasidagi ShEHM lar uchun rezervlashgan so‘zlar ro‘yxati keltirilgan:

absolute	and	array	begin	case
const	end	div	do	external
file	for	forward	function	goto
inline	interface	interrupt	mod	nil
procedure	type	program	record	unit
until	label	repeat	uses	set
var	shl	while	not	shr
with	if	of	string	xor
downto	implementation	or	then	else
in	packed	to		

Standart identifikatorilar oldindan aniqlangan o‘zgaruvchilarni, konstanta, protsedura va funksiyalarni belgilashga xizmat qiladi. Masalan, Sin(x) standart identifikatori, berilgan burchak sinusini hisoblash funksiyasini chaqiradi. Har qanday standart identifikatorini rezervlashgan so‘zlardan farqi shundaki, uni oldindan aniqlab olish mumkin. Lekin bu ko‘p holatlarda xatolikga olib keladi. Shuning uchun amaliyotda standart identifikatorilardan ularni o‘zgartirmasdan foydalangan maqulroqdir.

Foydalanivchining identifikatorilaridan dasturchi metka, konstanta, o‘zgaruvchilar, protsedura va funksiyalarni belgilashda foydalanadi. To‘g‘ri tanlangan identifikatori dastur tushinishni, o‘qishni osonlashtiradi va dasturni modifikatsiyalashda xato qilish extimolini kamaytiradi.

Masalan, oy, kun, yilni D harfi yoki boshqa biror bir belgidan ko‘ra Data identifikatori bilan begilash qulayroqdir.

Identifikatorilardan foydalanishning umumiy qoidalari mavjud:

1. Identifikatori faqat harf yoki chiziqcha belgisi bilan boshlanadi (bundan son yoki harf bilan boshlandigan metka mustasno).
2. Identifikatori harf, son va chiziqcha belgisidan tashkil topishi mumkin (probel, nuqta va maxsus belgilardan foydalanish mumkin emas).
3. Ikkita identifikatori oraligida hech bo‘lmaganda bitta probel bo‘lishi shart.
4. Identifikatorilar uzunligi 127 ta belgidan iborat, lekin faqat oldingi 63 tasigina hisobga olinadi xolos.

1 graph - xato, identifikatori son bilan boshlandi.

Block_56

Nomer. Doma - xato, identifikatorida nuqta ishtirok etayapti.

Konstantalar va o‘zgaruvchilar

Konstanta, o‘zgaruvchi, tiplashgan konstanta

Har qanday dastur qandaydir berilganlar bilan ishlagan taqdirdagina ma’noga ega bo‘ladi. Xuddi boshqa dasturlash tillari kabi, Pascal tili ham konstanta yoki o‘zgaruvchi ko‘rinishidagi berilganlar bilan ish olib boradi. Shunday qilib, dasturdagi har bir element *o‘zgaruvchi* yoki *konstanta* bo‘ladi. Konstanta va o‘zgaruvchilar uzlarining identifikatorilari (nomlari) orqali aniqlanadi va shu nomlar orqali ularga murojat etiladi.

Konstantalar deb dastur boshida e‘lon qilingan va dastur oxirigacha o‘zgarmaydigan qiymatlarga aytiladi. Konstantani aniqlash uchun rezervlashgan CONST so‘zidan foydalanamiz.

Yozilishi: CONST

<identifikatori> = <konstantaning qiymati>;

Masalan.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

CONST

Max = 1000;

Vxod = 'Segment 5';

O‘zgaruvchilarning konsantantalardan farqi shundaki, ular o‘z qiymatlarini dastur ishi davomida o‘zgartirishi mumkin. Har qanday o‘zgaruvchilar va konstantalar aniq bir berilganlar tipiga kiradi. Konstantalar tipini avtomatik tarzda kompilyatorlar yordamida aniqlanadi. O‘zgaruvchilarning tipi ular bilan ishlashdan oldin e‘lon qilinishi zarur. O‘zgaruvchilarni e‘lon qilish uchun VAR so‘zi qo‘llaniladi.

Yozilishi: VAR

<identifikatori> : <tip>;

Masalan.

VAR

Sum1, Sum2: real;

O‘zgaruvchining nomi "qobiq" hisoblanib, uni qiymatlar bilan to‘ldirish mumkin, lekin konstantalar bilan buni qilib bo‘lmaydi.

Konstanta va o‘zgaruvchilardan tashqari ikkala o‘zgaruvchi oralig‘ida qo‘llaniladigan *tiplashgan konstantalar* mavjud. "Tiplashgan" so‘zi konstantalarni e‘lon qilishda o‘zgaruvchilardagi kabi konstantaning tipi ham ko‘rsatilishi kerakligini bildiradi.

Yozilishi: CONST

<identifikatori>:<tip>=<qiymat>;

Masalan.

CONST

VideoSeg : word = \$B800;

Berilganlarning standart tiplari

Tip, skalyar tiplar, standart va foydalanuvchining tiplariga butun sonli, haqiqiy, literli, bulev tipidagi berilganlar, ko‘rsatgichlar

Tip - bu qiymatlar to‘plami bo‘lib, uni dastur ob‘ekti qabul qilishi mumkin va shu qiymatlar ustida olib boriladigan operatsiyalar yig‘indisidir. Masalan, 1 va 2 soni, butun sonlar tipiga kiradi, ularni qo‘shish, ko‘paytirish va boshqa arifmetik operatsiyalarni bajarish mumkin. Pascal tilida umumiy holatlarda, tiplarni e‘lon qilish uchun TYPE rezervlashgan so‘zidan foydalaniladi.

Yozilishi: TYPE

<Tip nomi> = <tip qiymati>;

Berilganlar tipi ikki guruhga bo‘linadi: skalyar (oddiy) va strukturalashgan (tarkiblashgan). *Skalyar tiplar* o‘z navbatida *standart va foydalanuvchi* tiplariga bo‘linadi.

Standart skalyar tiplarga *butun, haqiqiy, literli, ko‘rsatkich va bulev* tipidagi berilganlar kiradi.

Butun tipdagi berilganlar o‘nli yoki o‘n olti sistemalarda berilishi mumkin. Agar son 16 lik sistemada berilgan bo‘lsa, uning oldiga \$ belgisi quyiladi. 16 lik sistemasidagi sonlarning o‘zgarish chegarasi \$0000 dan \$FFFF gacha.

O‘nli sistemasidagi sonlar ikki xil usulda yozilishi mumkin:

Qo‘zg‘aluvchan va qo‘zg‘almas nuqtali o‘zgarmlar

Haqiqiy o‘nli sonlar oddiy arifmetik qoidalarga ko‘ra yoziladi. Sonning butun qismi kasr qismidan vergul orqali ajratiladi. Agarda nuqta bo‘lmasa, son butun son deb hisoblanadi. Sonning oldiga "+" yoki "-" belgisi quyish mumkin.

Masalan.

125 - butun o‘nli son

\$1FF - 16 lik son

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Qo‘zg‘aluvchi nuqta ko‘rinishidagi haqiqiy butun son quyidagi eksponensial ko‘rinishda tasvirlanadi: $mE+p$, bunda m - mantissa (nuqta bilan ajratilgan butun yoki kasr son), "E" o‘ning darajasini bildiradi, p - tartib (butun son).

Masalan.

$$5.18E+02 = 5.18 * 10^2 = 518$$

$$10E-03 = 10 * 10^{-3} = 0.01$$

Foydalaniuvchining tipi - sanaladigan va intervalli dasturchi tomonidan beriladi.

Strukturalashgan tipning asosini bir va bir nechta skalyar tipdagi berilganlar tashkil etadi. Strukturali tiplarga qatorlar, massivlar, to‘plamlar, yozuvlar va yangi tipdagi fayl va berilganlar: prosedurali va object tiplar kiradi.

Pascal tilidagi dasturlar protsedura va funksiyalardan tashkil topadi. Dasturning boshida PROGRAM so‘zi bilan boshlanuvchi dastur nomi turadi. *Dasturga nom* quyish shart emas, lekin dastur nomi bo‘yicha axtarilganda uni topish oson bo‘ladi, shuning uchun nom quyiladi. Dastur parametrlari standart identifikatori va kiritish-chiqarish Input va Output standart fayllaridan iborat bo‘ladi:

PROGRAM PacStat (Input, Output);

PROGRAM MathHandler (Input, Output);

Dastur nomidan keyin 7 bo‘limdan iborat bo‘lgan dastur bloki keladi: biblioteka modulidagi nomlar ro‘yxati (u USES so‘zi yordamida aniqlanadi), metkalar yozuvi, konstantalar yozuvi, berilganlar tipini aniqlash, o‘zgaruvchilarning yozuvi, protsedura va funksiyalarni yozuvi, operatorlar.

Dastur strukturasi quyidagi ko‘rinishga ega:

PROGRAM <nom> (Input, Output);

USES <1-nom, 2-nom,...>;

LABEL ...;

CONST ...;

TYPE ...;

VAR ...;

PROCEDURE <nom>;

<protsedura tanasi>

FUNCTION <nom>;

<funksiya tanasi>

BEGIN

<operatorlar>

END.

Operator bo‘limidan tashqari har qanday boshqa bo‘lim qatnashmasligi mumkin. Yozuvlar bo‘limi dasturda xohlagan miqdorda qatnashishi mumkin.

USES bo‘limi

Bu bo‘lim USES so‘zidan va standart foydalanuvchi biblioteka modullari nomlari ro‘yxatidan iborat bo‘ladi.

Yozilishit: USES <1-nom>,<2-nom>,...;

Misol.

USES Crt, Dos, MyLib;

Metkalarni ifodalash bo‘limi

Metka, metkalar

ni ifodalash bo‘limi (Label)

Pascal tilining har qanday operatori oldiga metka quyish mumkin, u shu metkali operatorga goto orqali dasturning xohlagan joyidan to‘g‘ridan-to‘g‘ri o‘tish mumkinligini ko‘rsatadi. *Metka* nom va undan keyin quyiladigan ikki nuqtadan iborat bo‘ladi. Nom sifatida son yoki identifikatori qatnashishi mumkin. Metka nomining uzunligi 127 simvolgacha bo‘lishi mumkin. Metkadan foydalanishdan oldin, u *metkalar*ni yozilishi

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

bo‘limida e‘lon qilingan bo‘lishi kerak Metkalarni yozilishi bo‘limi LABEL (metka) so‘zi bilan boshlanadi. Oxirgi nomdan so‘ng nuqta vegrul quyiladi.

Yozilishi: LABEL <nom,...>;

Misol.

LABEL

Metka1, Metka2, 111, Blok10;

Metka yozilgandan so‘ng operatorlar bo‘limida ikki nuqta quyiladi:

LABEL M1, M2; {metkalar yozuvi}

BEGIN

...

M1: <operator> {M1 ni operatorlar bo‘limida ishlatish}

...

M2: <operator> { M2 ni operatorlar bo‘limida ishlatish}

END.

metkalarni dastur kengaytmasi bo‘yicha ifodalash va qo‘llash mumkin.

Funksiya va protseduralarni ifodalash bo‘limi

Qism dasturi, standart protsedura va funksiyalar,

Bu bo‘limda qism dasturlarining tanalari joylashadi. *Qism dasturi deb* dasturning boshqa qismlaridan chaqirilishi mumkin va nomga ega bo‘lgan dastur birligiga aytiladi. Pascal dasturlash tilida qism dasturi rolini protsedura va funksiyalar bajaradi. Umumiy hollarda qism dasturi ham dastur kabi strukturaga ega. Qism dasturini ifodalash uchun dasturning boshida yoziladigan PROCEDURE va FUNCTION so‘zlaridan foydalaniladi.

Protsedurani yozilishi:

PROCEDURE <protsedura nomi {<parametrlar>};

<yozuvlar bo‘limi >

<operatorlar bo‘limi>

END;

Funksiyaning yozilishi:

FUNCTION <funksiyaning nomi {<parametrlar>} : <natija tipi>;

<yozuvlar bo‘limi>

<operatorlar bo‘limi>

END;

Protsedura va funksiyalar standart va foydalanuvchi tomonidan aniqlangan bo‘lishi mumkin. Standart protsedura va funksiyalar tilning bir qismi bo‘lib ularni e‘lon qilmasdan ham chaqirish mumkin. Foydalanuvchining protsedura va funksiyalari e‘lon qilinishi shart.

Operatorlar bo‘limi

Operator, Begin, End.

Pascal tilidagi dasturlarda operatorlar bo‘limi asosiy bo‘lim hisoblanib, bu bo‘limda o‘zgaruvchilar, konstantalar, o‘zgaruvchilarning qiymatlari e‘lon qilinib, ular ustida amallar olib boriladi va natijalar olinadi. Operatorlar bo‘limi *BEGIN* (boshlandi) so‘zidan boshlanadi, so‘ngra tilning operatorlari yoziladi, ular bir-biridan nuqta vergul orqali ajratiladi.

Bo‘lim *END* (tugadi) so‘zi bilan tugatiladi va nuqta quyiladi.

BEGIN

<operator;>

...

<operator>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

END.

Dasturdagi operatorlar yozilish ketma-ketligi bo‘yicha bajariladi.

Izohlar

Izoh, chegara belgilari

Izoh - bu dasturning xohlagan ifodasi bo‘lgan tushuntirish matni. *Izoh* matni () (**) bilan chegaralangan.

Misol.

```
{Regress dasturiga izoh}
```

```
(* Lagranj polinomini hisoblash uchun dastur *)
```

(**) chegaralarda bo‘sh joy qolishi mumkin emas. Matnda izoh boshlanadigan *chegara belgilari* bo‘lmasligi kerak.

Masalan, izoh matni

```
{ Misol {1} vazifa {4} }
```

bu kompilatsiyalash vaqtida xatolikka olib keladi. Lekin () ni (**) ga qo‘shib quyish ham mumkin va aksincha

```
(* Misol { 1 }vazifa { 4 } *)
```

```
{ Misol (* 1 *) vazifa (* 4 *) }.
```

Foydalanuvchining biblioteka modullari

Biblioteka moduli, biblioteka modulining strukturasi (UNIT, INTERFACE, IMPLEMENTATION)

Biblioteka moduli tushunchasi Turbo Pascal dasturlash tilining ideologiyasida dasturlash sitemasining asosini tashkil etadi. Xuddi shular asosida biblioteka qism dasturlari (protsedura va funksiyalar) tuziladi. Biblioteka moduli- Compile rejimida Destination = Disk direktoriyasi yordamida o‘rnatilgan bir yoki bir-nechta protsedura va funksiyalarining kompilyatsiyasi natijasidir. Modul ma’noga ega, u USES bo‘limida e’lon qilinadi, va uning yordamida dasturdagi har qanday protsedura yoki funktsiyaga murojat qilish mumkin.

Biblioteka modullarini tashkil etishda UNIT, INTERFACE, IMPLEMENTATION, BEGIN, END so‘zlariga murojat qilinadi. Sistema kompilyatsiyalanayotgan fayl strukturasi aniqlab, TPU-fayl (agar fayl ichida UNIT va x.k. so‘zlari bo‘lsa) yoki .EXE-fayl (agar UNIT, IMPLEMENTATION va x.k. lar bo‘lmasa) hosil qiladi. Birinchi holatda biblioteka moduli shakllanadi, ikkinchi holatda esa ishga tayyor bo‘lgan DOS yo‘qlovchi moduli hosil bo‘ladi.

*Biblioteka modulining umumiy strukturasi*ni ko‘rib chiqamiz:

```
UNIT < biblioteka modulining nomi >;
```

```
INTERFACE {interfeys seksiya} USES <ulanadigan modul nomi >,...;
```

```
<protsedura sarlavhasi /parametrlari ko‘rsatilgan 1-funksiya >
```

```
<protsedura sarlavhasi /parametrlari ko‘rsatilgan 2-funksiya > ...
```

```
<protsedura sarlavhasi /parametrlari ko‘rsatilgan n-funksiya >
```

```
IMPLEMENTATION {ishlatish seksiyasi}
```

```
USES <ulanayotgan modulning nomi >,...;
```

```
< parametrlarsiz sarlavha va protsedura tanasi /1-funksiyaning >
```

```
< parametrlarsiz sarlavha va protsedura tanasi /2-funksiyaning > ...
```

```
< parametrlarsiz sarlavha va protsedura tanasi /n-funksiyaning >
```

```
BEGIN {initsializatsiyalash seksiyasi } <operator>; ...
```

```
<operator>
```

```
END.
```

Biblioteka modulining nomi diskdagi fayl nomiga mos tushishi kerak. Masalan, agarda fayl Stat.PAS bo‘lsa, u holda modulning nomi Stat bo‘lishi lozim:

```
UNIT Stat;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Initsializatsiya seksiyasi modulning oxirgi seksiyasi hisoblanib, BEGIN va END (modul kod initsializatsiyasiga ega bo‘lmasa) so‘zlaridan yoki modul initsializatsiyasini bajarishi kerak bo‘lgan opererator qismidan tashkil topadi.

TPU-bibliotekasini tashkil qilishda har bir dasturchida uchraydigan tipik holatni ko‘rib chiqamiz. Diskda juda ko‘p qo‘llaniladigan fodalalanuvchining protsedura va funksiyalari saqlanadigan MyLib biblioteka modulini tashkil qilish talab etilsin.

Biblioteka modulini tashkil qilish

1. File bosh menyusi rejimini o‘rnatish.
2. Load rejimi yordamida protsedura va funksiyalar matni saqlangan MyLib.PAS faylini yo‘qlash (ularda xatolik bor deb faraz qilinadi).
3. Redaktor yordamida biblioteka modulini oluvchi strukturani tashkillashtirish (UNIT, IMPLEMENTATION va x.k. larni qo‘llash yordamida).
4. Compile bosh menyu rejimini o‘rnatish.
5. Disk holatida Destination qism rejimini o‘rnatish.
6. Compile rejimini aktivlashtirib, kompilyatsiyalashni bajarish.
7. Diskda MyLib.TPU biblioteka moduli avtomatik ravishda tashkillashtiriladi.
8. Bibliotekadagi protsedura va funksiyalarning vazifasi, nomi va parametrlari haqidagi malumot beruvchi qisqacha instruksiya yozish.

Biblioteka modullarini qo‘llash

1. Xotiraga oldindan tashkil qilingan MyLib biblioteka modulini qo‘llash ehtimoli bo‘lgan dasturni yo‘qlash, masalan., MyProg ni.
2. Bu dasturning USES bo‘limida biblioteka modulining nomini ko‘rsatish.
MyLib:
USES MyLib;.
3. Instruksiyaga asosan dasturda MyLib modulidagi kerakli protsedura va funksiyalarni ishlatish.
4. Dasturni yozib bo‘lgandan so‘ng asosiy menyuga chiqish.
5. Options rejimini o‘rnatish.
6. Directories qism rejimida UnitDerictories ni o‘rnatib, MyLib moduliga yo‘l ko‘rsatish.
7. Asosiy menyuga chiqish.
8. Run yordamida MyProg dasturini bajarishga start berish.
9. Dastur ishining to‘g‘riligiga ishonch hosil qilish.
10. Compile bosh menyusi rejimini o‘rnatish.
11. Disk holatida Destination qism rejimini o‘rnatish.
12. Compile rejimini aktivlashtirib, kompilyatsiyalashni bajarish.
13. Diskda dasturning tugallangan maxsuloti bo‘lgan, bajariluvchi MyProg.EXE moduli tashkillanadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

MA’RUZA № 6

MAVZU: CHIZIQLI, TARMOQLANUVCHI VA TAKRORLANUVCHI JARAYONLARNI DASTURLASH.

REJA:

1. Pascal dasturlash tilida kiritish va chiqarish operatorlari.
2. O‘zlashtirish operatori.
3. Chiziqli jarayonlarni dasturlash.
4. Paskal tilida shartli va shartsiz o‘tish operatorlari.
5. Tanlash operatori.
6. Paskal tilining takrorlash operatorlari.

Kalit so‘zlar: Pascal dasturlash tili, operatorlar, sodda operatorlar, kiritish, chiqarish, o‘zlashtirish, shartli operatorlar, dasturlash, shartsiz o‘tish operatori, protsedurani chiqarish operatori, bo‘sh operator, strukturali operator, tanlash operatori, selektor, parametrlar ro‘yxati, tanlash o‘zgarma slari ro‘yxati.

Dasturda ma’lumotlarning qiymatlarini xotiraga kiritishni bir necha usullarda bajarish mumkin. Sonli o‘zgaruvchilarga ularning qiymatini berishda o‘zlashtirish operatoridan foydalaniladi. Masalan:

A:=5; V:=6.143;

Dasturni o‘zgaruvchilarning turli qiymatlarida bajarish uchun **READ** - kiritish operatori mo‘ljallangan.

Kiritish operatori quyidagisha ko‘rinishlarda ishlatilishi mumkin:

1) READ(al,a2,...,an);

bunda, al,a2,...,an - o‘zgaruvchi qiymatlarini ketma- ket standart INPUT prosedura faylidan oluvshi o‘zgaruvchilar. O‘zgaruvchilarga qiymatlar turiga mos ravishda klaviaturadan kiritiladi. Aytaylik, A, V, S o‘zgaruvchilarga dastur bajarilishi davomida quyidagi qiymatlarni berish kerak bo‘lsin:

A=5, V=17, S=6.2.

Operator READ(A,B,S) ko‘rinishiga ega bo‘lib, sonlar qiymatlarini dastur bajarilishi davomida quyidagisha kiritish mumkin:

5 17 6.2 [Enter].

Agar o‘zgaruvchi REAL toifada aniqlangan bo‘lsa, uning qiymatini butun son yoki haqiqiy son ko‘rinishida kiritiladi. Mashinaning o‘zi butun sonni haqiqiy songa o‘tkazib oladi.

Masalan:

VAR A, B:REAL;

READ(A, B) operatorining ishlatilishi natijasida 4 va 5 sonlarini probel (bo‘sh joy) orqali kiritish mumkin.

2) READLN - bu operator kiritish jarayonida bo‘sh qator qoldiradi;

3) READLN(al,a2,...,an);

- operatorning bajarilishida avval al ,a2,...,an ga qiymat kiritilib, so‘ng keyingi satrga o‘tiladi. Bu operator oldingi ikki operatorga teng kuchlidir. EHM xotirasidagi ma’lumotlarni displey ekraniga chiqarish operatori - **WRITE** dir. Operator quyidagi bir neshta ko‘rinishlarda ishlatilishi mumkin:

1) WRITE(al,a2, ..., an);

bunda al,a2,...,an oddiy o‘zgaruvchilar, o‘zgarma slar yoki ifodalar bo‘lishi mumkin va ular standart OUTPUT prosedura fayliga chiqariladi.

Masalan:

WRITE('B ning qiymati =', V) operatori displey ekraniga:

V ning qiymati = va undan so‘ng V o‘zgaruvchining qiymatini chiqaradi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

WRITE operatorida butun va haqiqiy sonlarni ma’lum formatda chiqarish mumkin. Bu format ikki nuqta orqali o‘zgaruvchidan so‘ng ko‘rsatiladi.

Masalan:

WRITE(Y:5:2);

operatori bilan Y ning qiymatini chiqadrihda, Y ning hamma qiymatini chiqarish uchun 5 ta xona ajratilishi, ulardan ikkitasi kasr qismi uzunligini anglatadi (bunda sonning butun va kasr qismini ajratuvshi vergul (nuqta) ham hisobga olinishi zarur). Butun sonlarni chiqarishda kasr qismi formati ko‘rsatilmaydi.

Aytaylik, N=179 butun sonli qiymatni chiqarish kerak bo‘lsin. Chiqarish operatori buning uchun quyidagi ko‘rinishda bo‘lishi mumkin:

WRITE(`N=% N:3)

Bu yerda sonni tasvirlash uchun 3 pozitsiya ajratilgan. Agar format 3 dan ortiq berilsa, masalan,

WRITE(`N=`,N:5)

bo‘lsa, unda sondan oldin ikkita bo‘sh joy tashlanadi:

N= 179,

manfiy son uchun e’sa bitta bo‘sh joy tashlanadi: N= - 179.

Paskal tilida boshqa chiqarish operatorlari ham ishlatiladi. Parametrlarsiz chiqarish operatori **WRITELN** - displey ekranida yangi satrga o‘tishni ta’minlaydi.

3)WRITELN(al,a2,...,an);

- chiqarish operatori oldin al,a2,...,an larning qiymatlarini chiqaradi, so‘ng yangi qatorga o‘tishni ta’minlaydi. Shunday qilib, bu ham quyidagi ikki operatorga ekvivalent:

WRITE(al,a2,...,an); WRITELN;

Masalan, A, V, S qiymatlarini kiritish uchun quyidagi lavhadan foydalanish mumkin:

WRITE(`A, V, S qiymatlarini kiriting`);

READ(A, V, S);

Shunday qilib, A, V, S ning qiymatlarini kiritilishidan oldin ekranga quyidagi xabar chiqariladi:

A, V, S qiymatlarini kiritingshundan so‘nggina qiymatlarni kiritish mumkin, masalan, 5 17 6.2 [Enter].

O‘zlashtirish operatori

Odatda dastur natijasini hosil qilish uchun juda ham ko‘p oraliq hisob ishlarini bajarishga to‘g‘ri keladi. Oraliq natijalarni esa ma’lum muddatga saqlab turish lozim bo‘ladi. Bu ishlarni bajarish uchun tilning eng asosiy operatorlaridan biri bo‘lmish - o‘zlashtirish operatori ishlatiladi:

$\langle \text{o‘zlashtirish operatori} \rangle := \langle \text{o‘zgaruvchi} \rangle := \langle \text{ifoda} \rangle;$

Bu yerda := o‘zlashtirish belgisi hisoblanadi, bu belgini =(tenglik) belgisi bilan almashtirmaslik zarur. O‘zlashtirish operatorida := belgisining o‘ng tomonidagi <ifoda> qiymati aniqlanilib, so‘ng chap tomondagi o‘zgaruvchiga o‘zlashtiriladi yoki boshqacha qilib aytganda, ifoda qiymati o‘zgaruvchi nomi bilan xotirada eslab qolinadi. O‘zgaruvchining oldingi qiymati esa (agar u bo‘lsa) yo‘q bo‘lib ketadi.

O‘zlashtirish operatorini yozishdagi eng muhim narsa, bu ifoda va o‘zgaruvchilarning bir xil turli bo‘lishligidir.

O‘zlashtirish belgisining o‘ng tomonidagi ifodaning natijaviy turiga qarab, o‘zlashtirish operatorini uch xil guruhga ajratish mumkin: arifmetik o‘zlashtirish operatori, mantiqiy o‘zlashtirish operatori, belgisi o‘zlashtirish operatori.

Arifmetik o‘zlashtirish operatori

Butun yoki haqiqiy turli, sonli natija beruvchi ifodani (odatda bunday ifodani arifmetik ifoda deb ataladi) hisoblash uchun arifmetik o‘zlashtirish operatoridan foydalaniladi. Arifmetik ifodada qatnashuvchi barcha o‘zgaruvchilar haqiqiy yoki butun turli bo‘lishi kerak. Arifmetik ifoda- sonlar, o‘zgaruvchilar va funksiyalardan

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

tashkil topadi, hamda +, -, *, /, div, mod kabi amallar yordamida yoziladi. Arifmetik amallarni bajarilishi quyidagi tartibda bo‘ladi : *, /, div, mod, +, .

Ifodani bajarilishidagi bu tartibni o‘zgartirish uchun kichik qavslardan foydalaniladi. Ifodaning qavslar ichiga olib yozilgan qismlari mustaqil holda birinchi galda bajariladi.

Sanab o‘tilgan arifmetik amallarning vazifalari bizga matematika kursidan ma’lum. Lekin, bu ro‘yxatdagi div va mod amallari bilan tanish emasmiz. Div – butun bo‘lishni anglatadi, bo‘linmani butun qismi qoldirilib, qoldiq tashlab yuboriladi. Misol:

$$\begin{aligned} 7 \operatorname{div} 2 &= 3 \\ 5 \operatorname{div} 3 &= 1 \\ -7 \operatorname{div} 2 &= -3 \\ -7 \operatorname{div} -2 &= 3 \\ 2 \operatorname{div} 5 &= 0 \\ 3 \operatorname{div} 4 &= 0 \end{aligned}$$

Mod – butun sonlar bo‘linmasining qoldig‘ini aniqlaydi. $m \operatorname{mod} n$ qiymat faqat $n > 0$ dagina aniqlangan. Agar $m \geq 0$ bo‘lsa $m \operatorname{mod} n = m - ((m \operatorname{div} n) * n)$, $m < 0$ bo‘lsa $m \operatorname{mod} n = m - ((m \operatorname{div} n) * n) + n$, $m \operatorname{mod} n$ ning natijasi doim musbat sonidir.

Misol:

$$\begin{aligned} 7 \operatorname{mod} 2 &= 1 \\ 3 \operatorname{mod} 5 &= 3 \\ (-14) \operatorname{mod} 3 &= 1 \\ (-10) \operatorname{mod} 5 &= 0 \end{aligned}$$

Arifmetik ifodaga doir misollar :

$$\begin{aligned} 2 * 5 - 4 * 3, \\ 9 \operatorname{div} 4 / 2, \\ 45 / 5 / 3, \\ a + b / 2 * 7.2 - \operatorname{sqrt}(7), \\ \exp(2 - a) * 9.7 - 6.1 * 6.1 \end{aligned}$$

Paskal tilida darajaga ko‘tarish amali yo‘q, shuning uchun, bu amalni bajarishda logarifmlash qoidasidan foydalanamiz.

Misol: $y = a^n$, $a > 0$ ifodani hisoblashni ko‘rib chiqaylik. Tenglikni ikkala tomonini logarifmlaymiz:

$\operatorname{Ln} y = \operatorname{Ln} a^n$, logarifm xossasiga ko‘ra

$\operatorname{Ln} y = n \operatorname{Ln} a$, bu tenglikdan "u" ni aniqlaymiz,

$U = e^{n \operatorname{Ln} a}$ - bu tenglikni Paskal tilida quyidagicha yozish mumkin: $y = \exp(n * \operatorname{Ln}(a))$.

Endi sal murakkabroq arifmetik ifodalarni Paskal tilida yozilishini ko‘rib chiqaylik.

Matematik yozuvi	Paskal tilidagi yozuvi
$\frac{a+b}{c+d}$	$(a+b)/(c+d)$
$\frac{a(a+b)}{bc}$	$a*(a+b)/(b*c)$
$\frac{1}{1 - \frac{1}{1 - \frac{1}{x}}}$	$1/(1-1/(1-1/x))$
$2-(x-b)^2 - e^{ax} + \sin Cx$	$2-\operatorname{sqr}(x-b)-\exp(a*x)+\sin(c*x)$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$\frac{x \left(e^{\left(x^2 + y^2 \right)} - 1 \right)}{\sqrt{\left x^2 + y^2 \right }}$	$x * (\exp(x*x+y*y) - 1) / \text{sqrt}(\text{abs}(x*x+y*y))$
---	--

Endi arifmetik o‘zlashtirish operatoriga doir misollar ko‘rib chiqamiz:

$x := 0;$

$c := \text{sqrt}(a*a+b*b);$

$y := 2*pi*r; i := i+1; i := 5/4; x := a - b/2;$

O‘zlashtirish operatorining o‘ng tomonidagi ifodada qatnashuvchi o‘zgaruvchilar, albatta, bu operatoridan oldin o‘zining qiymatlariga ega bo‘lishi kerak. Aks holda, o‘zlashtirish operatori o‘z ishini bajara olmaydi. Dastur tuzishda ko‘pchilik yo‘l qo‘yadigan xatolikni quyidagi misolda taxlil qilib ko‘ring:

To‘g‘ri tuzilgan dastur

Program Misol;

Var

a,x,y:Real;

Begin

a:=2.3;

x:=3.1;

*y:=a*x;*

Writeln('y=',y);

End.

Noto‘g‘ri tuzilgan dastur

Program Misol;

Var

a,x,y:Real;

Begin

a:=2.3;

*y:=a*x;*

{o‘zlashtirish operatorining o‘ng tomonidagi “X” o‘zgaruvchining qiymati aniqlanmagan}

Writeln('y=',y);

End.

Mantiqiy o‘zlashtirish operatori

Agar o‘zlashtirish operatorining chap tomonidagi o‘zgaruvchi **boolean** (mantiqiy) turiga tegishli bo‘lsa, operatorning o‘ng tomonida natijasi **true** yoki **false** bo‘lgan mantiqiy ifoda bo‘lishi shart.

Mantiqiy ifoda - arifmetik ifoda, solishtirish belgilari va mantiqiy amallardan tashkil topadi. Mantiqiy ifodaning natijaviy qiymati **true** (rost) yoki **false** (yolg‘on) bo‘ladi.

Mantiqiy ifodada amallarning bajarilish tartibi quyidagicha:

1. Not
2. *, /, div, mod, and
3. +, -, or
4. =, <, >, <=, >=, <>

Mantiqiy ifodada ham amallar ketma-ketligini o‘zgartirish uchun kichik qavslardan foydalaniladi.

Mantiqiy ifodaga doir misollar:

1. $x < 2 * y$
2. **true**
3. **not not d**
4. $(x > y / 2)$
5. **d and (x=y) and b**
6. **(C or D) and (x=y) or not B**

Mantiqiy o‘zlashtirish operatoriga doir misollar:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
d:=true;  
b:=(x>y) and (k=0);  
c:=D or B and true;
```

```
VAR Global Flag:Boolean;  
FUNCTION GETSQR( x:real );  
Const SQRMAX=100;  
Begin  
  X:=x*x;  
  GlobalFlag:=( x>SQRMAX );  
  If GlobalFlag then x:=SQRMAX;  
  GetSQR:=x;  
End;
```

Belgili o‘zlashtirish operatori

Agar o‘zlashtirish operatorining chap tomonida **char** (belgili) yoki String (qatorli) turdagi o‘zgaruvchi ko‘rsatilgan bo‘lsa, u holda operatorning o‘ng tomonida belgili ifoda bo‘lishi zarur. Belgili qiymatlar ustida faqatgina qo‘shish (ulash) amalinigina bajarish mumkin. Shuning uchun, belgili ifoda-belgili o‘zgarmas, belgili o‘zgaruvchi yoki belgili turli funksiya bo‘lishi mumkin.

Belgili o‘zlashtirish operatoriga misollar:

```
s:='+-';  
d:='*/';  
k:=s+d;  
p:='Turbo Pascal';
```

Program Mkollej;

Var

s1,s2:String;

Begin

s1:='oliy';

s2:=' ta'lim';

s2:=s1+s2;

Writeln(s2);

End.

Natija:

oliy ta'lim

1- misol : Kvadrat tenglamani barcha hollar uchun to‘liq hisoblash dasturini tuzaylik.

$$ax^2+bx+c=0$$

$$D=\sqrt{b^2 - 4ac}$$

$$X_1=\frac{-b+\sqrt{d}}{2a}$$

$$X_2=\frac{-b-\sqrt{d}}{2a}$$

Program misol1;

Var a,b,c, x,x1,x2,d:real;

Begin

Read(a,b,c);

If d:=sqrt(b*b-4*a*c);

If d>0 then begin

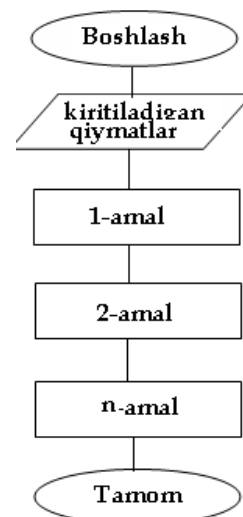
X1:=(-b+sqrt(d))/(2*a);

X2:=(-b-sqrt(d))/(2*a);

Writeln ('x1=',x1);

Writeln('x2=',x2); goto 4;

If d:=0 then



“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Begin  
X1:=-b/2*a;  
Writeln('x1=',x1); end  
Else  
Writeln( 'yechim yoq')  
4:End.
```

3. Paskal tilida chiziqli jarayonlarni dasturlash

Ta’rif: Dasturdagi bo‘yruqlar navbat bilan ketma-ket bajarilsa bunday dasturlar chiziqli dasturlar deb ataladi.

Chiziqli tuzilishga ega bo‘lgan algoritmlarda ko‘rsatmalar yozilish tartibida bajariladi. Ularning blok - sxemasini ishga tushirish, to‘xtatish, kiritish-chiqarish jarayoni bloki hamda avvaldan ma'lum jarayon bloklari yordamida tuzilib, bir chiziq bo‘ylab ketma-ket joylashgan bo‘ladi. Ya’ni hech qanday shart talab qilmaydigan algoritmga chiziqli algoritm deb qariladi. Uni blok sxemasi quydagicha bo‘ladi.

Chiziqli tuzilishdagi algoritmni tuzish masalani yechish uchun kerak bo‘ladigan boshlang‘ich ma'lumotlarni tashkil qiluvchi o‘zgaruvchilar nomi, ularning turi va o‘zgarish ko‘lamini aniqlashdan boshlanadi. Keyin oraliq va yakuniy natijalar o‘zgaruvchilarining nomlari, turlari va mumkin bo‘lsa, o‘zgarish ko‘lamini aniqlash kerak. Endi algoritm mana shu boshlang‘ich ma'lumotlarni qanday qayta ishlab oraliq va yakuniy natijalarni olish kerakligini aniqlashdan iborat bo‘ladi. Buni tushinish uchun turli misollarda aks ettirish.

Misol 1. a va b sonlarining o‘rta arifmetigini hisoblash dasturini tuzaylik.

Echish: Demak, $c = \frac{a+b}{2}$

Dasturi:

```
Program urt_arif(input,output);
```

```
var a,b: integer;
```

```
begin
```

```
read(a,b);
```

```
c:=(a+b)/2;
```

```
write(c);
```

```
end.
```

Misol 2. Tomonlari mos ravishda a, b, c teng bo‘lgan ixtiyoriy ABC uchburchak yuzini hisoblash algoritmini tuzaylik.

Tomonlari ma'lum bo‘lganda ABC uchburchakning yuzini topish uchun Geron formulasidan foydalanamiz

$$S = \sqrt{p(p-a)(p-b)(p-c)}.$$

Bunda

$$p = (a+b+c)/2$$

uchburchakning yarim perimetri.

1. Boshlang‘ich ma'lumotlar: a, b, c uchburchak tomonlari. Shuning uchun $a, b, c \in R$ va $a > 0, b > 0, c > 0$, ya'ni a, b, c — o‘zgaruvchilar nomi; ular haqiqiy sonli qiymatlar qabul qiladi. Shuni e’tiborga olish lozimki, bu uchta son uchburchak tomonlarini ifoda qilishi uchun ularning istalgan biri qolgan ikkitasi yig‘indisidan katta bo‘lmasligi, ya'ni

$$a < b+c, b < a+c, c < a+b$$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

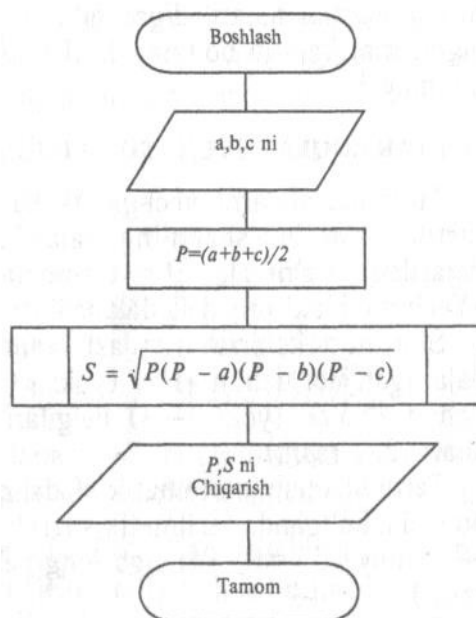
shartlar bajarilishi kerak. Shunday qilib, o‘zgarish ko‘lami yuqoridagi munosabatlar bilan aniqlanadi.

2. Natijalar: Berilgan formula bilan uchburchak yuzini hisoblash uchun uning yarim perimetrining qiymati kerak. Demak, p o‘zgaruvchining qiymati oraliq ma’lumot bo‘ladi. Yuqoridagi shartlarda $p \in R$ va $p > b$. Yakuniy natija: S — uchburchak yuzi. U $S \in R$ va $S > 0$ qiymatlar qabul qiladi.

Shunday qilib, ixtiyoriy ABC uchburchak yuzini EHMda hisoblash va bosmaga (yoki Display ekraniga) chiqarish

1. a, b, c - qiymatlarini EHM xotirasiga kiritish;
2. p ning qiymatini formula bilan hisoblash;
3. S ning qiymatini formula bilan hisoblash;
4. p va S larning qiymatlarini bosmaga chiqarish operatsiyalaridan iborat bo‘ladi.

Har qanday algoritmning blok-tarhi ishga tushirish blokidan boshlanadi. Uni EHMni ishga tayyorlash, boshlang‘ich ma’lumotlarni aniqlash va tayyorlash deb tushunish kerak. Hisoblashlarning tugaganligi ana shunday geometrik shakl bilan ko‘rsatiladi. Shuning uchun rasmdagi 1 va 6-bloklar ichiga mos kelgan operatsiyalar nomi yozib qo‘yilgan.



Boshlang‘ich ma’lumotlarni EHMga har xil qurilmalardan kiritish mumkin. Aniq bittasini tanlab olish ish sharoitiga bog‘liq. Shuning uchun umumiy kiritish-chiqarish bloklaridan (2- va 5-bloklar) foydalaniladi.

Uchinchi blokda bevosita hisoblash jarayoni, to‘rtinchi blokda esa kvadrat ildizdan chiqarish uchun tuzilgan kichik algoritim (yordamchi algoritim) dan foydalanish — avvaldan ma’lum jarayon ko‘zda tutilgan. Algoritim ko‘r-satmalari yozilish tartibida ketma-ket bajariladi. Ma’lumotlar blokdan blokka yuqoridan pastga uzatiladi. Shuning uchun ularni tutashtiruvchi chiziqqa ko‘rsatkichlar qo‘yilmagan.

Algoritmdan foydalanuvchi boshlang‘ich ma’lumotlarni berilgan shartlar bajariladigan qilib olishi kerak. Aks holda algoritimni bajarib bo‘lmaydi. U natijalilik xossasiga ega bo‘lmaydi.

Uchburchak yuzini topish dasturini PASKAL tilida tuzamiz.

```
Program uchburchak;  
Uses crt;  
Var a,b,c,p,S:real;  
Begin  
Write(`a= `); readln(a);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Write('b='); readln(b);
Write('c='; readln(c);
      P:=(a+b+c)/2;
      S:=sqrt(p*(p-a)*(p-b)*(p-c));
Writeln('Uchburchak yuzi S=',S,' ga teng');
End.
```

Dastur kodini kiritib [Ctrl+F9] klavishini bosish bilan dastur ishga tushadi. Natijani ko‘rish uchun [Alt+F5] tugmalar kombinatsiyasidan foydalaniladi va ekranda “a=”, “b=”, “c=” degan yozuv paydo bo‘ladi. Biz uchburchak tashkil qiladigan ixtiyoriy musbat 3 ta sonni kiritamiz va [Enter] tugmasidan foydalanib natija olamiz. Bu ishlarni paskal muhitida bajarish mumkin.

4. Paskal tilida shartli va shartsiz o‘tish operatorlari

Turbo Pascal tilidagi programmaning asosiy qismi operatorlar ketma-ketligidan iborat, har bitta operator berilganlar ustida amal bajaradi. *Operatorlarning buluvchisi* sifatida nuqta vergul belgisi ishlatiladi. Turbo Pascal tilidagi hamma operatorlar ikki guruhga bo‘linadi: sodda va strukturali.

Tarkibiga boshqa operatorlar kirmagan operatorlar sodda operatorlar deyiladi. Bunga o‘zlashtirish operatori, shartsiz o‘tish operatori, protsedurani chiqarish operatori va bo‘sh operatorlar kiradi.

O‘zlashtirish operatori (:=) o‘ng tomonda berilgan ifodani bajarishni va uning qiymatini chap tomonda turgan o‘zgaruvchiga tenglashtiradi. Ifoda va o‘zgaruvchining tipi bir xil bo‘lishi kerak.

Misol:

```
FuncKey := False;
Ch := 'G';
Sum := X + Y;
```

Shartsiz o‘tish operatori (goto) “...ga o‘tish” degan ma’noni anglatadi va u biror operator bajarilgandan keyin navbatdagi operatorni bajarishga emas balki boshqa biror belgi yordamida belgilangan operatorni bajarishda ishlatiladi. Eslatib o‘tamiz, belgi raqam yoki harf simbolidan iborat bo‘lishi mumkin.

Misol: GOTO 999;

```
GOTO EndBlock;
```

Goto operatori ishlatilganda belgi ta’sir qiladigan joy bu faqat shu operator yozilgan blok bo‘lishi mumkin. Boshqarishni boshqa blokka uzatish man etiladi.

Protseduraning chiqarish operatori foydalanuvchi tomonidan belgilangan protsedurani yoki standart protsedurani ishga tushirish uchun ishlatiladi. Masalan:

```
ClrScr;      {standart protsedurani chiqarish }
InitWork(True); {foydalanuvchi protsedurasini chiqarish }
```

Bo‘sh operator hech qanday amal bajarmaydi va uning tarkibida hech qanday simvollar yo‘q. Odatda bo‘sh operator lokal yoki global blokning oxiriga o‘tishda ishlatiladi:

```
LABEL Metka;
```

```
...
```

```
BEGIN
```

```
... GOTO Metka; {blok oxiriga o‘tish}
```

```
...
```

```
Metka:      {bo‘sh operatorga belgi bilan murojat qilingan}
```

```
END;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Murakkab operatorlar

Murakkab operatorlar qat’iyan belgilangan qoidalar bo‘yicha boshqa operatorlardan tuzilgan operatorlardir. Hamma strukturali operatorlar uch guruhga bo‘linadi:

tarkibiy, shartli, qaytariladigan.

Tarkibiy operator bu bir-biridan nuqta vergul belgisi va BEGIN va END operatorli qavslar yordamida ajratilgan operatorlar guruhidir:

```
BEGIN
  <operator;>
  ...
  <operator>
END;
```

Tarkibiy operator programmaning xohlagan tilning sintaksisi ruxsat beradigan qismida joylashishi mumkin.

Shartli operatorlar.

Shartli operator, shart, joylashtirilgan operatorlar.

Turbo Pascal tilida ikkita shartli operatorlar mavjud: IF va CASE. IF shartli operatori, operatorlarning bajarilish jarayonining tabiiy holatini o‘zgartiradigan eng ko‘p vositalardan biri. U quyidagi ko‘rinishlardan biriga ega bo‘lishi mumkin:

```
IF <shart> THEN <operator1>
  ELSE <operator2>;
IF <shart> THEN <operator>;
```

Shart - bu bulev tipdagi ifoda. Birinchi holatda, agar ifoda qiymati haqiqat bo‘lsa, <operator1> bajariladi, agar haqiqat emas bo‘lsa, <operator2> bajariladi. Ikkinchi holatda - agar ifoda natijasi True bo‘lsa, <operator> bajariladi, agar False bo‘lsa - IF operatoridan keyingi operator bajariladi. IF operatorlari *joylashtirilgan* bo‘lishi mumkin.

Misol:

```
Read(Ch);
  IF Ch='N' THEN Parol:= True ELSE Parol:= False;
Read(X);
  IF Parol = True THEN IF X = 100 THEN Write('Parol va kod to‘g‘ri.');
```

```
ELSE BEGIN
  Writeln ('Kodda xatolik mavjud.');
```

```
  Halt(1)
  END.
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Tanlash operatori. CASE mavjud variantlardan tanlash imkoniyatini beradi. U har biriga tanlash o‘zgarmlari ro‘yxati (ro‘yxat bitta o‘zgarmsdan iborat bo‘lishi mumkin) tegishli *selektor* deb nomlangan ifodadan va *parametrlar* ro‘yxatidan iborat.

Formati:

```
CASE <ifoda-selektor> OF
  <ro‘yxat 1>: <operator 1; >
  <ro‘yxat 2>: <operator 2; >
  ...
  <ro‘yxat N>: <operator N>
ELSE <operator>
END;
```

O‘zgarmlar tipi doim selektor tipiga to‘g‘ri kelishi kerak. Selektor uchun real va string tiplari man etilgan.

CASE operatori quyidagicha ishlaydi. Birinchi navbatda selektor-ifoda qiymati hisoblanadi, keyingi navbatda joriy selektor qiymatiga teng bo‘lgan o‘zgarms qatnashgan operator bajariladi. Agar hech qaysi o‘zgarms selektorning joriy qiymatiga teng bo‘lmasa ELSE so‘zidan keyingi operator bajariladi. Agar ELSE so‘zi bo‘lmasa END so‘zidan keyingi operator ishga tushadi, ya‘ni CASE chegarasidan keyingi operator.

Selektor butun sonli (-32768..32767 diapazonida bo‘lgan) bulev, liter yoki foydalanuvchi tipiga bog‘liq bo‘lishi kerak.

O‘zgarms qiymatlar ro‘yxati tasodifiy qiymat yoki diapazondan iborat, ular bir-biridan vergul yordamida ajratiladi. Diapazon chegaralari ikkita biri-biridan "." belgisi yordamida ajratilgan o‘zgarms sonlar yordamida yoziladi. O‘zgarmlar tipi selektor tipiga to‘g‘ri kelishi kerak.

Quyida CASE operatorining tipik yozilish tartibi ko‘rsatilgan:

Interval tipli selektor :

CASE I OF

```
1..10: Writeln ('raqam ', I:4, ' diapazon 1 - 10');
11..20: Writeln ('raqam ', I:4, ' diapazon 11 - 20');
21..30: Writeln ('raqam ', I:4, ' diapazon 21 - 30')
ELSE Writeln ('chislo ', I:4, ' kontrolya chegarasidan tashqarida')
END;
```

Butun son tipli selektor :

CASE I OF

```
1: Z := I + 10;
2: Z := I + 100;
3: Z := I + 1000
```

END;

Foydalanuvchi hisob tipli selektor:

VAR

Season: (Winter, Spring, Summer, Autumn);

BEGIN

... CASE Season OF

```
Winter: Writeln('Winter');
Spring: Writeln('Spring');
Summer: Writeln('Summer');
Autumn: Writeln('Autumn')
```

END;

END;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Paskal tilining takrorlash operatorlari

Yuqorida sanab o‘tilgan jarayonlardan biri, takrorlanuvchi jarayonlarni hisoblashni shartli operatorlardan foydalanib ham tashkil etsa bo‘ladi, lekin bunday jarayonlarni hisoblashni takrorlash operatorlari yordamida amalga oshirish osonroq kechadi.

Takrorlash operatorlarining 3 xil turi mavjud:

- parametrli takrorlash operatori;
- repeat takrorlash operatori;
- while takrorlash operatori.

Yechilayotgan masalaning mohiyatiga qarab, dastur yozuvchi o‘zi uchun qulay bo‘lgan takrorlash operatorini tanlab olishi mumkin.

Operatorning quyidagi ko‘rinishdagisi amalda ko‘proq ishlatiladi:

for k:= k1 to k2 do S;

bu yerda *for* (uchun), *to* (gacha), *do* (bajarmoq) - xizmatchi so‘zlari;

k - sikl parametri (haqiqiy turli bo‘lishi mumkin emas);

k1 - sikl parametrining boshlang‘ich qiymati;

k2 - sikl parametrining oxirgi qiymati;

S - sikl tanasi.

Operatorning ishlash prinsipi:

sikl parametri (sp) boshlang‘ich qiymat k1 ni qabul qilib, agar bu qiymat k2 dan kichik bo‘lsa, shu qiymat uchun S operatori bajariladi;

sp ning qiymati yangisiga o‘zgartirilib (agar k son bo‘lsa o‘zgarish kadami 1 ga teng, belgili o‘zgaruvchi bo‘lsa navbatdagi belgini qabul qiladi, va h.k.) yana S operatori bajariladi va bu jarayon $k > k2$ bo‘lguncha davom ettiriladi. SHundan so‘ng, sikl operatori o‘z ishini tugatib boshqarishni o‘zidan keyingi operatorga uzatadi.

Agar biz operatorlarning necha marta takroran hisoblanishini aniq bilsak, u xolda parametrli takrorlash operatoridan foydalanish maqsadga muvofiqdir.

Misol: $\sum_{i=1}^n \frac{1}{i}$ yig‘indini chekli n ta hadining yig‘indisini topish dasturini tuzish.

Program sum1;

var

S: real;

i,n: byte; {i va n o‘zgaruvchilar 255 dan katta bo‘lmagan, butun, natural sonlar}

begin

readln (n); S:= 0;

for i:=1 to n do

S:= S + 1/i;

writeln (S);

end.

Ayrim paytlarda, sikl parametrini o‘sib borish emas, balki kamayish tartibida o‘zgartirish mumkin, bu holda sikl operatori quyidagi formada yoziladi:

for k:= k2 downto k1 do S;

bu yerda *down to* (gacha kamayib) – tilning xizmatchi so‘zi.

Bu operatorda k parametri k2 dan toki k1 gacha kamayish tartibida (agar k - butun qiymatli o‘zgaruvchi bo‘lsa sikl qadami - 1 ga teng) o‘zgaradi. Operatorning ishlash prinsipi oldingi operatornikiday qolaveradi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Misol. YUqorida ko‘rsatilgan misolni dasturini qaytadan tuzaylik. Bu holda dasturdagi sikl operatorigina o‘zgaradi xolos:

for i:=n downto 1 do

qolgan operatorlar esa o‘z o‘rnida o‘zgarmay qoladi.

Dasturda parametrli takrorlash operatoridan foydalanish jarayonida, sikl parametrining qiymatini sikl tanasi ichida o‘zgartirmaslik lozim, aks holda operatorning ish ritmi buzilishi mumkin. Buni quyidagi misollarda ko‘rish mumkin:

To‘g‘ri tuzilgan dastur qismi

```
for i:=1 to 10 do  
Begin  
s:=i*i;  
writeln(s);  
end;
```

Noto‘g‘ri tuzilgan dastur qismi

```
for i:=1 to 10 do  
Begin  
s:=i*i;  
writeln(s);  
i:=i+3  
end;
```

Ma’lum bir jarayonlarning takrorlash parametrlari haqiqiy qiymatlar qabul qilishi mumkin, bu holda parametrli takrorlash operatoridan to‘g‘ridan-to‘g‘ri foydalanib bo‘lmaydi. Quyidagi misolda bunday takrorlashlarni qanday tashkil qilish mumkinligini ko‘ramiz:

Misol: $y=e^x$ funksiyasini $[-2,2]$ oraliqdagi «x» lar uchun hisoblash dasturini tuzing («x» ning o‘zgarish qadami 0,5 ga teng deb hisoblansin).

Funksiyani necha marta hisoblash kerakligini $N=$ formula bilan aniqlaymiz.

```
Program Function;  
Var x:real;  
      y:real;  
      i:integer;  
begin  
  x:=-2;  
      for i:=1 to 9 do  
      begin  
        y:=exp(x);  
        writeln(x,y);  
        x:=x+0.5  
      end
```

```
end  
end.
```

2. Repeat takrorlash operatori

YUqorida aytib o‘tganimizdek, sikldagi takrorlanishlar soni oldindan ma’lum bo‘lsa, parametrli (**for**) sikl operatori foydalanish uchun juda qulay. Lekin, ko‘pgina hollarda, takrorlanuvchi jarayonlardagi takrorlanishlar soni oldindan ma’lum bo‘lmaydi, sikldan chiqish esa ma’lum bir shartning bajarilishi yoki bajarilmasligiga bog‘lik holda bo‘ladi. Bu hollarda **repeat** yoki **while** sikl operatorlaridan foydalanish zarur. Agar sikldan chiqish sharti, takrorlanuvchi jarayonning oxirida joylashgan bo‘lsa **repeat** operatoridan, bosh qicmida joylashgan bo‘lsa **while** operatoridan foydalanish maqsadga muvofiqdir.

Repeat operatorining yozilish formasi quyidagicha bo‘ladi:

repeat S1; S2; ... SN until B;

bu yerda **repeat** (takrorlamoq), **until** (gacha) - xizmatchi so‘zlar;

S1, S2, ..., SN lar esa sikl tanasini tashkil etuvchi operatorlar;

B - sikldan chiqish sharti (mantiqiy ifoda).

Operatorning ishlash prinsipi juda sodda, ya’ni siklning tanasi B mantiqiy ifoda rost qiymatli natija bermaguncha takror - takror hisoblanaveradi. Misol sifatida, yana yuqoridagi yig‘indi hisoblash misolini olaylik.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Program Sum2;  
var i, n: Byte;  
    S: real;  
begin  
    readln(n);  
    S:=0; i:=1;  
    repeat  
        S:= S+1/i;  
        i:=i+1;  
    until i>n;  
    writeln (S)  
end.
```

Ayrim takrorlanish jarayonlarida sikldan chiqish shartini ifodalovchi mantiqiy ifoda hech qachon True (rost) qiymatga erishmasligi mumkin. Bu xolda dasturning takrorlash qismi cheksiz marta qaytadan hisoblanishi mumkin, ya'ni dasturchilar tili bilan aytganda «**dastur osilib qoladi**» shuning uchun, operatordagi shartni tanlashda e'tiborli bo'lish lozim. E'tiboringizga ya'na bir, ismni qidirib topish dasturini xavola qilamiz:

Program BRV;

```
Var  
    a,b:String[20];  
Begin  
    a:='Jamshid';  
    Repeat  
        Writeln("Tanlagan ismingizni kiriting");  
        Readln(B);  
        if a<>b Then writeln('Noto'g'ri') else writeln('YAshang to'g'ri topdingiz');  
    Until A=B;  
End.
```

3. While takrorlash operatori

Ahamiyat bergan bo'lsangiz, repeat operatorida siklning tana qismi kamida bir marta hisoblanadi. Lekin, ayrim paytlarda, shu bir marta hisoblash ham yechilayotgan masalaning mohiyatini buzib yuborishi mumkin. Bunday hollarda, quyidagi formada yoziluvchi while sikl operatoridan foydalanish maqsadga muvofiqdir:

while B do S;

bu yerda **while** (hozircha), **do** (bajarmoq) - xizmatchi so'zlari;

B - sikldan chiqishni ifodalovchi mantiqiy ifoda;

S - siklning tanasini tashkil etuvchi operator.

Bu operatorda oldin V sharti tekshiriladi, agar u **false** (yolg'on) qiymatli natijaga erishsagina sikl o'z ishini tugatadi, aks holda siklni tana qismi qayta - qayta hisoblanaveradi.

While operatoriga misol sifatida, yana yuqorida berilgan yig'indi hisoblash misolini ko'rib chiqaylik:

```
program sum3;  
var i, n: byte;  
    S: real;  
begin  
    readln(n);  
    i:=1; S:= 0;  
    while i<=n do  
        begin  
            S:= S + 1/i;  
            i:= i+1;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

end;
writeln (S)

end.

4. Bo‘sh operator

Bu operator o‘zidan keyingi operatorni aniqlab beradi xolos. Operatorlar ketma-ketligi orasida boshqa operatorlardan ";" belgisi bilan ajratilib turiladi. Bundan tashqari, bo‘sh operator metka bilan jixozlangan ham bo‘lishi mumkin.

Misol:

1. **begin** L1;; k:=5; M:=k+6; **end.**
2. **begin** M:=5; k:=M-2.7; L4: **end.**

Ayrim paytlarda, ba’zi bir operatorlarga bir nechta metka bilan murojaat qilishga to‘g‘ri kelganda bo‘sh operatoridan foydalanish qo‘l keladi.

S5;; S6;; S7: x:=0.5;

Punktuatsiya qoidalari

Operatorlarni yozishda quyidagi punktuatsiya qoidalariga roiya qilish kerak:

1. Nuqta vergul belgisi UNIT, USES, LABEL, TYPE, CONST, VAR rezervlangan so‘zlaridan keyin quyilmaydi va har bir opisaniyedan keyin quyiladi.
2. Nuqta vergul belgisi BEGIN so‘zidan keyin va END so‘zidan oldin quyilmaydi, chunki bu so‘zlar operator emas, balki operator qavslaridir.
3. Nuqta vergul operatorlarni ajratishi uchun xizmat qiladi, agar u operatorlar orasida quyilmasa kompilyatsion xato vujudga keladi.
4. Sikl operatorlarida nuqta vergul belgisi WHILE, REPEAT, DO lardan keyin va UNTIL dan oldin qo‘yilmaydi.
5. Shartli operatorlarda nuqta vergul THEN dan keyin va ELSE dan oldin quyilmaydi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

MA’RUZA № 7

MAVZU: PASKAL DASTURLASH TILIDA MASSIVLAR BILAN ISHLASH.

REJA:

1. Paskal dasturlash tilida massivlar bilan ishlash.
2. Bir o‘lchovli massivlar.
3. Ikki o‘lchovli massivlar va ularga doir masalalar.

Paskal dasturlash tilida massivlar bilan ishlash.

Umumiy nomga ega bo‘lgan bir xil ko‘rinishda tartiblangan elementlar ketma-ketligi massiv deyiladi. Massiv elementlari uning komponentlari deb ataladi. Komponenta tipi – baza tipi hisoblanadi. Har bir tip o‘zining indeksiga va nomiga ega,ular qavs ichiga keltiriladi.

Massiv elementlari ixtiyoriy tipda ,hattoki ma’lumotlar ham bo‘lishi mumkin. Massiv elementlarining tiplari bazali deyiladi. Massiv elementlarining soni programma ishlash jarayonida o‘nga ozlashtirib boriladi. Uning har bir alohida elementiga murojaat massiv elementlariga mos kelgan indeksleri bo‘yicha bo‘ladi. Massiv indeksi tushunchasi xuddi vektorlar indeksi tushunchasi kabi bo‘ladi Massivlarni e’lon qilish uchun **Array** of (massivda) so‘z birligi ishlariladi.

Yozilishi:

TYPE

<tip nomi>=array[indeks tipi] of <komponenta tipi>;

VAR

<identifikator,...>:<tip nomi>;

Massivlarni tiplarni e’lon qilmasdan turib ham qo‘llash mumkin.

VAR

<>identifikator,...:array[indeks tipi] of <komponenta tipi>;

Misol;

TYPE

Klass=(k1,k2,k3,k4);

Znak=array[1..255] of char;

VAR

M1: Znak;{Znak tipi tiplar bo‘limida oldindan keltirilgan}

M2: array[1..60] of integer;{M2 massivning yozilishi}

M3: array[1..4] OF Klass;

Mas: array[1..4] of integer;

Massivning baza tipi har qanday tip bo‘lishi mumkinligi sababli u boshqa massiv ham bo‘lishi mumkin.Natijada,Ko‘p o‘chamli massiv hosil bo‘ladi.

Masalan.

TYPE

Vector=array[1..4] of integer;

Massiv=array[1..4] of vetor;

VAR

Matr : Massiv;

Xuddi shu strukturani boshqa turdagi yozuvni qo‘llash natijasida ham hosil qilish mumkin.:

VAR

Matr: array[1..4,1..4] of integer;

Massiv bilan ishlashda konstantalar ham ishlatilishi mumkin.

CONST

G1=4 ; G2=6;

VAR

MasY : array[1..G1,1..G2] of real;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Massiv elementlari xotirada ketma-ket joylashadi. Indeksleri kichkina bo‘lgan elementlar xotoraning pastki indekslarida saqlanadi. Ko‘p elementli massivlarda eng o‘ng tarafidagi indeks birinchi bo‘lib o‘tib boradi. Masalan ; Agarda

A:array[1..5,1..5] of integer;

Bo‘lsa u holda massiv elementlari adreslarnig o‘shishi bo‘yicha joylashadi.

A[1,1]

A[1,2]

.....

A[1,5]

A[2,1]

A[2,2]

Massivlarni bir butun holda ishlatilgan paytda massiv nomlaridagi indekslardagi kvadrat qavsga olinmay ishlatiladi. Massivlar kiritish operatorida “teng” yoki “teng emas” operatsiyalarida ishlatilishi mumkin

Bu amallarda ishlatiladigan massivlar bir xil tipdagi indeks va komponentalarda ega bo‘lib strukturasi bir biriga o‘xshash bo‘lishi kerak.

Masalan,

A va B massivlar Var A,B: array[1..20] of real;

Ko‘rinishida ifodalangan bo‘lsa unda natija quyidagicha bo‘ladi.

A=B True, agarda **A** massivning elementlarining qiymatlari **V** massiv elementlarining qiymatlariga mos ravishda teng bo‘lsa.

A<>V True, agarda **A** massiv elementining bron bir qiymati **V** massiv elementining qiymatlariga mos ravishda mos bo‘lsa.

A :=V **V** massiv elementining hamma qiymatlari **A** massiv elementlari tomonidan o‘zlashtirilsa **V** massiv elementlarining qiymatlari o‘zgarmasdan qoladi.

Massiv e‘lon qilingandan keyin uni elementlarini nomlari qavs ichida ko‘rsatilgan holda ishlatish mumkin.. Masalan: **mas[2]**, **vektorZ[10]** massivning ikkinchi va uchinchi elementlariga murojaat etishni bildiradi.

Ikki o‘lchovli massivlar ikita indeks n o‘lchovlilarida esa n ta indeks ko‘rsatiladi Masalan, **MatrU[4,4]** bu yozuv **MatrU** massivning 4 ta qator 4 ta ustundagi elementini bildiradi. Array tipidagi qiymatlar bilan ishlashda quyidagi holatlar bo‘lishi mumkin:

VAR

A,D : array[1..4] of real;

B:array[1..10,1..15] of integer;

I,J: integer;K:integer;S:real;

Bu operatsiyani FOR operatori yordamida ham bajarish mumkin:

FOR I: =1TO 4 Do A[I]:=0;

Ikki o‘lchovli massivlarga indeks qo‘yish uchun ichma-ich joylashgan operatorlar ishlatiladi:

FOR I:=1TO 10 DO

FOR J:=1 TO 15 DO

B[I,J]:=0;

Paskal algoritmik tilida massiv elementlarini birdaniga kiritish-chiqarish imkoniyati yo‘qligi sababli elementlar bittadan kiritiladi. Massiv elementiga qiymatni o‘zlashtirish operatori yodamida beriladi, initsializatsiya misolda ko‘rsatilgandek ,lekin ko‘p hollarda **READ** yoki **READLN** operatori yordamida o‘zlashtiriladi va sikl operatoridan foydalanamiz.

FOR I:= 1 TO 10 Do

FOR J:=1 TO 15 DO

READLN(D[I,J]);

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

WRITELN operatori ishlatilishi sababli har bir qiymat yangi qatordan kiritiladi. Alohida elementlarni qiymatlarini ham kiritish mumkin. Bunda quyidagi operatorlardan foydalaniladi:

READ (A[3]);

READ (B[6,9]);

Bu yerda A vektoriga 3ta element qiymati va V matritsaning 6- qator 9-ustuniga joylashgan elementlari qiymatlarini kiritadi.

Ikkala qiymat ham ekranning bitta qatorida ,kursirning joriy pazitsiyasidan teriladi. Massiv elementlari qiymatini chiqarish ham xuddi shunday bajariladi, lekin bunda WRITE yoki WRITELN operatori qo‘llaniladi.

FOR I:=1 TO 4 DO

Writeln (A[I]); {A massiv elementlari qiymatini chiqarish}

Yoki

FOR I:=1 TO 10 DO

FOR J:=1 TO 15 DO

Writeln (V[I,J]); {B massiv elementlari qiymatini chiqarish}

1-Misol:

5 ta elementdan hosil bo‘lgan X massivning elementlarini kiriting.

Har bir elementning kvadrati va ildizini ekranga chiqaring.

Yechish:

PROGRAM P7_2;

VAR

Sum,I:integer;

Sr:real;

X:array [1..5] of integer;

Kor,Kv:array[1..5] of real;

Begin

Sum:=0;

Writeln(‘5 ta butun qiymatlarni kiriting:’);

FOR I:=1 TO 5 DO BEGIN

Write(‘I-sonni kiriting: ‘);

Readln (X[I]);

Kor[I]:= sqrt(X[I]);

Kv[I]:=sqr(X[I]);

END;

FOR I:=1 TO 5 DO Write(Kor[I]:8:2);

Writeln;

FOR I:=1 TO 5DO Write (Kv[I]:8:2);

Readln;

END.

2-misol:

3x3 massiv elementlarini kiriting va har bir qator yig‘indisini hisoblang.

Yechish:

PROGRAM P7_8;

VAR

Xarray[1..3,1..3] of integer;

Sum:array[1..3] of integer;

I,J:integer;

BEGIN

Randomize;

FOR I=1 TO 3 DO

FOR J:=1 TO 3 DO BEGIN

X[I,J]:=random(300);

END;

FOR I:=1 TO 3DO

Sum[I]:=0

FOR I:=1 TO 3 DO

FOR J:=1 TO 3 DO

Sum[I]:=Sum[I]+X[I,J];

FOR I:=1 TO 3 DO

Write(Sum[I]:3);

Readln;

END.

Massivdan nusxa olish deb, bitta massiv elementlari qiymatlarini boshqa massiv elementlarini o‘zlashtirishiga aytiladi. Nusxa olish bitta o‘zlashtirish operatori yordamida amalga oshirish mumkin,masalan **A:=D** yoking **FOR** operatori yordamida .

FOR I: =1TO 4 DO A[I]:=D[I];

Ikkala holda ham massiv elementlari qiymatlari o‘zgaraydi, A massiv elementlarining qiymatlari D massiv elementlari qiymatiga teng bo‘lib qoladi. Ko‘rinib turibdiki ,ikkala massiv ham strukturasi bo‘yicha bir-biriga o‘xshash. Ko‘pchilik holatlarda massivda qaysidir elementlarni izlashga to‘g‘ri keladi. Masalan: A massivning nechta elementi nol qiymatga ega

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

ekanligini bilish talab etiladi. Buning uchun qo‘shimcha o‘zgaruvchi K ni kiritamiz va FOR , IF operatorlarida foydalanamiz.

```
K:=0;
```

```
FOR I:=1 TO 4 DO
```

```
IF A[I]=0 THEN K:=K+1;
```

Sikl bajarilgandan keyin K o‘zgaruvchi A massivning nolga teng bo‘lgan qiymatlarini o‘z ichiga oladi. Massiv elementlari qiymatlarini joyini almashtirish massivning bazali tipiga o‘xshash tipdagi yordamchi o‘zgaruvchi yordamida amalga oshiriladi. Masalan, A massivning birinchi va beshinchi elementlari qiymatlarini joyini almashtiring.

```
Vs:=A[5]
```

```
A[5]:=A[1];
```

```
A[1]:=Vs;
```

Ma’ruza № 8

MAVZU: PASKALDA SATRIY KATTALIKLAR BILAN ISHLASH.

REJA:

1. String tipidan matnli ma’lumotlar
2. Satriy kattaliklar ustida funksiya va proseduralar yordamida bajaraladigan amallar
3. Satriy kattalikka doir masalalar.

Turbo Pascal dasturlash tilida String tipidan matnli ma’lumotlar bilan ishlashda foydalaniladi. Bu tipning har bir elementi xotirada 1 bayt joy egallovchi belgilardan iborat bo‘ladi. Satriy o‘zgaruvchida belgilar miqdori 0 dan 255 tagacha bo‘lishi mumkin. Satriy kattalik o‘zgaruvchilar (Var) bo‘limida quyidagicha e’lon qilinishi mumkin.

```
var d:string;
```

```
satr:string [15];
```

Bu yerda d –kattalikning uzunligi 0 dan 255 tagacha, satr-kattalikning uzunligi esa 0 dan 15 gacha bo‘lishi nazarda tutilgan. Har qanday satriy tipidagi kattalikda birinchi baytning indeksi 0 ga teng bo‘lib, unda satriy kattalik uzunligi haqidagi ma’lumot joylashadi. Satriy kattalikning birinchi belgisi ikkinchi baytni egallaydi va uning indeksi birga teng.

Satrlar miqdorlaning elementlari kattalik nomi va kvadrat qavslarga olingan elementning tartib nomeri yordamida aniqlanadilar.

Misol:

```
Satr:='Geologiya' bo'lsa, satr[1]='G'-ga, satr[2]='e'-ga va h.
```

Quyidagi dastur yordamida satriy kattalikning uzunligini uning birinchi baytiga ko‘ra, ya’ni 0-indeks orqali aniqlaymiz.

```
var t:string;i:byte;
```

```
begin
```

```
t:='informatika';i:=ord(t[0]);writeln(i);end.
```

Dasturdagi ord(t[0]) protsedurasi t-kattalikning uzunligini aniqlashda yordam beradi. Bu qiymatni Length(t) funksiyasi yordamida ham aniqlash mumkin. Uzunlik qiymati mazkur kattalik uchun 11-ga tengligi dastur natijasidan aniqlanadi. Quyidagi dasturni yordamida satriy kattalikning har bir elementini alohida «ko‘rishimiz» mumkin:.

```
var t:string; i,k:byte;
```

```
begin
```

```
t:='informatika';i:=ord(t[0]);
```

```
for k:=1 to i do
```

```
writeln(k,t[k]);
```

```
end.
```

Endi satriy kattaliklar ustida ayrim funksiya va proseduralar yordamida bajaraladigan amallar bilan tanishamiz:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Concat($s_1, [s_2, \dots, s_N]$) – funksiyasi (tipi String) s_1, s_2, \dots, s_N satriy kattaliklarni «yig‘indisini» beradi (bu funksiya vazifasini ‘+’ amali ham bajaradi)

Misollar keltiramiz:

Ifoda	Natija
‘r1+’+’r2+’+’r3+’+’...+’+’rn’	‘r1+r2+r3+...+rn’
Concat(‘Rezistor ‘, ‘-qarshilik ‘, degan ma’noni ‘, anglatadi’)	‘Rezistor-qarshilik degan ma’noni anglatadi’

Copy(st, t1, d) – funksiyasi (tipi String) st satriy kattalikning t1 tartib raqamli belgisidan d-ta simvolni (t1-dan boshlab) nusxalash imkonini beradi.

Misollar keltiramiz:

St-qiymati	Ifoda	Natija
‘Elektr qarshilik o‘tkazgich uzunligiga bog‘liq’	Copy(St,8,10)	‘qarshilik’
‘Mol-modda miqdori’	Copy(St,11,6)	‘miqdor’

Delete(st, t1, d) - protsedurasi st satriy kattalikning t1 tartib raqamli belgisidan d-ta simvolni (t1-dan boshlab) o‘chirish imkonini beradi. Delete protsedurasidan foydalanganda uning uzunligi kamayadi.

Misollar keltiramiz:

St-qiymati	Operator	Natija St-ning yangi qiymati
‘abcdefg’	Delete(S,3,2)	‘abefg’
‘abcdefg’	Delete(S, 2,6)	‘a’

length(st) - funksiyasi (tipi Integer) st satriy kattalikning uzunligini aniqlaydi.

Misollar keltiramiz:

St-qiymati	Ifoda	Natija
‘Zomin tog‘-o‘rmon qo‘riqxonasi’	Length(S t)	30
‘’’	Length(S t)	1
‘mediana’	Length(S t)	6
‘(a+b)*c’	Length(S t)	7

Pos(st1, st) - funksiyasi (tipi Integer) st satriy kattalikning tarkibida st1 satriy kattalik “joylashgan” bo‘lsa, uning tartib raqamini beradi, aks holda 0 qiymatni beradi. St1 satriy kattalik bir necha marotaba uchragan holda ham birinchisining tartib raqamini beradi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Misollar keltiramiz:

st1 qiymati	Ifoda	Natija
‘abcdef’	Pos(‘cd’,S2)	3
‘abcdcdef’	Pos(‘cd’,S2)	3
‘abcdef’	Pos(‘k’,S2)	0

Val (a2, x,d) - protsedurasi a2 satriy kattalikni mumkin bo‘lgan holda, x-ga haqiqiy yoki butun tipidagi son sifatida, aks holda x-ga 0-qiymatini o‘zlashtiradi.

Misollar keltiramiz:

1-dastur	Natija	Hulosa
<pre> program val1; var a2:string;d,x:integer; begin a2:='rrr'; Val(a2,x,d); writeln(x); end. </pre>	0	A2-satriy miqdorni butun x-soniga o‘zlashtirishida uni son ko‘rinishida ifodalay olmasligi tyfayli, x-ga 0 qiymatni o‘zlashtirdi
2-dastur	Natija	Xulosa
<pre> program val2; var a2:string;d,x: integer; begin a2:='2011';Val(a2,x,d); writeln(x/20:4:1); end. </pre>	100.6	A2-satriy miqdorni butun x-soniga o‘zlashtirdi, uni ko‘rsatilgan formatda 20 ga bo‘lib chop

Quyidagi proseduralarni mustaqil o‘rganing va misollar ko‘rib chiqng:

Insert (st1, st, t1) - protsedurasi st1 satriy kattalikning t1 tartib raqamli belgisidan st satriy kattalik tarkibiga joylashtiradi.

Str (x, st) – protsedurasi x haqiqiy yoki butun tipidagi son bo‘lsa, uni satriy kattalikga aylantiradi.

Yana ikkita satriy kattalik bilan =, <>, <,>, >=, =< munosabat amallari ham bajariladi. Bu munosabatlar ikkita satriy miqdorlarni taqqoslash uchun ishlatiladi. Taqqoslash natijasi *true* yoki *false* qiymatlarni qabul qiladi va bu amal Concat amaliga nisbatan past darjali hisoblanadi. Satriy miqdorlarni taqqoslash quyidagicha bajariladi: chapdan o‘ngga elementlar taqqoslanib boriladi va bu teng indeksli elementlar “mos” kelmaguncha davom ettiriladi. Mos kelmagan elementlarning qaysi biri ASCII jadvalida katta qiymatli indeksga ega bo‘lsa, shu satriy miqdor katta hisoblanadi. Agar satriy miqdorlarning uzunligi har xil bo‘lib, bosh qismlari bir xil elementlardan iborat bo‘lsa, ikkita satriy miqdordan “uzuni” katta hisoblanadi. Bir xil uzunlikka ega va mos indeksli belgilari bir xil bo‘lgan satriy miqdorlar o‘zaro teng hisoblanadilar.

Misollar:

Ифодалар	Натижа
‘True1’<’True2’	True
‘Mother’>’MOTHER’	True

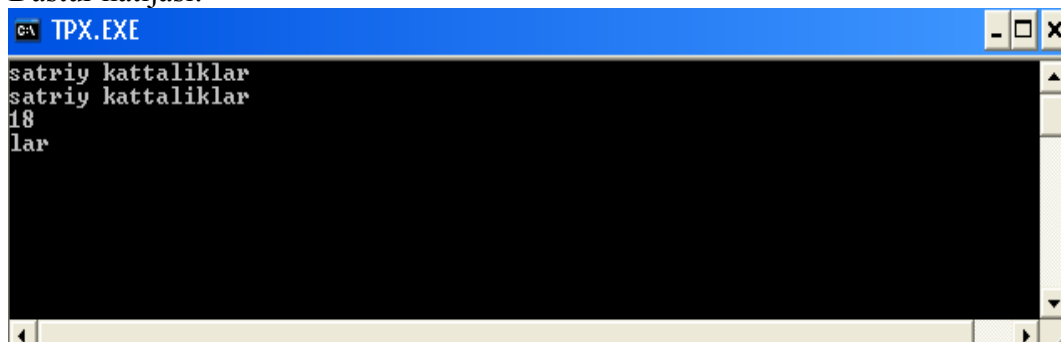
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

‘Geografiya’>‘Geologiya’	False
‘Cat’=‘Cat’	True
‘uzunlik’>‘balandlik’	True

Yuqorida keltirilgan funksiya va protseduralardan dasturlarda qo‘llashga miollar keltiramiz:

```
1-misol.  
uses crt;  
var  
s1,s2,s3,s4,s5:string;  
begin clrscr;  
s1:='satriy';  
s2:='kattaliklar';  
s3:=concat(s1,' ',s2);  
writeln(s3);  
s4:=s1+' '+s2;  
writeln(s4);  
writeln(length(s3));  
s5:=copy(s3,length(s3)-2,3);  
writeln(s5);  
end.
```

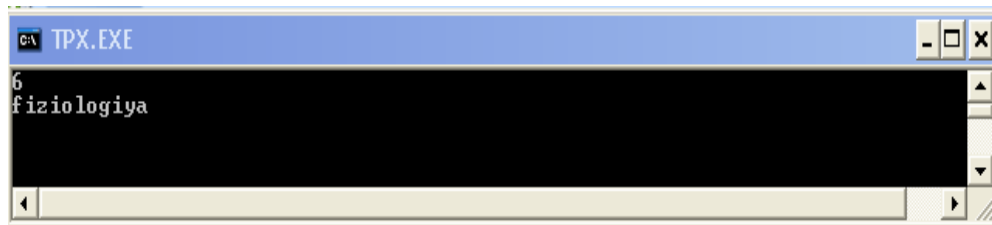
Dastur natijasi:



```
TPX.EXE  
satriy kattaliklar  
satriy kattaliklar  
18  
lar
```

```
2-misol.  
uses crt;  
var  
s1,s2,s3,s4,s5:string;  
i:integer;  
begin clrscr;  
s1:='SAMARQAND';  
i:=pos('QAND',s1);  
writeln(i);  
s2:='biologiya';  
s3:='fizika';  
delete(s2,1,2);  
delete(s3,5,2);  
insert(s2,s3,5);  
writeln(s3);  
readln;  
end.  
dastur natijasi:
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



Kiritilgan ixtiyoriy jumla tarkibidagi “A” belgisini “O” belgisiga almashtirish chop etuvchi dastur tuzing.

1-usul	2-usul
<pre> PROGRAM sim1; VAR jumla:string;i,d: INTEGER; SIM:STRING; BEGIN read(Jumla);d:=length(jumla); writeln(d); FOR I:=1 TO D DO BEGIN IF COPY(JUMLA,I,1)='A' THEN JUMLA[I]='O'; END; WRITELN(JUMLA); readln; END. </pre>	<pre> PROGRAM sim1; VAR jumla:string;i,d: INTEGER; SIM:STRING; BEGIN read(Jumla);d:=ord(jumla[0]); writeln(d); FOR I:=1 TO D DO BEGIN IF JUMLA[I]='A' THEN JUMLA[I]='O'; END; WRITELN(JUMLA); readln; END. </pre>

Burchak qiymati gradus o‘lchovida kiritilgan holat uchun sinus, kosinus, tangens funksiyalari qiymatini mos belgini tanlash yo‘li bilan hisoblovchi dastur tuzing.

```

PROGRAM char1;
  uses crt;
  VAR sim:char; i,grad:INTEGER;javob,radian:real; label 15,20;
  BEGIN
    writeln('burchak qiymatini gradus o‘lchovida kiriting');
    readln(grad); radian:=(pi*grad)/180;
    writeln('sinus ', grad,' ni hisoblash uchun "s" tugmasini tanlang');
    writeln('kosinus ', grad,' ni hisoblash uchun "c" tugmasini tanlang');
    writeln('tangens ', grad,' ni hisoblash uchun "t" tugmasini tanlang');
    sim:=readkey;
    case sim of
      's': javob:=sin(radian);
      'c': javob:=cos(radian);
      't': if cos(radian)=0 then goto 15 else javob:=sin(radian)/cos(radian);
    end;
    WRITELN(javob:5:2);goto 20;
    15:writeln ('qiymatimavjud emas');
    20: readln;
  END.

```

Tarkibida raqamlar bo‘lgan s1-satriy kattalikning barcha raqamlridan yangi s2 –satiy kattalik yaratuvchi dastur tuzing.

Program Satriy_kattaliklar;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

```
Var s1, s2: string; i: byte;  
begin  
  writeln('Tarkibida raqamlar bo‘lgan satriy kattalikni kiriting');  
  readln(s1);  
  s2:='';  
  for i:=1 to length(s1) do  
    if (s1[i]>='0') and (s1[i]<='9')  
      then s2:=s2+s1[i];  
  writeln('natija ',s2);  
end.
```

Dasturni quyidagi qiymat uchun tekshirib to‘g‘ri tuzilganiga ishinch hosil qiling:
S1:=’Balandligi 375 m bo‘lgan poytaxtdagi teleminora’;
Natija: 375

Masala

S-satriy miqdorning boshida joylashgan barcha bo‘shliq belgilarini o‘chiruvchi dastur tuzing.

```
Program Sinov1;  
Var s: string[80];  
Begin  
  writeln('Boshida bir necha bo‘shliq belgisi qatnashgan s-satriy miqdor ni kiriting');  
  readln(s);  
  while (pos(' ',s)=1) and (length(s)>0) do  
    delete(s,1,1);  
  write(Natija',s);  
end.
```

MUSTAHKAMLASH UCHUN SAVOLLAR

1. Paskal dasturlash tilida matnli ma’lumotlar bilan ishlashda qanday tipidan foydalaniladi?
2. Concat funksiyasini vazifasi nimadan iborat? Misollar keltiring.
3. Copy funksiyasi nima uchun qo‘llaniladi?
4. Delete protsedura nima va unga misol keltiring.
5. Length funksiyasi nima ish bajaradi, misol bilan tushuntiring.
6. Pos funksiyasi qanday vazifani bajaradi va qachon qiymati nolga teng bo‘ladi.
7. Val protsedurasini vazifasini tushuntiring.
8. Paskalda berilgan satr ichiga boshqa satrni joylashtirishning imkoni bormi? Javobingizni tushuntiring.
9. Sonli miqdorni satrli miqdorga o‘tkazib bo‘ladimi? Javobingizni izohlang.

FOYDALANILGAN ADABIYOTLAR

1. Фаронов В. В. Turbo Pascal. — СПб.: ВХВ- Санкт-Петербург, 2004. — 1056 с. 21-23 сс.
2. Слинкин Д.А. Основы программирования на Турбо-Паскале: Учебно-методическое пособие для студентов вузов. Шадринск: Изд-во Шадринского пединститута, 2003. - 244 с. 76-80 сс.
3. М.У.Ашуров, Н.Д.Мирзахмедова. Turbo Pascal дастурлаш тили.(услубий кўлланма), Тошкент ТДПУ – 20116- 10. 74-79 betlar.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

MA’RUZA № 9, 10

MAVZU: QISM DASTURLAR: PROTSEDURA VA FUNKSIYALAR BILAN ISHLASH.

Reja:

1. Qism dasturlar haqida umumiy ma’lumotlar.
2. Parametrlı protseduralar.
3. Protsedura operatorlar.
4. Funksiya haqida umumiy ma’lumot.

Tayanch so‘z va iboralar: funksiya, protsedura, qism dastur, Pascal dasturlash tili, dasturlash.

Qism dasturlar haqida umumiy ma’lumotlar

Dasturlash amaliyotida shunday hollar uchraydiki, bunda bir xil hisob kitoblarnı bajarishga to‘g‘ri keladi, lekin har xil boshlang‘ich ma’lumotlar bilan. Bir xil yozuvlarnı chiqarib tashlash va shu bilan dasturnı soddaroq va tushunarliroq qilish uchun, ushbu takrorlanuvchi hisob-kitoblarnı zaruriyati doirasiga qarab ko‘p marotaba ishlatish mumkin bo‘lgan mustaqil dastur sifatida ajratish mumkin. Ma’lum bir algoritmnı amalga oshiruvchi va o‘ziga umumiy dasturnı har xil qisimlardan murojaatga imkon beruvchi dasturning bunday avtonom qismi qism dastur deyiladi.

Qism dasturlar dasturning ma’lum bir ajratilib berilgan sohasi ko‘rinishida rasmiylashtiriladi, bunda qism dasturning kirish va chiqish sohasi yaqqol belgilangan bo‘ladi. Qism dasturining mustaqil harakteri shundaki, u o‘zini har xil foydalanuvchilarga tuzishga imkon beradi. Bunga dasturlash bo‘yicha ishni taqsimlash ro‘y beradi, bu bilan dastur tuzishni yakunlash tezlashadi. Bu ishni osonlashtirish uchun, asosiy dastur va qism dasturdagi o‘zgaruvchi nomlari bir biriga qaram bo‘lmagan holda tuzilgan. Agarda, masalan, asosiy dasturga A nomli o‘zgaruvchi bor bo‘lsa, unda xuddi shunday, faqatgina qism dasturdagi A nomli o‘zgaruvchi, asosiy dasturgagi A o‘zgaruvchining fizik qiymati bilan hech qanday bog‘liq bo‘lmagan umuman boshqa fizik qiymatga ega bo‘lishi mumkin.

Bu bilan qism dasturlarining ijobiy tomonlari tugamaydi masalan, yana xotiraning tejashni kursatish mumkin, qism dasturlarda ishlatiladigan, o‘zgaruvchilarnı saqlash uchun qullaniladigan xotira faqatgina qism dastur ishlaydigan vaqtda ajratiladi. Bu xotira qism dasturining bajarilishi tugagach bo‘shatiladi.

Qism dasturlarnı qo‘llash dasturlashning eng progressiv metodlaridan biri-strukturalı dasturlashni amalga oshirishga imkon beradi.

Paskal tilida qism dasturning ikki ko‘rinishi ajratiladi: protsedura (PROCEDURE) va funksiya (FUNCTION)

Ixtiyoriy dastur o‘ziga bir necha protsedura va funksiyanı qamrab olishi mumkin. Paskal tili dastur strukturasiining umumiy ko‘rinishiga marojaat qilaylik. Protsedura va funksiya bayon qilish qismining o‘zgaruvchilar qismidanoq keyin e‘lon qilinadi.

Ikkita qism dasturga ega bo‘lgan dastur strukturasiiga misol

PROGRAM NOM (INPUT, OUTPUT);

(*Asosiy dasturning bayonı bo‘limi *)

LABEL-Belgilar bo‘limi;

CONST- Konstantalar bo‘limi;

TYPE- Tiplar bo‘limi;

VAR- o‘zgaruvchilar bo‘limi;

I qism dastur P1 I

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

I qism dastur P2 I

(*Asosiy dasturning operatorlari bo`limi *)

BEGIN

```
Operatorlar;  
Qism dasturni chaqirish     P1;  
Operatorlar;  
Qism dasturi chaqirish     P1;  
Operatorlar;  
Qism dasturlari chaqirish   P2;  
Operatorlar;  
Qism dasturni chaqirishi   P1;  
Operatorlar;
```

END.

Dasturning bajarilishi asosiy dasturning operatorlaridan boshlanadi. Zaruriyat tug`ilsa qism dastur chaqiriladi va uning operatorlari ishga kiradi.

Ixtiyoriy qism dastur o`z navbatida qism dasturiga ega bo`lishi mumkin. Bayonotni soddalashtirish uchun o`zida boshqa qism dasturlarni saqlamagan, ularni chaqirish esa asosiy dasturdan amalga oshiriladigan qism dasturlar bilan chegaralanamiz.

Qism dastur xuddi asosiy dastur kabi rasmiylashtiriladi ya`ni sarlavha, bayonlar bo`lishi va operatorlar bo`limidan iborat.

Protsedura-operatorlar

Dastur tuzish jarayonida uning turli joylarida ma`nosiga ko`ra bir xil, mustaqil xarakterga ega bo`lgan va yechilayotgan asosiy masalaning biror qismini hal qilishga mo`ljallangan murakkab algoritmdan bir necha marotaba foydalanishga to`g`ri keladi. Masalan, matritsalarini ko`paytirish, matritsani vektorga ko`paytirish, chiziqli bo`lmagan tenglamani yechish, chiziqli algebraik tenglamalar tizimini yechish, faktorial hisoblash, yig`indi hisoblash va hokazo kabi masalalarni hal qilish algoritmlari ko`plab masalalarni yechishning bosh algoritmlarida qayta-qayta ravishda, turli boshlang`ich ma`lumotlar bilan birgalikda qatnashishi mumkin, Bunday hollarda, malakali dasturchi dastur matnini ixchamlashtirish, dasturning ishonchlilik darajasini oshirish, dasturni tahrir qilish (otladka) ni tezlashtirish va dasturning umumiylikini (universalligini) ta`minlash uchun protsedura hamda funksiyalardan kengroq foydalangan holda mukammal dastur yaratishga harakat qiladi.

Protsedura va funksiyalar mustaqil dasturli ob`ektlar hisoblanadi. Bu mustaqil dasturli ob`ektni dasturchi o`z xohishiga va undan olinadigan natijalarga ko`ra protsedura yoki funksiya ko`rinishida aniqlashi mumkin. Odatda olinadigan natija yagona qiymatli bo`lsa funksiyadan, olinadigan natijalar soni bir nechta bo`lsa protseduradan foydalanish maqsadga muvofiqdir.

Protsedurani yozish strukturasi xuddi asosiy dastur strukturasi kabi bo`lib, faqat sarlavhalari bilangina farq qiladi xolos:

```
procedure <protsedura ismi>(<formal parametrlar ro`yxati>);  
label <metkalar ro`yxati>;  
const <o`zgarma slarni kiritish>;  
type <yangi tiplarni aniqlash>;  
var <o`zgaruvchilarning tiplarini e`lon qilish>;  
<qism dasturgagina tegishli bo`lgan ichki protsedura va iunksiyalar e`loni>;  
begin  
  <protseduraning tana qismi>  
end;
```

Protseduralar va funksiyalarni aniqlash asosiy dasturning var (o`zgaruvchilarning tiplarini e`lon qilish) bo`limida bajariladi. Protseduradan dasturda foydalanish uchun uning ismi va faktik parametrlar ro`yxati yoziladi. Shunda protsedura o`ziga belgilangan ishni bajarib, o`zining faktik parametrlari orqali asosiy dasturga o`z natijasini beradi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Protseduraning e'loni va unga murojaat qilishni kelgusida ko'riladigan misollar orqali o'zlashtirib olamiz.

Parametrsiz protseduralar

Yuqorida aytib o'tilganidek, protsedura hisoblab bergan natijalar uning faktik parametrlari orqali asosiy dasturga uzatiladi. Lekin, ayrim paytlarda protsedura parametrsiz ham bo'lishi mumkin. Bu holda asosiy dasturning barcha parametrlari protsedura parametrlari rolini bajaradi. Parametrsiz protsedurada ham protseduraning barcha bo'limlari saqlanib qoladi, faqat parametrlar ro'yxatigina qatnashmaydi.

Protseduralarni aniqlash va ulardan foydalanishni quyidagi misolda ko'rib chiqaylik:

M i s o 1: $u = \max(x + y, x * y)$, $v = \max(0.5, u)$ — berilgan x va y haqiqiy sonlardan foydalanib u va v qiymatlarni aniqlash.

Bu yerda x, y — qiymatlari kiritiladigan haqiqiy tipli o'zgaruvchilar.

1. Masalani yechish dasturining protseduradan foydalanmay tuzilgan shakli:

```
Program max;
var
x, y, u, v: real;
a, b, s: real;
begin
{x, y — miqdorlarni kiritish};
readln(x,y);
a:=x+y\ b:=x*y;
if a > b then S:=a else S:=b; u:=S;
a:=0.5; b:=u;
if a > b then S:=a else S:=b;
v:=S;
{olingan natijalar};
writeln(u, v) end.
```

Ahamiyat bersangiz, dasturdagi shartli operator ikki marta takrorlanib, bir xil ish bajardi.

2. Masalani yechish dasturining parametrsiz protseduradan foydalanib tuzilgan shakli (endi yuqoridagi dasturda yo'l qo'yilgan kamchilikni protseduralar orqali tuzatishga harakat qilamiz):

```
Program max;
var x, y, u, v: real; a, b, S: real;
procedure max1;
begin
if a>b then S:=a else S:=b;
end;
begin
readln(x, y);
a:=x+y; b:=x*y;
max1; {max1 protsedurasiga 1-marta murojaat qilinmoqda}
u:=S
a:=0.5; b:=u;
max1; {max1 protsedurasiga 2-marta murojaat qilinmoqda}
v:=S;
writeln(u,v);
end.
```

Asosiy dasturning operatorlar qismida ikki marta yozilgan max1 parametrsiz protsedurasiga murojaat e'lon qilingan protsedurani ikki marta asosiy dasturga olib kelib

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

ishlatishni ta'minlaydi. Ahamiyat berilsa, ikkinchi dastur birinchi protsedurasiz tuzilgan dasturga ko'ra ixchamroq va soddaroqdir. Biz kiritgan protsedura hozircha faqat ikkita haqiqiy son ichidan kattasini aniqlab berdi xolos. Shuning uchun dasturning hajmini kamaytirishdan erishgan yutuq salmoqli bo'lmadi. Lekin, protseduralar asosan ko'p hajmli matndagi amallarni, vazifalarni bajarishga mo'ljallanadi va bu holda erishilgan yutuq salmog'i ancha yuqori bo'ladi.

Parametrsiz protseduraning asosiy kamchiligi — uning asosiy dasturga va undagi ma'lum parametrlarga bog'lanib qolganligidir

Parametrli protseduralar

Protsedura bilan asosiy dasturni bog'laydigan bosh omil — protsedura parametrlaridir. Parametrlarni ikkita tipga ajratiladi: qiymatli parametrlar (parametr-qiymat), o'zgaruvchili parametrlar (parametr - o'zgaruvchi).

Parametr-qiymatlar protseduraning ishlash jarayonini ta'minlovchi parametrlar hisoblanadi, ya'ni ular asosiy dastur qiymatlarini protseduraga uzatadigan parametrlardir.

Endi yuqorida ko'rib chiqilgan sonlarning eng kattasini topish algoritmining dasturini qiymatli parametr bilan yozilgan protseduralar orqali amalga oshiraylik:

```
Program max;  
var x, y, u, v: real; S: real;  
procedure max2 ( a, b: real );  
begin  
if a>b then S:=a else S:=b;  
end;  
begin  
read (x, y);  
max2 (x + y, x * y); u:=S; max2 (0.5 , u); v:=S; writeln (u, v)  
end.
```

bu yerda a, b — protseduraning qiymatli formal parametrlari.

Protseduraga murojaat qilishda formal va faktik parametrlarning tiplari o'zaro mos kelishi kerak, aks holda dastur xato tuzilgan hisoblanadi. Yuqoridagi dasturdan ko'rinib turibdiki, a va b formal parametrlar o'rniga natijaviy qiymatlari ma'lum bo'lgan ifodalar qo'yildi. Demak, qiymatli faktik parametrlar o'rniga shu tipli natijaga erishuvchi ifodani yozish mumkin. Bundan tashqari, protsedurada kiritilgan a va b parametrlar faqat protseduraning ichidagina ma'noga ega, tashqarida, misol uchun asosiy dasturda ular tushunarsiz, qiymatlari aniqlanmagan miqdorlardir. Shuning uchun qiymatli parametrlarga protsedura natijalarini o'zlashtirib, asosiy dasturga uzatib bo'lmaydi.

Yuqorida tuzilgan dasturning asosiy kamchiligi shundaki, topilgan katta son doim S o'zgaruvchiga o'zlashtirilayapti. Misolimiz shartiga ko'ra esa natijalar u va v o'zgaruvchilarga o'zlashtirilishi kerak. Shuning uchun, dasturda ikki marta qo'shimcha $u:=S$ va $v:=S$ o'zlashtirish operatorlari yozildi.

Bu kamchilikni tuzatish uchun protseduraga yana bir parametrni kiritamiz. Lekin, kiritilgan bu parametr protseduraga qiymat olib kirmaydi balki, protsedura natijasini asosiy dasturga olib chiqib ketadi. Bunday parametrni *parametr-o'zgaruvchi* deb ataladi.

Parametr-o'zgaruvchini parametr-qiymatdan farq qilish uchun protsedurani aniqlashdagi parametrlar ro'yxatida o'zgaruvchi oldidan oarxizmatchi so'zi yoziladi. Parametr-o'zgaruvchidan so'ng albatta uning tipi ko'rsatib qo'yiladi.

di. Yuqorida aytganimizdek, formal parametr-qiymat o'rniga protseduraga murojaat qilganda shu tipli ifodani yozish mumkin bo'lsa, parametr-o'zgaruvchi uchun bu hol mutlaqo mumkin ernas.

Protsedurani mukammallashtirib borish dinamikasini his etish uchun yana, yuqorida ko'rilgan maksimumni topish misolini parametr-o'zgaruvchi ishlatgan holda ko'rib chiqamiz:

```
Program max;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
var
x, y, u, v: real;
procedure max3(a, b: real; var S: real);
begin
if a>b then S:=a else S:=b;
end;
begin
readln (x, r/);
max3( x + y, x * y, u); {x+y va x*y ifodalarining kattasi u o‘zgaruvchiga
o‘zlashtirilmoqda}
max3( 0.5, u, v); {0.5 va u ifodalarining kattasi v o‘zgaruvchiga o‘zlashtirilmoqda} writeln(
u, v)
end.
```

Shunday qilib, bitta dasturni protseduraning uch xil varianti uchun tuzib chiqib, natijada ixcham va sodda dasturga ega bo‘ldik.

Protseduralarni aniqlashda shu paytgacha oddiy tipli parametrlardan foydalanib keldik. Lekin, biz shuni yaxshi bilamizki, Paskal tilida hosilaviy tiplar ham mavjud. Parametr-o‘zgaruvchiga hosilaviy yangi tiplar berish xuddi oddiy skalyar tip berish kabi amalga oshirilaveradi. Ammo parametr-qiymatlarda yangi tiplar masalasiga batafsilroq yondashish kerak.

Biz yuqorida eslatib o‘tdikki, faktik parametr formal parametr-qiymatga mos tipli ixtiyoriy ifoda bo‘lishi mumkin. Lekin, Paskal tilida ixtiyoriy tipli qiymatlar uchun shu tipdagi natija beruvchi hech qanday amal ko‘zda tutilmagan. Shuning uchun, bu tiplar uchun faktik parametrlar faqat shu tipga mos o‘zgaruvchilar bo‘lishi mumkin xolos. Bunday hol, xususiy holda massivlar uchun ham o‘rinlidir.

Faraz qilaylik, dasturda o‘zgaruvchilar quyidagicha e‘lon qilingan bo‘lsin:

```
const rc=20;
```

```
type vector = array [1..N] of real;
```

```
var u, v. real; x, y: vector;
```

Bu yerda $u = \max(x)$, $v = \max \{/\./\}$ larni aniqlash talab qilinayotgan bo‘lsin.

Vektorning eng katta hadini topishni albatta protsedura ko‘rinishida tashkil qilamiz:

```
procedure max1 (A:vector; var S: real);
```

```
var i: integer;
```

```
begin
```

```
S:=A[1];
```

```
for i:=2 to n do if A[i] > S then S:= A[i]
```

```
end.
```

Asosiy dasturda bu protseduraga murojaat

```
max1 (x, u); max1 (y, v); ko‘rinishda amalga oshiriladi.
```

Protseduradagi A vektorni parametr-qiymat sifatida yozib qo‘yganimiz uchun, protseduraga qilinayotgan har bir murojaatda A vektorga mos ravishda X va Y vektorlar ko‘chirib yoziladi va so‘ng protsedura o‘z ishini bajaradi. Biz bilamizki, bir tarafdin, massivlarning ustida ko‘chi-rish amalini bajarishga ancha vaqt ketadi, ikkinchi tarafdin, har safar yangidan protseduraga qilingan murojaatda A vektor uchun xotiradan qo‘shimcha joy ajratiladi. Shuning uchun, protseduraning sarlavhasida quyidagicha almashtirish qilsak, yuqoridagi ikki kamchilikni bartaraf qilgan bo‘lamiz:

```
procedure max1 (var A: vector; var S: real);
```

Endi protsedurani e‘lon qilish, undan foydalanib dastur yaratish malakasini hosil qilganimizdan so‘ng uni e‘lon qilishning sintaksis qoidalarini ko‘rib chiqaylik.

Protsedurani aniqlash (e‘lon qilish) quyidagicha amalga oshiriladi:

```
<protsedurani aniqlash> ::= <protsedura sarlavhasi>; <blok> -
```

Bu yerda. <blok> tushunchasi to‘liqligicha <dastur tanasi> tushunchasi bilan bir xil sintaksis qoida asosida aniqlangani tufayli bu tushunchaga ortiq qaytib o‘tirmaymiz.

Endi esa <protsedura sarlavhasi>ga ta‘rif beramiz:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

<protsedura sarlavhasi> ::= Procedure <protsedura ismi> |

Procedure <protsedura ismi> (<formal parametrlar ro‘yxati>)

Protsedura ismi dasturchi tomonidan tanlanadigan oddiy identifikator hisoblanadi.

Formal parametrlar ro‘yxati quyidagicha aniqlanadi:

< formal parametrlar ro‘yxati > ::= < formal parametr
lar seksiyasi > { ; . < formal parametrlar seksiyasi > }

Formal parametrlar seksiyasi deganda protsedura parametrlarining parametr-qiymat va parametr-
o‘zgaruvchilardan iborat bo‘lishi tushuniladi:

< formal parametrlar seksiyasi > ::= { < ism > { , < ism > } :

< ism tipi > var < ism > { , < ism > } : < ism tipi >

bu yerda < ism > — formal parametrlar sifatida ishlatiladigan identifikator.

Endi yuqoridagi aniqlashlarga tushuntirishlar berib o‘tamiz.

Yuqoridagi **Bekus** — **Naur** formulalaridan ko‘rinib turibdiki, formal parametrlar ro‘yxati (agar u mavjud bo‘lsa) bitta yoki bir nechta o‘zaro nuqta- vergul (;) belgisi bilan ajratilgan seksiyalardan tashkil topgan. Har bir seksiyada esa o‘z navbatida bitta yoki bir nechta o‘zaro nuqta- vergul bilan ajratilgan formal parametrlar qatna-shishi mumkin. Protseduradagi formal parametrlar sonini dasturchining o‘zi protsedurani aniqlash mohiyatidan kelib chiqqan holda tanlaydi.

Misol 1:

Procedure P(A:Char; 5:Char; Var C:Real; Var £>:Real; £:Char);

bu yerda formal parametrlar ro‘yxati beshta seksiyadan iborat: *A, B, E* lar Char tipli qiymatlar, *C, D* lar Real tipidagi o‘zgaruvchilar. Shu bilan bir qatorda har bir seksiya faqat bitta parametrni o‘z ichiga olmoqda.

Bir xil tipli, hamda ketma-ket joylashgan qiymatlarni va o‘zgaruvchilarni bitta seksiyaga birlashtirib protsedura sarlavhasini quyidagicha yozish ham mumkin:

Procedure P(A,B:Char; Var C,D:Real; E:Char);

Shunday qilib, *seksiya* deganda bir xil tipli parametr-qiymatlar yoki parametr-
o‘zgaruvchilarning ro‘yxatini tushunish mumkin.

Ko‘pchilik boshlovchi dasturchilar yo‘l qo‘yadigan quyidagi xatoliklardan ehtiyot bo‘lmoq zarur:

Procedure P(Var J:Real; 7:Real); bu sarlavha

Procedure P(Var *,y:Real); sarlavha bilan bir xil efnas.

Aniqlangan protseduraga murojaatni yoki protsedura operatoridan qanday foydalanishni aniqlashni ko‘rib chiqaylik (yaratilgan protsedurani «Aktivlashtirish», ya’ni ishlatish):

<protsedura operatori> ::= <protsedura ismi> ^protsedura ismi (<faktik parametrlar ro‘yxati>)

Agar protsedura aniqlanishida parametrsiz bo‘lsa, unga murojaat qilish ham faqat protsedura ismini yozish bilangina amalga oshiriladi.

Agar protsedura aniqlanishida parametrli bo‘lsa, albatta protsedura-operator ham unga mos faktik parametrlar ro‘yxatiga ega bo‘ladi. Shu parametrlar orqali protseduraga murojaat qilinayotganda formal parametrlar faollashtiriladi:

< faktik parametrlar ro‘yxati > ::= < faktik parametr > { , < faktik parametr > }

Protseduralar haqida umumiy ma’lumot

Protseduralar. Ixtiyoriy protsedura o‘zida bir nechta operatorlar va ularning bajarilishidan kelib chiqan bir nechta natijasini o‘zida saqlashi mumkin.

PROCEDURE NOM (formal parametrlar);

I Bayonlar bo‘limi I

BEGIN

I Operatorlar bo‘limi I

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

END;

Sarlavhada PROCEDURE hizmatchi so‘z, undan so‘ng aylana qavsga olingan protsedura nomi va formal parametrlar ko‘rsatilgan.

Protsedura bayoni bo‘limi dasturning bayoni bo‘limiga o‘xshash bo‘lib, belgilar, konstantalar, tiplar, o‘zgaruvchilar bo‘lishidan iborat va o‘z navbatida protsedura va funksiyalar.

Operatorlar bo‘limi BEGIN va END va operatorlar qavsga olinadi, buning ustiga asosiy dasturdan farqli ravishda END dan so‘ng nuqta vergul qo‘yiladi.

Protsedura o‘zining nomi bilan chaqiriladi.

Nom (faktli parametrlar)

Formal va faktli parametrlar yordamida ma‘lumotlar dasturdan protsedura va aksincha proseduradan dasturga uzatiladi. Har bir formal parametr o‘zining tipi bilan birgalikda ko‘rsatiladi. Unga mos keluvchi faktli parametr tipsiz ko‘rsatiladi. Formal va faktli parametrlar o‘rtasida parametrlar soni ularning tipi va ketma-ketlik tartibi bo‘yicha muvofiqlik bo‘lishi kerak.

Agar bir nechta formal parametrlar bir xil tipga ega bo‘lsa, unda ularni bir xil ro‘yhatga birlashtirish mumkin, ya‘ni parametrlarni vergul orqali sanab o‘tib, undan so‘ng ularning tipi ko‘rsatiladi.

Proseduraning sarlavhasi mavjud bo‘lsin:

```
PROCEDURE SUMMA (A, B, C: INTEGER);
```

bu yerda A, B, C - formal parametrlar.

Protsedurani quyidagicha chaqirish mumkin

```
SUMMA (5, M, 7);
```

Bu yerda 5, M, 7-faktli parametrlardir. Har bir formal parametr faktli parametrlarga mos keluvchi qiymatni qabul qiladi. Shunday qilib protsedura quyidagi qiymatlariga ega bo‘lamiz: A=5, B=M, C=7.

Protsedura parametrlari o‘rtasida qiymat-parametrlari va o‘zgaruvchi parametrlari ajratiladi. Qiymat parametrlari protseduraning kirish parametrlari rolini bajaradi. Ular faktli parametrlar qiymatlarini qabul qila oladi, lekin o‘zlarining qiymatlarini faktli parametrlarga uzata olmaydi. O‘zgaruvchi-parametrlar protseduraning kiruvchi va chiquvchi parametrlari rolini bajaradi. Ular faktli parametrlar qiymatlarining qabul qilishi, ularni protsedurada o‘zgartirishi va yangi qiymatlarini faktli parametrlarga qaytarishi imkoniyatiga ega. Formal parametrdagi parametr- o‘zgaruvchilarni ajratish uchun ular oldidan VAR so‘zi qo‘yiladi. Protseduraga misol:

```
PROCEDURE EXPRES (A, B, C: REAL; VAR X, Y: REAL);
```

```
VAR Z: REAL;
```

```
BEGIN
```

```
Z:=A+B+C;
```

```
X:=SQR (Z); (*Z ning kvadrati *)
```

```
Y:= SQRT (Z); (*Z dan ildiz chiqarish *)
```

```
END;
```

Ushbu protsedurani quyidagicha chaqirish mumkin:

```
EXPRES (25.4, 44.6, 30, X1, Y1)
```

A, B, C formal kirish parametrlari A=25.4, B=44.6, C=30 faktli parametrlariga muvofiq qiymatlarini qabul qiladi. Ushbu qiymatlarda protsedura bajariladi. Protseduraning bajarilishidagi natija x, y bo‘lib, o‘z qiymatlarini muvofiq ravishda x1, y1 afledish parametrlarga uzatadi. Shunday qilib, asosiy dasturda X1=10000, Y1=10 ega bo‘lamiz.

O‘zaro muvofiq formal va faktli parametrlarni bir xil belgilashga yo‘l qo‘yiladi. Masalan, protsedurani chaqirish quyidagi ko‘rinishga ega bo‘lishi mumkin:

```
EXPRES (25.4, 44.6, 30, x, y)
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Faktli parametrlar sifatida konstanta, o‘zgaruvchilar, ifodalar bo‘lishi mumkin. Ularning tipi formal parametr tipiga muvofiq bo‘lishi shart. Masalan, protsedurani chaqirish quyidagicha bo‘lishi mumkin.

```
A: =25.4; C: =30;  
EXPRES (A, 44.6, C, X1, Y1)
```

yoki

```
D: =44.6; R:=10;  
EXPRES (25.4, D, R+20, Z, T)
```

Ixtiyoriy vaziyatda ham formal parametrlar $A=25.4$, $B=44.6$, $C=30$ qiymatlarni qabul qiladi. Faktli parametrlarning o‘zgaruvchi –parametrlari sifatida o‘zgaruvchilar va massivlar bo‘lishi mumkin; formal parametrlar faqat nomlar bilan ifodalangan. (konstanta va ifodalar ruxsat etilmaydi)

Agarda protseduraga parametr sifatida faqatgina bitta qiymatni emas, balki massivni uzatish kerak bo‘lsa, unda faktli parametr sifatida massivning nomi bo‘lishi kerak. Bunda formal parametr VAR so‘zidan so‘ng massiv tipi bilan ko‘rsatiladi. Massivning o‘zining bayoni esa asosiy dasturning TYPE bo‘lishiga qilinadi.

Masalan asosiy dasturda quydagi A massivning bayoni mavjud bo‘lsin:

```
CONST N=10;  
TYPE MAS=ARRAY [1 ..N ] OF REAL;  
VAR A: MAS;
```

Unga PRIM protsedurasining formal parametrlari quyidagi ko‘rinishiga ega bo‘lishi mumkin:
PROCEDURE PRIM (K: INTEGER; VAR X: MAS);

Quyidagi PRIM (N, A); faktli parametrlarda K formal parametr N faktli parametr qiymatini, X formal parametr esa A massivning qiymatini qabul qiladi.

Protsedurada X massivning qiymatlari o‘zgarishi mumkin, va massiv A yangi qiymatlarni qabul qiladi.

- Parametrsiz protseduralarni qo‘llashga ruxsat etiladi. Bu holda formal va faktli parametrlar yo‘q bo‘ladi. Protsedurada asosiy dasturdagidek kabi bayon bo‘limini yo‘q bo‘lishi mumkin.

Misol 1 Butun sonlarning $1\ 2\ 3\ \dots\ N$ ko‘paytmasi va $1+2+3+\dots +N$ yig‘indining hisobini “SUMMA” protsedurasi ko‘rinishida rasmiylashtiraylik.

Yig‘indi va ko‘paytmani arifmetik va geometrik progressiya formulalarini qo‘llab hisoblashimiz mumkin. Ammo biz har xil parametrlarining vazifasini namoyishi qilish uchun ataylab sikldan foydalanamiz.

1 Dastur

```
(* -----  
I Prasedurani rasmiylashtirishi I  
----- *)  
PROGRAM PT9 ( INPUT, OUTPUT);  
VAR N: INTEGIR;  
SUM, PR: REAL;  
(* "SUMMA" protsedurasi ----- *)  
PROCEDURE SUMMA (K: INTEGIR; VAR X , Y :REAL );  
VAR I: INTEGER;  
BEGIN  
X: =0, Y: =1;  
FOR I: =1 TO K DO  
BEGIN  
X: =X+I;  
Y: =Y*I;  
END  
END;  
END;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

(*asosiy dastur *)

```
BEGIN
  WRITE ('N ning qiymatini kiriting:');
  READ (N);
  SUMMA (N, SUM, PR);
  WRITELN ('SUM=', SUM:5);
  WRITELN ('PR =', PR:5);
```

END.

```
=====
= N ning qiymatini kiriting :10
= sum =5.500000 E+01
= PR =3.628800E +06
=====
```

1 Dastur asosiy dasturning operatorlaridan boshlab xisoblanishi boshlanadi. N ning qiymati kiritilgandan so`ng “SUMMA “ protsedurasi chaqiriladi. Bunda K formal parametr N faktli parametrning qiymatini qabul qildi. X summa va Y ko`paytmalar qiymatlarinig protseduradagi xisobi asosiy dasturning SUM va PR faktli parametrlariga uzatiladi. Asosiy dastur bajarilishi davom etadi:SUM va PR qiymatlari displey ekraniga chiqariladi.

Asosiy dasturning bayoni bo`limida N, SUM PR o`zgaruvchilar tanishtirilgan; protseduraning bayoni bo`limida esa faqatgina protsedurada qo`llaniladigan I o`zgaruvchi tanishtirilgan.

Funksiyalar

Funksiyalar. Qism dasturning ikkinchi ko`rinishi - funksia - protseduraga o`xshash rasmiylashtiriladi. Funiksiyaning farqli xususiyati: u faqat bitta bajarish natijasiga ega (lekin bir nechta kiruvchi parametrlarga ega bo`lishi mumkin) natija funiksiya nomi bilan belgilanadi va asosiy dasturga uzatiladi.

Funiksiya quydagi ko`rinishda rasmiylashtiriladi:

FUNCTION NOM (formal parametrlar):

```
-----
I   Bayonlar           bo`limi       I
-----
```

BEGIN

1. -----

```
I           Operatorlar bo`lishi       I
-----
```

END;

Faktli parametrlarning ko`rsatilishi bilan birgalikda funksiya o`zining nomi bilan chaqiriladi.

Funiksiyani bevosita ifodaning ichiga chaqirish mumkin. Funiksiyani chaqirishda uning tipi ko`rsatilmaydi.

2 Misol. $F=M!-K!$ faktoriallarining farqi (ayirmasi)ni topish talab etilsin. $n!$ bu natural ketma-ketlikning n ta soning ko`paytmasini ifodalanishini eslatib o`tamiz:

1.2.3...n.

Faktorlar hisobini funksiya ko`rinishida rasmiylashtiriladi:

```
FUNCTION FACT (N: INTEGER): INTEGER ;
```

```
VAR P, I : INTEGER;
```

```
BEGIN
```

```
P: =1;
```

```
FOR I: =2 TO N DO
```

```
  P: =P*I;
```

```
FACT: =P;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

END;

Bu yerda FACT-funksiya nomi. Faktorial hisobining natijasi uning nomi bilan ifodalanadi. Funksiyaning tipi haqiqiy. Butin N tipli o‘zgaruvchi formal parametrdir.

M va K faktli parametrlari ko‘rsatilishi bilan birgalikda funksiya o‘zining FACT nomi bilan chaqiriladi:

$$F = \text{FACT}(M) - \text{FACT}(K)$$

Nomlarining ta‘hiri maydoni. Asosiy dasturning bayoni bo‘limida e‘lon qilingan nomlar asosiy dasturning bayoni bo‘limiga va ixtiyoriy qisim dasturga (protsedura va funksiya) da o‘z kuchini yo‘qatmaydi. Ushbu nomlar global nomlar deyiladi. Qism dasturga e‘lon qilingan nomlar faqatgina shu qism dasturda va unda e‘lon qilingan ixtiyoriy funksiya va protsedura doirasida o‘z ta‘sirini yo‘qotmaydi. Bunday nomlar lokal nomlar deyiladi. Ular asosiy dasturning operatorlari uchun ochiq emas.

2 Dastur

```
(* ----- *)
I   Funksiyaning rasmiylashtirish           I
-----*)
PROGRAM PT11 (INPUT, OUTPUT) ;
      VAR F, M, K: INTEGER;
(* Funksiya "FACT" -----*)
FUNCTION FACT (N: INTEGER): INTEGER;
VAR P, I: INTEGER;
BEGIN
P:=1;
FOR I:=2 TO N DO
      P:=P*I;
FACT:=P;
END;
      (*Asosiy dastur ----- *)
BEGIN
      WRITE ('M, K ning qiymatini kiriting:');
READ (M, K);
F:=FACT(M)-FACT(K);
WRITELN ('F=', F: 5);
END.
=====
M, K ning qiymatini kiriting: 7 4
= K=5016
=====
```

Shunday qilib ko‘rib chiqilgan FACT funkisiyali 12.2 dasturda, asosiy dasturning bayon bo‘limidagi F, M, K o‘zgaruvchilar global hisoblanadi. Funksiyaning bayoni bo‘limidagi P, I o‘zgaruvchilar va N parametr lokaldir: ularning ta‘sir doirasi funksiyaning ichidadir.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

MA’RUZA №11

MAVZU: PASKALDA FAYLLAR BILAN ISHLASH.

Reja:

1. Faylli toifaning ifodalanishi.
2. Toifalashgan va toifalashmagan fayllar.
3. Matnli fayllar.

Tayanch so‘z va iboralar: Fayllar, Pascal ABC, standart modullar, rost, yolg‘on, toifa, qayta ishlash.

Faylli toifaning ifodalanishi

Fayl, fayl o‘zgaruvchisi

Fayl - bu tashqi tashuvchida xotiraning nomlangan qismi bo‘lib, berilganlarni saqlash uchun xizmat qiladi. Ko‘p hollarda fayllar matnli qatorlar va yozuvlardan iborat bo‘lib, qattiq yoki egiluvchan disklarda joylashadi. Faylni e‘lon qilish uchun File of iborasi ishlatiladi. Faylga kirish uchun maxsus faylli o‘zgaruvchini qo‘llash kerak (uni F bilan belgilaymiz). Agar fayl yozuvlardan iborat bo‘lsa, u holda yozuv maydonlariga kirish uchun qo‘shimcha o‘zgaruvchi kiritish kerak (uni R bilan belgilaymiz).

Yozilishi:

TRUE

<toifa nomi> = <komponentalar toifasi >;

VAR

<F> : File of <toifa nomi>;

<R> : <toifa nomi>;

Turbo PascalABC kompilyatori 3 toifadagi fayllarni qo‘llash imkonini beradi: matnli, toifalashgan, toifalashmagan. Bularni o‘rganishdan avval Turbo PascalABCning kiritish va chiqarish standart vositalari bilan tanishib chiqish zarur.

Fayllarni qayta ishlash vositalari

Matnli fayllar, toifalashgan fayllar, toifalashmagan fayllar, Assign, tashqi faylni ochish, kiritish-chiqarish operatsiyalarida xatolik holatlar, faylni yopish, bazi bir fayllar nomini o‘zgartirish va o‘chirish.

PascalABCda uchta sinf fayllari mavjud: Matnli, toifalashgan va toifalashmagan. PascalABCning fayllar sistemasida DOs operatsion sistemasining imkoniyatlaridan to‘la foydalanish mumkin. Tilda har bir faylga aniq toifadagi faylli o‘zgaruvchi qo‘yiladi, shuning uchun fayllar bilan qo‘llashdan oldin xxxxxxxx ni o‘rnatish lozim. Buning uchun tilda quyidagi protseduradan foydalaniladi:

Assign (VAR F; Name: string);

bu erda F - fayl toifali o‘zgaruvchi, qatorli ifoda Name esa operatsion sistemaning talablariga javob beruvchi faylni to‘la nomini o‘z ichiga oladi. Fayl nomining umumiy ko‘rinishi quyidagicha:

Disk:\KatalogOstiNomi\...\KatalogOstinomi\FaylNomi

<Disk> Identifikatori mantiqiy qurilmaning A dan Z gacha bo‘lgan literi orqali beriladi. Agar \KatalogOstiNomi\ e‘lon qilinmasa, fayl joriy ost katalogda deb qabul qilinadi. Faylning to‘la nom uzunligi 79 belgidan oshmasligi kerak. Fayllar bilan ishlash uchun eng avvalo uni ochish zarur. Buning uchun PascalABC tilida ikkita

protsedura mavjud:

Reset(VAR F : File);

- bor faylni ochish;

Rewrite(VAR F : File); - yangi fayl tashkil qiladi va ochadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Ikkala prosedurani yozishda ham File parametri har qanday toifadagi fayl o‘zgaruvchisini bildiradi. Reset prosedurasida yordamida tashqi faylni ochish, agarda diskda u bo‘lmasa, dasturni bajarishda xatolikga olib keladi. Kiritish-chiqarish operatsiyalaridagi bunday xatoliklarga olib kelish holatlarni maxsus ioRezult fuksiyasi kuzatib boradi. Misol. Faylni standart ochish.

```
Assign(F, o‘);
```

```
Reset(F);
```

Bunday fayllarni ochish, berilganlarni klaviaturadan olib kirish imkoniyatini beradi. Turli toifadagi fayllarni ochishda Reset prosedurasidan foydalanilganda qandaydir farq bo‘lishi mumkin. Tekstli fayllarga (Tekst toifali) nisbatan prosedura faylini faqat o‘qish uchun ochadi. Toifalashmagan fayllar uchun ochadi. Toifalashmagan fayllar uchun prosedura yana bitta word toifadagi RecSize parametrini qo‘shadi. Bu parametr fayl almashinuvi funktsiyasida yozuv uzunligini o‘rnatadi. Reset prosedurasida toifalashmagan fayl uchun quyidagi ko‘riishga ega:

```
Reset(VAR F: File; RecSize: word);
```

Rewrite prosedurasida yangi fayl tashkil qiladi va ochadi. Bu proseduradan foydalanish alohida etiborni talab qiladi. Mavjud fayl nomi bilan yangi faylni tashkil qilish yoki ochishda Rewrite prosedurasida shu nomli eski faylni o‘chirishga olib keladi. Yangi toifalashmagan fayllarni ochishda Rewrite prosedurasida yozuv uzunligini keltirish uchun word toifasidagi qo‘shimcha RecSize prosedurasida qo‘shiladi. U holatda prosedura quyidagi ko‘rinishga ega bo‘ladi:

```
Rewrite(VAR F: File; RecSize: word);
```

Agar Rewrite prosedurasida tekstli fayllarda qo‘llanilsa, u holda, keyinchalik yangi kiritilayotgan berilganlar uchun faqat yozish operatsiyasini qo‘llash mumkin. Faylni yopish operatsiyasi har qanday ochilgan fayllar ustida bajariladigan operatsiyalarni tugatilishini ko‘rsatadi. Bunga quyidagi prosedura xizmat qiladi.

```
Closse(VAR F);
```

Closse prosedurasida fayl o‘zgaruvchisini tashqi fayl bilan aloqasini Assign prosedurasida yordamida yo‘qotadi. Misol. WORK.TXT nomli oddiy matnli fayl tashkil qilish:

```
VAR
```

```
F: text;
```

```
BEGIN
```

```
Assign(F, 'WORK.TXT');
```

```
Rewrite(F);
```

```
Write(F, 'Oddiy matnli fayl');
```

```
Closse(F);
```

```
END.
```

Fayllarga xizmat qilish vositalariga qayta nomlash va bazi bir fayllarni yo‘qotish proseduralarini kiritish mumkin. Bu proseduralardan foydalanish fayl toifasiga bog‘liq emas. ReName(VAR F; NewName : string); Bu prosedura har qanday toifadagi ochilmagan F faylini qayta nomlaydi. Yangi nom NewName qatorida beriladi. Erase(VAR F); Bu prosedura F o‘zgaruvchi orqali berilgan har qanday toifadagi ochilmagan faylni yo‘qotadi. Ikkala prosedurani ham ochilib bo‘lgan fayllar uchun ishlatib bo‘lmaydi. Aks holda operatsion sistema tomonidan keraksiz osoratlar paydo bo‘lishi mumkin. Y‘qotish va qayta nomlash operatsiyalarini faqatgina bor fayllarga qo‘llash mumkin, aks holda dastur bajarilishida xatolikga yo‘l qo‘yiladi.

Misol. Faylni yo‘qotish va qayta nomlash.

```
VAR
```

```
F: File;
```

```
Ch: char;
```

```
St: string;
```

```
BEGIN
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Write('Fayl nomini kiriting: ');
Readln(St);      { nomni o'qish }
Assign(F, St);   { Fayl o'zgaruvchisiga nom tayinlash }
Write ('Faylni yo`qotish (U), Qayta nomlash(P), Chiqish(V)'); Readln(Ch);
CASE Ch OF
'U', 'u' : Erase(F); { faylni yo`qotish }
'P', 'n' : BEGIN
Write('Faylni yangi nomini kiriting: '); Readln(St);
Rename(F, St); { faylni qayta nomlash }
END;
'B', 'v' : HALT(1);
END; { case }
END.
```

Keltirilgan misolda u yoki bu harakatlarni tanlash, klaviaturadan nima kiritilayotganiga bog‘liq. Agar diskda St nomli fayl bo‘lmasa, dasturning bu varianti xatolik holatlarga yo‘l qo‘ymaydi. Fayl operatsiyalari aniq va xatosiz ishlashi uchun maxsus ioRezult funksiyasidan foydalanish zarur. Funktsiya parametrlarsiz ishlaydi va Integer toifasidagi parametrlarni qaytaradi. Dasturda bu funksiyalardan fayl operatsiyalarini bajarish vaqtida, kiritish -chiqarish operatsiyasining standart tekshiruvini o‘chirib qo‘yilganda foydalanish mumkin. Bu maqsadlar uchun maxsus operatsiyadan yoki dastur matni ichida be-riladigan{\$I} kompilyatoridan foydalaniladi.

Misol. Diskda faylning borligini tekshirish uchun variant.

```
VAR
F: File;
St: string;
BEGIN
Write('Fayl nomini kiriting:');
Readln(St);
Assign(F, St);
{$I-}      { xatolikni standart tekshirish o'chirish }
Reset(F);  { fayl ochish }
{$I+}      { xatolikni standart tekshirishni yoqish }
IF Ioresult = 0 THEN BEGIN
Writeln('Fayl bor va ochiq');
Close(F);  { faylni `yopish }
END
ELSE
Writeln(o'+St+' nomli fayl disketada yo`q);
END.
```

Toifalashgan fayllar

Joriy pozitsiya, to`gridan- to`g`ri kiritish fayli, indeksli fayllar.

Toifalashmagan fayllarga aniq toifadagi fayllar kiradi. Ko‘pchilik hollarda bu fayllar yozuvlardan iborat bo‘lib, turli berilganlar bazasini tashkil etishda qo‘llaniladi. Bunday fayli o‘zgaruvchilar dasturda quyidagicha e‘lon qilinadi:

```
TYPE
FileRec = RECORD
...
END;
VAR
F : File of FileRec;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Agar matnli fayllarda belgilanganlar simvollardan tashkil topgan bo‘lsa, toifalashmagan fayllar yozuvlardan tashkil topgan deb qaraladi. Bunday berilganlarning o‘lchov birligi sifatida yozuvning o‘zi qabul qilinadi. Yozuv uzunligi SizeOf(FileRec) bilan aniqlanadi. Bu yozuvlarni katta kapilyatorlar bilan parallel qaralgANDa aytish mum-kinki, so‘z to‘gridan – to‘g‘ri kiritish fayli haqida borayapdi.

Seek (VAR F; NumRec: Longint)

Protsedura F joriy faylning NumRec tartibli o‘zgaruvchisini o‘rnatadi; F - toifalashgan va toifalashmagan berilganlarning fayl o‘zgaruvchisi. Toifalashmagan faylni ochishda joriy pozitsiya yozuv boshiga o‘rnatiladi va 0 chi tartib nomerga ega bo‘ladi, fizik yozuv mantiqiy yozuv tartib nomeridan bittaga kam bo‘ladi.

Misol. Telefonlar haqidagi ma‘lumot misolida toifalashmagan fayllar bilan ishlash uchun tuzilgan dasturni ko‘rib chiqamiz.

```
PROGRAM BookRone;
```

```
TYPE
```

```
StFIO = string[10];
```

```
StRhone = string[10]; RecBook = RECORD { fayl uchun yozuv }
```

```
FIO : StFio;
```

```
Rhone: StPhone;
```

```
END;
```

```
VAR
```

```
BookFile : File of RecBook; { RecBook faylli yozuv o‘zgaruvchilari }
```

```
Work : RecBook; { yozuvlarga o‘tish uchun o‘zgaruvchilar }
```

```
Ind : byte;
```

```
PROCEDURE inputRecE; { joriy yozuvni ekranga chiqarish }
```

```
BEGIN Read(BookFile, Work);
```

```
With Work do Writeln(‘FIO: ‘,FIO,’ telefon: ‘,Rhone);
```

```
END; { inputRecE }
```

```
PROCEDURE inputAllrec; { fayning hamma yozuvlarini ekranga chiqarish }
```

```
BEGIN
```

```
Seek(BookFile, 0); { faylni birinchi mantiqiy yozuvini o‘rnatish }
```

```
Writeln(‘*** Telefonlar haqidagi ma‘lumotnomani ekranga chiqarish’); WHILE (NOT Eof(BookFile)) DO inputRecE;
```

```
END; { inputAllrec }
```

```
PROCEDURE AddRec;
```

```
{ klaviaturadan sanaydi va faylga yozuvni qo‘shadi }
```

```
BEGIN
```

```
Writeln ( ‘ Fayl yozuvini chiqarish’);
```

```
WITH Work DO BEGIN
```

```
Write (‘Familiyasini kiriting:’);
```

```
Readln (FIO); { familiyalarni kiriting }
```

```
Write (‘Telefon nomerini kiriting: ‘);
```

```
Readln (Rhone); { telefon nomerini kiritish }
```

```
Write (BookFile,Work);
```

```
END; { with }
```

```
END; { AddRec }
```

```
PROCEDURE UrdateRec(NumRec : Longint);
```

```
{tartib nomeri bo‘yicha fayl yozuvini o‘zgartiring }
```

```
BEGIN
```

```
Seek(BookFile, NumRec); {tartib nomeri bo‘yicha fayl pozitsiyasini o‘rnatish }
```

```
Writeln (‘-- yozuvning eski qiymati --’);
```

```
inputRecE; { yozuvni chiqarish va fayl pozitsiyasini keyingi yozuvga o‘tkazish }
```

```
Seek(BooFile, NumRec); { oldingi pozitsiyaga qaytarish Writeln(‘ yangi qiymat’);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
AddRec; { yozuvni o‘zgargan qiymatini chiqarish }
END; { UrdateRec }
BEGIN { BookRhone }
Assign(BookFile, RHONE.DAT’); {RHONE.DAT nomli yangi fayl tashkil qilish}
Rewrite(BookFile);
Write(‘ RHONE.DAT’ yozuvining 5 ta yozuvini tashkil qilish);
FOR Ind := 1 TO 5 DO { klaviaturadan 5 ta yozuvni tashkil qilish} AddRec;
Write(‘Yozuvlarni tashkil qilish tugatildi. ... tugmachasini bosing’);
Readln;
inputAllrec; {barcha yozuvlarni ekranga chiqarish }
Write(‘Faylning 2 ta yozuvini o‘zgarishi. ... tugmachasini bosing’);
Readln;
UrdateRec(1); {1-chi tartib raqam faylning 2-chi mantiqiy yozuviga to‘g‘ri keladi}
Write(‘Davom ettirish uchun istalgan klavishani bosing...’);
Readln;
inputAllrec; { o‘zgargan faylni ekranga chiqarish }
Close(BookFile); { faylni yopish}
END. { BookRhone }
```

Kerak bo‘lsa, dasturni asosiy qism ishi tugaganidan so‘ng, RH-ONE.DAT faylini o‘chirish uchun Erase ni o‘qish mumkin. Toifalashgan fayllar o‘qish - yozish rejimida ish yuritishni tashkil qilishni taominlaydi. Bu imkoniyat amaliy masalalarni echishda qaysi toifadagi fayllar bilan ishlash qulayligini aniqlashda muxim ro‘l o‘ynaydi. Dasturni bajarilishida toifalashgan berilganlar mashina xotirasida berilganlar bilan bir xil ko‘rinishda boshladi shuning uchun qator oxiri yoki karetkani qaytarish boshqaruvini kuzatishning xojati yo‘q.

To‘gridan-to‘g‘ri kiritish fayllari bilan ishlashda qo‘shimcha quyidagi vositalardan foydalanish mumkin:

Truncate(VAR F)

Bu protsedura F faylining hamma komponentalarini yo‘qotadi. FileRos(VAR F) : L’ngint

Yuqoridagi funktsiya F faylining joriy pozitsiyasini L’ngint toifadagi qiymatda qaytaradi.

FileSize(VAR F) Funktsiya F faylining o‘lchovini L’ngint qiymatida qaytaradi. Bo‘sh fayl uchun FileSize 0 qiymatni qaytaradi. I‘result orqali tashqi tashuvchilarga xatolarni tarmoqlash uchun murojat qilinadi. Yozuv toifalashmagan faylning oxiriga yozilishi uchun, joriy fayl pozitsiyasini fayl oxiriga ko‘chirib o‘qish zarur. Yangi fayl tahlil etilayotganda har bir yozuv tashkillangandan so‘ng, bu avtomatik ravishda bajariladi. Agar fayl tashkil qilinib bo‘lgan va Seek buyicha o‘rnatilgan fayl pozitsiyasi (FileRos(F) < FileSize(F)) faylining oxirida tursa, holda fayl oxiri chaqiruvni Seek(F, FileSize(F)) ga ko‘chirish imkonini beradi. Har bir fayldagi fizik va mantiqiy yozuvlar soni mos tushadi, fizik yozuv pozitsiyalashgan tartib raqami mantiqiy yozuv raqamidan bitta kam bo‘ladi.

Misol. BookPhone dasturi yozuvni fayl oxiriga ulovchi AddRecTOEND protsedurasi bilan to‘ldirilishi mumkin.

```
PROCEDURE AddRecToEND;
```

```
BEGIN
```

```
Seek(BookFile, FileSize(BookFile));
```

```
{ joriy pozitsiyani fayl oxiriga o‘rnatish }
```

```
AddRec; END; { AddRecToEND }
```

Misol. Familiyalari bo‘yicha abonentlar ro‘yxatini chiqarish dasturi.

Berilganlar toifasi

oldingi dasturdagidek qoladi.

```
PROGRAM FindFIO;
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
CONST
CountRec: Integer = 0;
VAR
BookFile: File of RecBook;
Work: RecBook;
Maska: StFIO;
BEGIN
Assign(BookFile, 'Phone.dat');
Reset(BookFile); { bor fayl ochilgan }
Write('Izlash uchun familiyani kiriting:'o);
Readln(Maska);
WHILE (NOT Eof(BookFile)) DO BEGIN
{ tartiblashmagan faylning hamma yozuvlari bo'yicha }
Read(BookFile, Work);
WITH Work DO
IF Ros(Maska, FIO) <> 0 THEN BEGIN { topidi }
Ins(CountRec);
Writeln('Familiya: ',FIO,' telefon:',Phone);
END;
END; { while }
Writeln(',Maska,' = ',CountRec uchun yozuvlar soni );
Close(BookFile);
END. { FindFIO }
```

Yozuvlar tartiblashmagan familiyalar bo'yicha joylashganda, qidiruv barcha faylar bo'yicha ketma-ket olib boriladi. Bu tartibda yozuvni axtarish ancha qiyin kechganligi sababli, bosh faylga nisbatan indeksli fayllarni taqil etishga olib keladi. Indeksli fayllar aniq kalitlar asosida tartiblangan yozuv tartib nomerlarini o'z ichiga oladi. Bu tashqi tashuvchilarga murojat etishni engillashtiradi.

Toifalashmagan fayllar

Toifalashmagan fayl ochish, bloklar.

Maksimal tezlikda ishlaydigan fayllarni ko'rib o'tamiz. PascalABC tilidagi toifalashmagan faylni belgi va baytlar yig'ndisidan iborat deb qabul qilinadi.

Toifalashmagan faylni ochish uchun File rezervlashgan so'zidan foydalaniladi:

```
VAR
```

```
    UntedFile : File;
```

Toifalashmagan fayl to'gridan-to'g'ri kiritish fayli bo'lib, o'qish va yozish operatsiyasini bir vaqtning o'zida yozish imkoniyatini beradi. Bu fayllar uchun eng asosiy parametrlar baytlardagi yozuv uzunligi hisoblanadi. Yozuv uzunligi bir baytga ega bo'lgan toifalashmagan faylni ochish quyidagicha bajariladi:

```
    Rewrite(UntedFile, 1);
```

yoki

```
    Reset(UntedFile, 1);
```

Faqatgina toifalashmagan fayllar uchun qo'llaniladigan ikkinchi parametr faylning bir seansdagi yozuv uzunligini beradi.

Toifalashmagan fayllarni kiritish-chiqarish1

Toifalashmagan fayllarda kiritish-chiqarish operatsiyalarini Read va Write

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

operatorlaridan foydalanib, berilganlarni uzatish paytida yaxshi muvaffaqiyatga erishmaslik mumkin. Shuning uchun berilgan toifadagi fayllar uchun Turbo-PascalABCda kiritish-chiqarish operatsiyasini tezlatish uchun quyidagi 2ta yangi protseduralar ishlatiladi.

BlokRead(VAR F : File; VAR Buf; Count : word {;result:word});

Bu protsedura Buf o‘zgaruvchisining F faylidan aniq sondagi bloklarni xotiraga sanaydi.

Buf parametri F fayldan yigiladigan axborotlarni xoxlagan o‘zgaruvchisini bildiradi. Count parametri sanalayotgan bloklar sonini beradi. Rezult parametri shart bo‘lmagan parametr hisoblanib, protsedura chaqirilganda sanalgan yozuvlar sonini o‘z ichiga oladi.

Rezult parametrlarini qo‘llanilishi sanalgan bloklar soni Count parametrda berilgan bloklar sonidan kam bo‘lishligini oldindan aytadi. Agar Result chaqirish paytida ko‘rsatilgan bo‘lsa, u holda kiritish-chiqarish situatsiyasida xatolik bo‘lmaydi. O‘qish vaqtidagi shunga o‘xshash va boshqa xatolarni kuzatish uchun I‘result funktsiyaning {\$I-}, {\$I+} opsiyalari ishlatiladi.

Bl‘kWrite(VAR F : File; VAR Buf; Count:word {;result:word});

Bu protsedura Buf o‘zgaruvchisidagi yozuvlar sonini F fayliga tez uzatish uchun xizmat qiladi. Bl‘kWrite protsedurasining barcha parametrlari Bl‘kRead protsedurasining parametrlariga mos tushadi. Ikkala protsedura ham bloklarni kiritish-chiqarish operatsiyalarini bajaradi.

Bloklarning baytlardagi hajmi quyidagi formula orqali aniqlanadi:

Hajm Count *RecSize, bu erdaRecSize - ochilish paytida berilgan fayl yozuvining o‘lchovi. Bu protseduralarning qulayligi shundaki, foydalanuvchi fayl operatsiyalari uchun bufer o‘lchovini o‘zi aniqlashi mumkin. Bu imkoniyat resurslarni rejalashtirishda qo‘l keladi.

Misol. Bu dastur faylni berilgan hajm bo‘yicha bir necha bo‘laklarga bo‘lib, har bir bo‘lakni diskarga alohida yozish imkoniyatini beradi.

```
PROGRAM DivFile;
CONST
  SizeBufer = 25600;      { kiritish-chiqarish uchun bufer o‘lchovi }
TRUE
  BufFile = array[1..SizeBufer] of Byte;
VAR
  Bufer: BufFile;        { Bufer }
  W1, W2, W3 : word;
  KolSa, Siz : Longint;
  NameFile, NamePart: string[60]; { chiqarish va bosh fayl nomi }
  FileMain, FilePart: File;   { toifalashmagan fayllar uchun o‘zgaruvchilar }
  Flag1, Flag2 :
boolean;   { mantiqiy operatsiyalar uchun fayllar }
FUNCTION Answer : boolean;
{ Xatolik bo‘lgan paytda foydalanuvchi bilan muloiot uchun funktsiya }
VAR
  Ch : char;
BEGIN
  Write(‘Ishni davom ettirasizmi? (X/Y)’);
  REPEAT
    Readln(CH);
  CASE CH OF
    ‘DO,’DO: Answer := True;
    ‘N’,’n’: Answer := False;
  END; { case }
  UNTIL CH IN [‘X’,’x’,’Y’,’y’];
END; { Answer }
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
BEGIN { DivFile }
  Writeln('DivFile faylni bo‘laklarga bo‘lish’);
  Flag1 := True;
  WHILE Flag1 DO BEGIN { bosh fayllar bo‘yicha katta tsikl }      Write(bo‘lish uchun fayl
nomini kiriting:’);
    Readln(NameFile);
    Assign(FileMain, NameFile);
    Reset(FileMain, 1);      { bosh faylni ochish, yozuv uzunligi 1 }
    KolZar := FileSize(FileMain); { baytlardagi fayl o‘lchovi }
    WHILE KolSar > 0 DO BEGIN { bosh fayl oxirigacha }
      Writeln('Faylning bo‘lagi’,NameFile,’ =’,KolSar);
      Write('Faylni bo‘lish’);
      Flag2 := Answer;
      Writeln;
      Write('Chiqish faylini nomini kiriting‘o);
      Readln(NameRart);
      Assign(FilePart, Namerart);
      Rewrite(FileRart, 1);   { yozuv uzunligi 1 a teng bo‘lgan bosh fayl bo‘lagi uchun chiqish
faylini tashkil etish }
      IF Flag2 THEN BEGIN
        Write('Chiqish faylini o‘lchovini kiriting‘);
        Readln(Siz);
        IF Siz > KolZar THEN
          Siz:= KolZar;
        END ELSE
          Siz:= KolZar; { aks holda bo‘lak o‘lchovi bosh fayl qoldiqning o‘lchoviga teng }
        { Kiritish-chiqarish buferi o‘lchoviga korrektsiya kiritiladi }
        W1 := Siz DIV SizeBufer; { to‘la bufer soni }
        W2 := Siz MOD SizeBufer; { qoldiq }
        Write('Nusxa olish’);
        FOR W3 := 1 TO W1 DO BEGIN { to‘la buferlar bo‘yicha }
          Bl’kRead(FileMain, Bufer, SizeBufer);
          Bl’kWrite(FilePart, Bufer, SizeBufer);
          Write('+');
        END;
        IF W2 <> 0 THEN
          BEGIN
            BlokRead(FileMain, Bufer, W2);
            BlokWrite(FilePart, Bufer, W2);
            Write('+');
          END; { if W2 <> 0 }
        Writeln;
        Closes(FilePart);      { bo‘lak uchun faylni yopish }
        KolZar := KolZar - Siz; { bosh fayl qoldiqni hisobi }
      END; { while KolZar > 0 }
      Closes(FileMain);
      Write('Yangi faylni bo‘lish, ‘);
      Flag1 := Answer;
    END; { while Flag1 }
  END. { DivFile }
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Bunday dasturga katta hajmli diskdagi berilganlar to‘plamini kichik hajmli tashuvchilarga olib o‘tishda extiyoj tug‘iladi Dastur matnini ko‘paytirmaslik uchun kiritish-chiqarish xatoliklari tekshirilmaydi. Bularni lokalizasiyalashda IOresult standart funksiyasidan foydalanish mumkin. Agar Bl’kRead va Bl’kWrite 4 ta parametrga kengaytirilgan chaiqruv qo‘llanilsa, u holda dastur ichida uzatilayot-ganayotgan axborotlar sonini nazorat qilish mumkin.

Matnli fayllar

Matnli faylni 0 dan 256 gacha bo‘lgan belgilar qatori ketma-ketligi deb qarash mumkin. Matnli qatorni yozish uchun Text standart toifasidan foydalaniladi

VAR

F: text; {F- fayl o‘zgaruvchisi }

Har bir qator qator oxiri markeri bilan tamomlanadi. Amaliyotda bunday markerni ikkita belgi ketma-ketligi tashkil qiladi: Satrga o‘tish chr(13) va karretkani qaytarish chr(10). Ushbu ikkita belgi Matnli fayllar ustidagi standart harakatlarni belgilaydi. SYSTEM modulidagi standart ochiladigan Inut va ‘utut fayllari Text toifali bo‘ladi.

Matnli fayllarni kiritish va chiqarish

Aend(VAR F : text), Readln(VAR F : text; V1 [,V2,...Vn]), Writeln(VAR F : text; V1 [,V2,...Vn]), Yeoln(VAR F: text), Eof(VAR F:text), SeekE‘ln(VAR F: text), SeekEof(VAR F: text), Fiush(VAR F: text), SetTextBuf(VAR F : text; VAR Buf[; Size : word]) protsedura va funksiyalari

Matli fayllar o‘z spetsifikasiga ega. Read o‘qish va Write yozish protseduralarining maxsus kengaytmasi belgili bo‘lmagan toifalar bilan ishlash imkoniyatini beradi.

Read(A, Ww) larning chaqirilishi, bu erda Ww - word toifasidagi o‘zgaruvchi, sonlar ketma-ketligini F faylidan o‘qib, Ww. O‘zgaruvchisiga o‘zlashtiradi. Boshqa hollarda dasturda xatolikka olib keladi.

Matnli fayllarni ochishni standart usullarda olib borish mumkin:

Fayl o‘zgaruvchisiga mos fayl nomini qo‘yish (Assign protsedura-si), yangi matn faylini ochish (Rewrite protsedurasi); yoki fayl o‘zgaruvchisiga mos fayl nomini qo‘yish (Assign protsedurasi), bor faylni ochish (protsedura Reset).

Matnli fayllar o‘zining spetsifikasiyasiga qarab, o‘qish yoki yozish operatsiyalaridan bittasi bilan ishlash imkoniyatini beradi. Shuning uchun matnli fayllar bilan ishlashda yana bitta protsedura, faylni ochish protsedurasi paydo bo‘ladi:

Aend(VAR F : text);

Bu protsedura faylni ochib, kursorni fayl oxiriga keltirib qo‘yadi. Aend protsedurasiga qo‘yiladigan talablar ham xuddi Reset va Rewrite protseduralariga qo‘yilgan talablarga o‘xshash.

Matnli fayllarni qayta ishlash uchun Read va Write protseduralari qo‘llaniladi. Readln protsedurasi qatorlarni bo‘luvchi maxsus til vositasi hisoblanadi. Uning vazifasi xuddi Read ning vazifasiga o‘xshash va o‘qishni qator oxirining markerigacha olib boradi va boshqa yangi iatorga o‘tishni ta‘minlayladi. Writeln, protsedurasi hamma kattaliklarni yozishni ta‘minlaydi. Protseduralar yozilishining ko‘rinishi quyidagicha:

Readln(VAR F : text; V1 [,V2,...Vn]);

Writeln(VAR F : text; V1 [,V2,...Vn]);

bu erda V1...Vn turli toifadagi o‘zgaruvchilar. Read va Readln protseduralari orasidagi fari nimada degan savolga quyidagicha javob berish mumkin. Read protsedurasi berilganlarni hammasini bir qatorga olib kirish imkonini beradi Readln operatsiyasi esa albatta bir qator

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

tugagandan keyin avtomatik ravishda ikkinchi qatorga o‘tishni ta‘minlaydi, yaniy berilganlarni turli qatorlardan olib kirish mumkin. Write va Writeln protseduralarida ham xuddi shunday.

Kiritish chiqarish operatsiyalarini bajarayotgan paytda maxsus E’ln, Eof, SeekE’ln, SeekEof funksiyalaridan tashkil topgan til vositasini ishllash mumkin.

Agarda faylning joriy pozitsiyasi fayl oxiri markerida turgan bo‘lsa Yeoln(VAR F: text) funksiyasi, Truening bulev qiymatni qaytaradi. Qolgan hamma holatlarda funktsiyaning qiymati False bo‘ladi.

Agarda kursor oxirgi komponentadan keyin turgan bo‘lsa, Eof(VAR F: text) Truening bulev qiymatni qaytaradi, aks holda False bo‘ladi.

Misol. Matnli faylning birinchi qatoridan oltita ketma-ket belgilarni o‘qish.

```
VAR
  F: text;
  St: string[6];
BEGIN  Assign(F, 'EHAMRLE.RAS'); { EHAMRLE.RAS fayli bor bo‘lishi kerak }
Reset(F);
  WHILE NOT Eofln(F) DO BEGIN { birinchi qator oxirini tekshirish }
    Read(F, St);
    Writeln('St = ', St); { ekranga chiqarish }
  END;
  Readln(F);           { 2- qatorga o‘tish }
  Closse(F);
END.
```

SeekE’ln(VAR F: text) funktsiya kursor qator oxirida turganda True ning bulev qiymatni qaytaradi. Aksi holda funktsiya False qiymatni qaytaradi.

SeekEof(VAR A: text) funktsiyasi agarda ko‘rsatkich fayl oxiri markerida turgan bo‘lsa Truening qiymatni qaytaradi. Bu funktsiyalarni ishlash yoki fayllardan sonli qiymatlarni qo‘shishda qulaylik yaratadi. Diskdagi matnli fayllar bilan tashqi dastur orasidagi axborot almashuvini ko‘rib chiqamiz. Berilganlarni olib kirish uchun operativ xotiradagi kiritish- chiqarish buferidan foydalanamiz.

PascalABC tilida standart kiritish -chiqarish buferi 128 bayt hajmga ega bo‘ladi. Har qaysi ochilgan faylga o‘zining buferi qo‘yiladi. Tashqi axborot tashuvchilariga murojat qilishni optimallashtirish vazifasi qo‘yiladi.

Masalan, Writeln protsedurasi hamma berilganlarni buferga ketma-ket yozadi. Oxirgi bufer band bo‘lgan holatdagina yozuvlar tashqi qurilmaga yoziladi. Yozuvlarni diskga ko‘chirib bo‘lgandan so‘ng bufer keyingi axborot qabul qilish uchun bo‘shaydi. Yozuvlarni bu tartibda yozish juda ko‘p variantni oladi Shuning uchun maxsus kiritilgan Flush(VAR F: text) protsedurasi Write va Writeln protseduralari yordamida yozuvlarni diskdagi fizik yozuvlar bilan boglash imkoniyatini beradi. Protsedurani chaqirish yozish uchun berilgan hamma belgi-larni haqiqiyiligini ularni tashqi faylga ezilishini garantiyalaydi. Bu prosedularni Rewrite va Aend. Protseduralari orqali ochilgan matnli fayllar uchun ishllash mumkin. Amaliy dasturlarda bu protseduralardan qisman foydalanilmaydi.

Amaliyotda matnli fayllarni qayta ishlash xotiradagi hamma fayllarni sanab chiqish va undan keyin fayllarni diskga yozish uchun xizmat qiladi.

SetTextBuf(VAR F : text; VAR Buf[; Size : word]) protsedura F fayliga Size hajmdagi o‘zining kiritish- chiqarish buferini o‘rnatish imkonini beradi Size parametri chaqirishda tashlab ketilishi mumkin.U holda bufer o‘lchovi SizeOf(Buf) ga mos bo‘ladi. F fayl o‘zgaruvchisiga SetTextBuf ning ishlashi o‘zgaruvchiga yangi tashqi faylni ishllashgacha tarqalishi mumkin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Misol. Kiritish- chiqarish buferining vazifasi

VAR

F: text;

Sh: char;

Buf: array[1..2048] of char; { Bufer 2 Kbait }

BEGIN

Assign(F, RaramStr(1)); { buyruq qatoridan faylga nom qo'yish }

SetTextBuf(F, Buf); { faylga 2 Kbait buferni o'rnatish }

Reset(f);

WHILE NOT Eof(F) DO BEGIN { faylni ekranga chiqarish }

Read(F, Ch);

Write(Ch);

END;

END.

Nazorat savollari

1. Matnli fayl deb qanday faylga aytiladi?
2. Satr oxiri markeri nima?
3. Aend(VAR F : text) ni vazifasi nima?
4. Readln(VAR A : text; V1 [,V2,...Vn]) ni vazifasi nima?
5. SeekE'ln(VAR A: text) nima vazifa bajaradi?
6. Writeln(VAR F : text; V1 [,V2,...Vn]) protsedurasini ishlashini tushuntiring.
7. E'lon(VAR F: text) va Eof(VAR F:text) protseduralari qanday vazifa bajaradi?
8. SeekE'ln(VAR F: text), SeekEof(VAR F: text) protseduralarining vazifasini tushuntiring.
9. Flush(VAR F: text) protsedurasining ishlashini tushuntiring.
10. SetTextBuf(VAR F : text; VAR Buf[; Size : word]) protseduraning ishlashini tushuntiring.

MA'RUZA №12

GRAFIKA MODULI VA ULAR BILAN ISHLASH.

Reja:

1. Pascal ABC va Turbo Paskal dasturlash tillarining standart modullari.
2. Dastur ishini bajaruvchi protseduralar.
3. Parametrlar bilan ishlash funksiyalari.

Turbo Paskalda modullar. Standart modullar

Shaxsiy kompyuterlarning eng katta kamchiligi ularda amaliy dasturlar kutubxonasining to'liq emasligidadir. Katta EHMlarda dastur tuzuvchilar uchun juda katta dasturlar kutubxonasi xizmat qilar va ulardan foydalanib tuzilgan dasturlar o'zlarining ishonchlilik darajasi bilan yuqori turar edi. SHaxsiy kompyuterlarning bu kamchiligini yo'qotish uchun Turbo-Paskalda modullar tushunchasi kiritilgan. Umuman olganda, har bir malakali dastur tuzuvchi o'z dasturini protsedura va funksiyalardan foydalanib tuzadi. Lekin, bu protsedura va funksiyalardan boshqa dasturlarda foydalanish uchun ularning matnlarini qayta ko'chirib yozish lozim bo'ladi.

Turbo Paskalda bu masalani yechish uchun modullar yaratilib, ularni kompilyatsiya qilinadi va bu moduldan boshqa dasturlarda bemalol foydalanilaveriladi.

Turbo-Paskal tilining yaratuvchilari quyidagi zarur va foydali modullarni yaratib, dasturchilar uchun juda katta qulayliklar yaratishgan:

1. **SYSTEM** moduli – standart protsedura va funksiyalarni o'z ichiga olib, avtomatik tarzda barcha dasturlar uchun ochiqdir;
2. **DOS** moduli – MS DOS operatsion tizimsi bilan ishlashni tashkil qiluvchi funksiya va protseduralardan tashkil topgan;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3. **CRT** moduli – ekran, klaviatura va IBM rusumidagi kompyuterlarning tovushli dinamik bilan ishlash protseduralarini o‘z ichiga olgan;
4. **GRAPH** moduli – kompyuterning grafik imkoniyatlaridan foydalanib yaratilgan funksiya va protseduralarning katta to‘plami;
5. **PRINTER** moduli – bu kichkinagina modul printer qurilmasi bilan ishlashni osonlashtiradi;
6. **OVERLAY** moduli – katta dasturlarni bir nechta bo‘laklarga ajratishning kuchli vositasi bo‘lib, bir qancha protseduralar va funksiyalardan tashkil topgan.

Foydalanuvchi modullarini yaratish

Modullardan foydalanish uchun dastur sarlavhasidan

Program <dastur nomi>;

keyin quyidagi qator yozilishi kerak:

Uses <modul ismi>;

Agar dasturda bir nechta modul ishlatilsa, ularning ismlari ketma-ket yozib qo‘yiladi:

Uses <modul ismi1>,<modul ismi2>,>...,<modul ismiN>;

Turbo-Paskal bizga o‘zimizning modullarimizni yaratib olish imkonini ham beradi.

Foydalanuvchi modullari quyidagi strukturada bo‘ladi:

Unit <modul ismi>;

Interface

...
{ ochiq e‘lonlar bo‘limi - interfeys seksiyasi }

...
Implementation

...
{ yopiq e‘lonlar bo‘limi }

...
Begin

...
{ Initsializatsiya bo‘limi }

...
End.

Agar modul o‘z ichida boshqa modullardan foydalansa *Interface* xizmatchi so‘zidan keyin *Uses* <modullar ro‘yxati>; yoziladi.

Interfeysli bo‘lim modulning bir qismi bo‘lib, ***Interface*** va ***Implementation*** so‘zlari orasida joylashadi. Bu bo‘limda o‘zgarmaslar, ma‘lumotlar turi, o‘zgaruvchilar, protsedura va funksiyalarni aniqlash mumkin. Bu kiritilganlar mazkur modulda qatnashuvchi barcha dasturlar va modullarda bemalol ishlatilishi mumkin. Bo‘limda sanab o‘tilgan protsedura va funksiyalarning tana qismlari ***Implementation*** so‘zidan keyin aniqlanadi (ularning sarlavhalari aynan saqlanib qolishi kerak). Bu bo‘limda ham, faqat shu bo‘lim uchungina "Ko‘rinadigan" (ishlatishi mumkin bo‘lgan) e‘lonlar bo‘limi qatnashishi mumkin. Initsializatsiya seksiyasi ***Begin*** va ***End*** so‘zlari ichiga olib yoziladi. Agar ***Begin*** so‘zi tushirib qoldirilgan bo‘lsa, demak bu seksiya yo‘q hisoblanadi. Initsializatsiya seksiyasida boshqarishni asosiy dasturga uzatguncha bajariladigan operatorlar joylashgan bo‘ladi. Bu operatorlar asosan dasturni ishga tushirishga tayyorlab beradi.

Misol sifatida X va Y buning sonlarining maksimumi va minimumini aniqlovchi modulni yarataylik:

Unit Stud;

Interface {ochiq e‘lonlar bo‘limi – interfeys seksiyasi}

function min(x,y:integer):integer;

function max(x,y:integer):integer;

Implementation {yopiq e‘lonlar bo‘limi}

function min(x,y:integer):integer;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Begin
  if x<=y      then min:=x else min:=y;
End;
function max(x,y:integer):integer;
Begin
  if x>y then max:=x else max:=y;
End;
Begin
  {Initsializatsiya seksiyasi yo‘q}
End.
```

Biz zarur modulni hosil qildik, endi uni kompilyatsiya qilishimiz lozim. Kompilyatsiya natijasida **Stud.tpu** ismli fayl hosil qilinishi kerak. Kompilyatsiya qilinmagan modulning ismi esa shunga mos holda **Stud.pas** bo‘lishi kerak.

Bu moduldan foydalanish dasturi quyidagicha bo‘lishi mumkin:

```
Uses Stud;
Var
  A,b,c,d:integer;
Begin
  Write(‘A va B larni kiriting >’);
  Readln(a,b);
  C:=max(a,b);
  Writeln(‘Max=’,C);
  C:=min(a,b);
  Writeln(‘Min=’,C);
  D:=max(a,b)+min(a,b);
  Writeln(‘Max+Min=’,D);
End.
```

Quyida esa ekran rangini tanlash moduli misol sifatida ko‘rsatilgan:

```
Unit Colors;
Interfase
Type
  Colortype =Array[0..15] of Byte;
Const
  Black:byte=0;blue:byte=1;
  Green:byte=2;cyan:byte=3;
  Red:byte=4;magenta:byte=5;
  Brown:byte=6;lightgray:byte=7;
  Darkgray:byte=8;lightblue:byte=9;
  Lightgreen:byte=10;lightcyan:byte=11;
  Lightred:byte=12;lightmagenta:byte=13;
  Yellow:byte=14;white:byte=15;
Var
  Currcolors:colortype absolute Black;
  Procedure setMonoColors;
  Procedure setColorColors;
Implementation
Const
  ColorColors:Colortype=(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);
  MonoColors:ColorType=(0,1,7,7,7,7,7,7,7,7,7,7,7,7,15,15);
Procedure SetMonoColors;
Begin
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
CurrColors:=MonoColors;  
End;  
Procedure SetColorColors;  
Begin  
    CurrColors:=ColorColors;  
End;  
Var  
    Ch:Char;  
Begin  
    Write(  
        Readln(ch);  
        If ch in ['M', 'm', 'M', 'm'] then SetMonoColors;  
End.
```

Dastur ishini bajaruvchi protseduralar

YUqorida aytganimizdek, System modulining protseduralari va funksiyalari barcha dasturlar uchun ochiq bo‘lib, ulardan keng foydalanish mumkin. System modulining ismini modullar ro‘yxatida ko‘rsatish shart emas.

Quyida ushbu modulning ma‘lum bir qism protseduralar va funksiyalari bilan qisqacha tanishib chiqamiz:

1) Dastur ishini bajarish protseduralari.

Exit protsedurasi

Vazifasi: aktiv ishchi blokidan chiqish, bajarilayotgan ishni yakunlash;

Aniqlanishi: **Exit**.

Halt protsedurasi

Vazifasi: dastur bajarilishini to‘xtatadi va OS ga boshqarishni qaytaradi;

Aniqlanishi: **Halt[(ExitCode:Word)];**

bu yerda majburiy bo‘lmagan (o‘rta qavs majburiy emas belgisi) ExitCode parametri dasturning yakunlanish kodini beradi, agar bu parametr bo‘lmasa bu kod nolga teng bo‘ladi.

RunError protsedurasi

Vazifasi: dasturning bajarilishini to‘xtatadi va bajarilish vaqtidagi xatolarni aniqlaydi;

Aniqlanishi: **RunError[(ErrorCode:Word)];**

bu yerda ErrorCode parametrlining xatolik nomeri haqidagi ma‘lumoti ekranga chop etiladi.

Turlarni almashtirish funksiyalari

Chr funksiyasi

Vazifasi: ASCII jadvalidagi tartib raqami berilgan butun songa mos bo‘lgan belgini aniqlaydi;

Aniqlanishi: **Chr(N:Byte):Char;**

bu yerda N belgining ASCII jadvalidagi tartib raqamini ifodalovchi butun, musbat son.

Ord funksiyasi

Vazifasi: sanalma turli qiymatlar bo‘yicha uning tartib sonini aniqlash;

Aniqlanishi: **Ord(X):LongInt;**

bu yerda X – sanalma turli qiymat.

Round funksiyasi

Vazifasi: haqiqiy turli qiymatni yaxlitlab, katta butun, son hosil qiladi;

Aniqlanishi: **Round(X:Real):LongInt;**

Trunc funksiyasi

Vazifasi: haqiqiy turli qiymatning kasr qismini tashlab yuborib, butun son hosil qiladi;

Aniqlanishi: **Trunc(X:Real):LongInt;**

Arifmetik funksiyalar bu funksiyalarning yozilishi va ulardan foydalanish qoidalari 3-bobdagi o‘zlashtirish operatori mavzusida to‘liq berib o‘tilganligi uchun, ularga to‘xtalib o‘tirmaymiz.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Satrlar bilan ishlash funksiyalari va protseduralari.

ConCat funksiyasi

Vazifasi: satrlarni ketma-ket ulaydi;

Aniqlanishi: ConCat(S1[,S2,...,SN]:String):String;

bu yerda S1 qatorni keyingi sanab o‘tilgan qatorlar bilan ularni yozilish tartibida ulaydi.

Copy funksiyasi

Vazifasi: satr ichidan yangi satr hosil qilish;

Aniqlanishi: Copy(S:String;Index,Count:Ineger):String;

bu yerda S – berilgan satr;

Index – S satrining nechanchi belgisidan boshlab, yangi satr hosil qilish kerakligini aniqlaydi;

Count – yangi satrdagi belgilar soni.

Delete protsedurasi

Vazifasi: berilgan satr ichidan satr ostini olib tashlaydi;

Aniqlanishi: Delete(Var S:String;Index:Integer;Count:Integer);

bu yerda S – berilgan satr;

Index – shu tartib raqamli belgidan boshlab, S satrdan satr osti olib tashlanadi;

Count – olib tashlanadigan satrdagi belgilar soni.

Insert protsedurasi

Vazifasi: berilgan satrga yangi satr qo‘shadi;

Aniqlanishi: Insert(S1:String; Var S2:String; Index:Integer);

bu yerda

S2 – berilgan satr;

S1 – qo‘shiladigan satr;

Index – S2 satrning qaysi tartib raqamli hadidan boshlab yangi satr qo‘shilishini anglatadi.

Length funksiyasi

Vazifasi: satr uzunligini aniqlaydi;

Aniqlanishi: Length(S:String):Integer;

bu yerda S satridagi belgilarning sonini aniqlanadi:

Pos funksiyasi

Vazifasi: satrdan satr ostini qidiradi;

Aniqlanishi: Pos(SubStr,S:String):Byte;

bu yerda SubStr – S satrda qidirilayotgan satr osti.

Agar SubStr satr osti S satrida topilsa, Pos funksiyasi mos kelgan birinchi belgining tartib raqamini beradi, agar bu satr osti S satrda bo‘lmasa, funksiya nul qiymat beradi.

Str protsedurasi

Vazifasi: sonli qiymatni uning satrli ko‘rinishiga o‘tkazadi;

Aniqlanishi: Str(X[:Width[:decimals]];Var S:String);

bu yerda yozilishi shart bo‘lmagan Width va decimals parametrlari mos ravishda, S satrining hadlar sonini va haqiqiy sonning verguldan keyingi xadlar sonini ifodalaydi.

Val protsedurasi

Vazifasi: satrli qiymatni uning sonli ko‘rinishiga o‘tkazadi;

Aniqlanishi: Val(S:String;Var V; Var Code:Integer);

bu yerda S – berilgan satr;

V – S satrning uning sonli ko‘rinishiga o‘tkazganidan keyin hosil bo‘lgan sonni saqlash joyi;

Code – butun turli o‘zgaruvchi.

Agar S satrni songa aylantirib bo‘lmasa Val protsedurasi bajarilganidan so‘ng Code o‘zgaruvchisi nul qiymatni qabul qiladi.

Parametrlar bilan ishlash funksiyalari

Dasturni ishga tushirishda unga uzatiladigan qiymatlarni parametrlar deb hisoblaymiz.

Misol: Tizimli dasturlashning quyidagi Copy buyrug‘ini ko‘raylik:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Copy file.dat c:\prog\file2.dat

ya’ni, aktivlashgan katalogdagi file.dat faylining nusxasini «S» diskning prog katalogiga file2.dat nomi bilan ko‘chirish. Bu yerda Copy buyrug‘iga parametr sifatida ”file1.dat” va “file2.dat” qatori uzatilmoqda.

Turbo-Paskalda parametrlar bilan ishlash uchun ParamCount va ParamStr funksiyalari mavjud.

ParamCount funksiyasi

Vazifasi: buyruqli qatordan dasturga uzatilgan parametrlar sonini aniqlaydi;

Aniqlanishi: ParamCount:Word;

ParamStr funksiyasi

Vazifasi: ko‘rsatilgan nomerdagi parametрни aniqlaydi;

Aniqlanishi: ParamStr(N:Integer):String;

Adreslar bilan ishlash funksiyasi

Addr funksiyasi

Vazifasi: ko‘rsatilgan ob’ektning adresini aniqlaydi;

Aniqlanishi: Addr(X):Pointer;

bu yerda X – ixtiyoriy turli o‘zgaruvchi yoki dasturda e’lon qilingan protsedura yoki funksiya ismi.

Seg funksiyasi

Vazifasi: ko‘rsatilgan o‘zgaruvchi adresining segment qiymatini aniqlaydi;

Aniqlanishi: Seg(X):Word;

bu yerda X – ixtiyoriy turli o‘zgaruvchi yoki dasturda e’lon qilingan protsedura yoki funksiya ismi.

Modulning boshqa protsedura va funksiyalari

FillChar protsedurasi

Vazifasi: ko‘rsatilgan qiymat bilan ketma-ket kelgan va chekli sondagi baytlarni to‘ldiradi;

Aniqlanishi: FillChar(Var X;Count:Word;Value);

Hi funksiyasi

Vazifasi: argumentning katta baytini aniqlaydi;

Aniqlanishi: Hi(X):Byte;

Lo funksiyasi

Vazifasi: argumentning kichik baytini aniqlaydi;

Aniqlanishi: Lo(X):Byte;

Move protsedurasi

Vazifasi: ko‘rsatilgan miqdordagi baytlardan tezkor (operativ) xotiraning bir qismidan ikkinchi qismiga nusxa ko‘chiradi;

Aniqlanishi: Move(Var Source, Dest; Count: Word);

Random funksiyasi

Vazifasi: tasodifiy sonni aniqlaydi;

Aniqlanishi: Random([Range: Word]);

bu yerda aniqlanadigan tasodifiy son 0 va Range sonlari oralig‘ida yotadi.

Randomize protsedurasi

Vazifasi: tasodifiy sonni hosil qiluvchi generatorni ishga tushiradi;

Aniqlanishi: Randomize;

UpCase funksiyasi

Vazifasi: kichik lotin harflarni katta harflarga o‘tkazadi;

Aniqlanishi: UpCase(Ch:Char):Char; funksiya rus alfaviti harflarini ham katta harflarga o‘tkazadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

MA’RUZA №13

MAVZU: PASKAL TILIDA SODDA SHAKLLAR CHIZISH.

Reja:

1. Pascal ABC va Turbo Pascal dasturlash tlining grafik imkoniyatlari.
2. Grafik operatorlar va ularning vazifalari.
3. Standart grafik protseduralar.

Tayanch so‘z va iboralar: Pascal ABC, Turbo Pascal, grafik imkoniyatlar, operatorlar, figuralar, chizmalar.

Grafik figuralarni ko‘rish

Rectangle, Bar, Var3D, DrawRoly, FillRoly,

GRAPH bibliotekasida berilgan parametrlar asosida turli figuralarni ko‘rish protseduralari mavjud. Chiziilarning rangi stili va qalinligi SetColor i SetFillRattern, SetFillStyle protsedurasi yordamida o‘rnatiladi.

To‘g‘ri burchaqlarni ko‘rish uchun bir nechta protseduralar mavjud. Bulardan birinchisi bir o‘lchovli to‘g‘ri burchaqni chizish:

Rectangle(X1, Y1, X2, Y2 : Integer),

Bu erda X1, Y1 - lar chap yuqori burchak koordinatalari, X2, Y2 - lar chap past burchak koordinatalari.

Bar(x1, y1, x2, y2: Integer) protsedurasi yordamida yuqoridagidan ham effekliroq bo‘lgan ustunlari bo‘yalgan to‘g‘ri burchaklarni ko‘rish mumkin.

Bo‘yash rangi SetFillStyle bilan o‘rnatiladi.

Var3D(x1, u1, x2, u2: Integer; Derth: word; Tor: voolean) - Shch o‘lchovli Bo‘yalgan to‘g‘ri burchakni chizadi. Derth parametri o‘ch o‘lchovli kontr chuqurligi parametrini ko‘rsatuvchi piksellar sonini beradi.

Ko‘p hollarda uning qiymati to‘g‘ri burchaqning to‘rttdan bir kengligi qiymatiga teng: Derth := (X2-X1) DIV 4; Masalan: SetFillStyle(1,3); Bar3D(10,10,50,100,10,True);

To‘g‘ri burchaklarni Line yoki LineTo yordamida chizish mumkin. Lekin . Turbo PascalABC da har qanday ko‘p burchaklarni chizish uchun DrawRoly protsedurasi mavjud. Uning yozilishi quyidagicha:

DrawRoly(NumRoints: word; VAR RolyRoints)

RolyRoints parametra toifalashmagan parametr hisoblanib ko‘p burchakning har bir kesisish kordinatisini o‘zichiga oladi.

NumRoints parametr RolyRoints. ning kordiNOTalar sonini beradi.

Yoy va aylanalar chizish

Circle, Arc, GetArcCoords, Ellirse, Fillellirse, RieSlice, Sector.

Ey va aylanalarni chizishda burchaklar qiymatni berish uchun qutb kordinata sistemasi qo‘llaniladi. Aylana chizish protsedurasining yozilishi quyidagicha:

Circle (X, Y, Radius: word) Bu erda X va Y - aylana markazi, Radius - uning radiusi.

Masalan, bu fragment 450,100 markazli nuqtada turgan 50 piksel radiusga teng bo‘lgan ochiq zangor rangdagi aylana chizadi:

SetColor(LightGreen);

Circle(450,100,50);

PsevDO hajmli figuralarni chizish uchun yoylardan foydalanadi. Ularni quyidagi protsedura yordamida chizish mumkin:

Arc (X,Y: Integer; StAngle, ENDAngle, Radius: word)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

bu erda X, Y - aylana markazi, $StAngle$ va $ENDAngle$ - boshlanich va oxirgi burchak, $Radius$ - radius. Rang

$SetColor$. yordamida o‘rnatiladi. Agarda $StAngle = 0$ va $Angle = 359$, bo‘lsa u holda to‘liq aylana chiziladi. Misol tariqasida oldindan $Circle(450, 100, 50)$ yordamida chizilgan aylanada 0 dan 90 gradusgacha bo‘lgan qizil rangdagi yoyni chizish dasturini ko‘rib chiqamiz.

```
SetColor(Red);
```

```
Arc(450,100,0,90,50);
```

Har xil turli egri chiziqlar bilan ishlashda koordinatalar to‘g‘risidagi axborotni bilish talab qilinadi va buning uchun Arc ga murojat qilinadi. Buni olish $GetArcCoords$ ($VAR ArcCoords$

:

$ArcCoordsType$) protsedurani bajarish bilan amalga oshiriladi.

$ArcCoords$ o‘zgaruvchi standart toifalarga kiradi.

```
TRUE
```

```
ArcCoordsType = RECORD
```

```
  x, y      : Integer;
```

```
  xStart, yStart : Integer;
```

```
  xEND, yEND   : Integer
```

```
END;
```

Ellipsislar chizish uchun $Ellirse$ ($X, Y: Integer; StAngle, ENDAngle: word; XR, UR: word$) protsedurasidan foydalanamiz.

Bu erda X, Y - displey koordinatalaridagi ellipsisning markazi, XR va UR - lar gorizontal va vertikal shilar. Ellipsisning eyi $StAngle$ boshlanich burchagidan to $ENDAngle$ oxirgi burchakkacha joriy rangda chiziladi.

$StAngle=0$ va $ENDAngle=360$ larning qiymatlari to‘lla elipsis chizishga olib keladi.

Misol: och havo rangli ellipsisni chizish.

```
SetColor(LightCyan);
```

```
Ellirse(100,100,0,360,30,50);
```

Ellipsisning ichidagi fon ekran foni bilan mos tushadi. Bo‘yalgan ellipsisni hosil qilish uchun maxsus $FillEllirse$ ($X, Y: Integer; xR, yR: word$) protsedurasi ishlatiladi.

Bu erda X, Y - displey koordinatalaridagi ellipsisning markazi, xR va yR - lar gorizontal va vertikal shilar. Tshldirish $SetFillStyle$ yoki $SetColor$: protsedurasi yordamida o‘rnatiladi.

```
SetFillStyle(WideDotFill,Green); { to‘ldirish stilini o‘rnatish }
```

```
SetColor(LightRed);           { ellipsisni chizish uchun rang }
```

```
FillEllirse(300,150,50,50);
```

Bu frangmentda ellipsis ochiq qizil rangda, egri chiziq esa $WideDotFill$ dagi yashil rang bilan chiziladi.

$RieSlice$ ($X, Y: Integer; StAngle, ENDAngle, Radius: word$) protsedurasi aylana sektorini maska bilan to‘ldiradi va chizadi. Bu erda X, Y - aylana markazi,

sektor esa $StAngle$ dan boshlanib, $ENDAngle$. gacha chiziladi. Sektor konturi joriy rang bilan chiziladi.

1. To‘g‘ri va egri chiziq funktsiyalar grafigini chizishni dasturlash.

Grafik tasvirlarni ekranda ifodalash uchun Turbo PascalABC da $GRAPH$ moduli qo‘llaniladi. Unda 79 ta protsedura va funktsiyalar mavjud.

$GRAPH$ moduli $USES$ suzi orqali dasturga ulanadi:

```
USES Grah;
```

Drayverlar

Grafika drayverlari, standart drayverlar, foydalanuvchi drayverlari

Grafika drayverlari quyidagi (. jad. 13.1) da keltirilgan.

Jad. Grafika dayverlari

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Drayver	Adapter
CGA.BGI	IBM CGA, MCGA
EGAVGA.BGI	IBM EGA, VGA
HERC.BGI	Hercules (moNo)
ATT.BGI	AT&T 6300 (400 iator)
RC3270.BGI	IBM 3270 RC
IBM8514.BGI	IBM 8514

Shrifltlar

Standart shrifltlar, foydalanuvchining shrifltlari,

Matnlarni grafika rejimlarida chiqarish standart yoki foydalanuvchining shrifltlari orqali amalga oshirilishi mumkin.

Jad.Standart ShIFTLlar

Shrift	Fayl
TrirlexFont	TRIr.CHR
SmallFont	LITT.CHR
SansSerifFont	SANS.CHR
GothicFont	GOTH.CHR

Videorejimlar

Videorejim, drayver toifasi,

Drayver toifalari va rejimlari belgili konstantalar toifada yoki sonlar bilan berilishi mumkin. Videorejimlari aniqlovchi konstantlar jad. 13.3 da berilgan.

Jad.Videorejimlar

Drayver	Rejim	Imkoniyati	Fayl
CGA (1)	CGAC0, CGAHi	320x200 640x200)	CGA.BGI
EGA (3)	EGALo, EGAHi	640x200(640x350)	EGAVGA.BGI
VGA (9)	VGALo, VGAHi	640x200(640x350)	EGAVGA.BGI
HERC (8)	HERCMONOHl	720x348	HERC.BGI

Videorejimni inisializasiyalash

InitGrah, GetDriverName, GetGrahMode, GetModeRange, DetectGrah, GetMaxMode, SetGrahMode, RestoreSRTMode, videosaoifa, faol videosahifa, ko‘rinadigan videosaoifa, SetActiveRage SetVisualRage, GrahRezult, GrahErrorMsg

Birinchi navbatda grafik rejimlardan birortasini o‘rnatuvchi InitGrah protsedurasi chaqiriladi:
Yozilishi

InitGrah(VAR DriverVar, ModeVar: Integer; RathToDriver: string);

DriverVar i ModeVar o‘zgariuvchilari drayver va rejimlarning 13.3 tab. da keltirilgan qiymatlarini beradi.

Masalan: DriverVar := VGA; ModeVar := VGALo;

Display toifaini bilmagan foydalanuvchilar uchun Detect standart konstantasi mavjud. Agarda bu qiymat DriverVar parametri bilan o‘zlashtirilgan bo‘lsa:

DriverVar := Detect;

u holda InitGrah avtomatik ravishda kerakli drayverni inisiallab, display uchun kerakli rejimni o‘rnatadi. Uchinchi parametr RathToDriver - GRAPH moduliga marshrut o‘rnatadi. USES Srt, Grah; VAR DriverVar, ModeVar: Integer; BEGIN

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

DriverVar := Detect; InitGrah(DriverVar, ModeVar, oo); Grafika rejimni ClosseGrah protsedurasi yordamida yopiladi. Matn rejimidan grafika rejimiga o‘tish algoritmi quyidagi fragmentda ko‘rsatilgan: USESSrt, Grah; VAR DriverVar, ModeVar: Integer;

```
BEGIN
  CrIScr;
  Writeln(oMatnli rejimo);
  ...
  DriverVar:= Detect;
  InitGrah(DriverVar, ModeVar, oo);
  OutTextXY(300,250,oGrafika rejimio)
  ...
  RestoreSRTMode;
  Writeln(o 1-Matn rejimio);
  ...
  SetGrahMode(ModeVar);
  OutTextXY(300,250,o 1-Grafika rejimio)
  ... ClosseGrah
END.
```

Vidio Buferning xotirasi bir nechta bo‘limlarga bo‘linadi. Ularni videosaxifalar deyiladi. Saxifalar tartib nomeri 0 dan boshlanadi..

Jad. ViDOSaxifalar rejimi Drayver

Nomer	Rejim	Nomer	Imkoniyat	Rang	Saxifalar
EGA 3	EGALO EGAHI	0	640x200	16	4
		1	640x350	16	2
VGA 9	VGALO VGAMED	0	640x200	16	4
		1	640x350	16	4
HERC 7	HERC- MONOHI	0	720x348	2	2

Har bir alohida momentda ekranda bir saxifa tasvirlanishi mumkin va u ko‘rinuvchi deb ataladi. Berilgan momentda tasvir aks ettirilgan saxifa aktiv saxifa deyiladi. Videosaxifalar bilan ishlash uchun ikkita protsedura qo‘llaniladi: SetActiveRage va SetVisualRage. SetActiveRage(Rage: word) protsedurasi aktiv saxifani o‘rnatadi. Masalan: SetActiveRage(1); Saxifani tashkillab bo‘lgandan so‘ng, uni SetVisualRage(Rage : word) protsedurasi yordamida ekranda ko‘rsatish mumkin. Bu erda Rage – ko‘rinadigan saxifa tartib raqami

Masalan:

```
SetActiveRage(0); {0 saxifani ekranda ko‘rsatish }
OutText(o 0 - saxifao); { ekranda qator paydo bo‘ladi }
SetActiveRage(1); {1- aktiv saxifa }
OutText(oStranisa 1o); { ! - saxifada tasvirni shaqllananishi } { Lekin ekranda qator yshi ! }
Readln; SetVisualRage(1); {1 - saxifani ko‘rsatish, ekranda iator} Grafika dasturlarida ham boshqa dasturlardagi kabi xatoliklar bo‘lishi mumkin. Bunday xatoliklarni aniqlash uchun quyidagi ikkita funktsiyadan foydalaniladi: GrahRezult i GrahErrorMsg. GrahRezult: Integer - bu funktsiya, agarda xatolik bo‘lsa, 0 qiymatni aks holda -15..-1 diapozondagi sonlarni qaytaradi. Misol: USES Grah; VAR ErrorNumber: Integer;
BEGIN
ErrorNumber:= GrahRezult;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Ekranda tasvirni ko‘rish uchun koordinatalar sITEMasidan foydalaniladi. Hisob ekran koordinatasi (0,0) bo‘lgan yuqori chap burchaqdan boshlanadi..

Tasvirni ko‘rish uchun, xech bo‘lmaganda tavar chiqarish bosh nuqtasini ko‘rsatib o‘tish zarur. Matnli rejimda bu vazifani kursor bajaradi. Grafika rejimida kursor bo‘lmaydi, lekin ko‘rinmas joriy ko‘rsatgich bor CR (Current Pointer), bu xuddi o‘sha kursor, lekin u ko‘rinmaydi. Matnli rejimlarda kursorni GOTOXY protsedurasi yordamida joylashtirish siljitish mumkin, grafika rejimlarida esa Sr ni joylashtirish uchun bir nechta protsedura va funktsiyalar mavjud.

Birinchi navbatda bu MoveTo i MoveRel.

MoveTo (X,Y: Integer)

Bu protsedura joriy ko‘rsatgichni X,Y koordinatali nuqtaga joylashtiradi (siljitadi). **Masalan,** MoveTo(200,100) protsedurasi SR ni 200,100 koordinatali nuqtaga joylashtiradi .

MoveRel (dX, dY: Integer)

Protsedurasi CR ni gorizontal buo‘yicha dX nuqtaga, vertikal bo‘yicha dY nuqtaga o‘rnatadi. MoveRel(5,10) protsedurasi esa Sr ni 200,100 koordinatali nuqtadan 205,110 koordinatali nuqtaga joylashtiradi.

Joriy ko‘rsatgichni nazorat qilib borish uchun GetX va GetY lardan foydalaniladi.

Masalan:

VAR

Xros, Yros: Integer;

...

Xros := GetX;

Yros := GetY;

...

GetMaxX: Integer i GetMaxY: Integer,

X- va Y-koordinatalarini rejim uchun mos bo‘lgan maksimal qiymatlarini qaytaradi. Masalan:

X := 6000/10;

Y := 2000/2;

IF ((X > GetMaxX) OR (Y > GetMaxY)) THEN BEGIN

Write(o!- diapazonni buzilganligi!o);

MarginHadling {xatlarni tuzatish uchun foydalanuvchi protsedurasi}

END;

MoveTo(I,J);

Matnli rejilardagi kabi grafika ekranini ham bitta katta darcha yoki bir -nechta kichik darchalar deb qarash mumkin. Eslatib o‘tamiz, darcha bu ekranning to‘g‘ri burchakli sohasi bo‘lib, to‘la ekran fuksiyalarini bajaradi. Darcha o‘rnatilgandan so‘ng, ekran yoshi deb faraz qilinib, hamma kiritish chiqarish operatsiyalari shu darchada bajariladi.

Ekrandagi hamma tasvirlarni o‘chirish uchun parametrlarga ega bo‘lmagan ClearDevice protsedurasi qo‘llaniladi.

SetViewRort(x1,y1,x2,y2: Integer; Clir: boolean) protsedurasi esa , darchalarni tashkil qiladi; bu erda x1, y1 - chap tepa burchak koordinatalari, x2, y2 – so‘ng past burchak koordinatalari. Clir parametri tasvir darcha chegarasiga simagan paytda, tasvir kesilib qoladimi (Clir:= True) yoki yo‘qmi (Clir:= False) ligini aniqlaydi.

Darchani tozalash uchun ClearViewRort protsedurasi qo‘llaniladi.

Joriy darcha atributlari quyidagi protsedura yordamida o‘rnatiladi:

GetViewSettings(VAR Vr : ViewRortTyre).

Vr o‘zgaruvchi ViewRortTyre ning standart toifasiga kiradi:

TRUE

ViewRortTyre = RECORD

x1, y1, x2, y2 : Integer;

Clir : boolean

END;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Shuni yodda saqlash kerakki, matnli darchalardan farqliroq, grafik darcha fonni o‘rnatish `SetBkColor` va tozalash `ClearViewRort` buyruqlari o‘rnatilgandan so‘ng fonni ekranning umumiy foni bilan birgalikda o‘zgartiradi. Shuning uchun grafik darchani (aniqrogi "Bo‘yashni") `SetFillStyle` va `etFillRattern`. protsedurasi yordamida o‘rnatiladi. To‘g‘ri burchaqni `Bar` protsedurasi yordamida ko‘riladi

```
SetViewRort(100,50,500,200,True);  
SetFillStyle(1,3); { fon toifasni tanlash}  
Bar(100,50,500,200); { fonli to‘g‘ri burchaq}
```

Nuqta va chiziqlarani chiqarish

RutRixel, GetRixel, Line, SetColor, LineTo, LineRel, SetLineStyle.

CHiziini ekranga chiqarish rejimi nuqtani chiqarish uchun GRAPH bibliotekasida `RutRixel(X, Y: Integer; Color: word)` protsedurasidan foydalanamiz. bu erda X i Y - nuqtani ekranda joylashish koordinatasi, Color - uning rangi. Color ning mumkin bo‘lgan qiymatlari quyidagi jad. 13.6 da keltirilgan.

Jad. Rangli shkala

Rang	Kod	Rang	Kod
Black	0	DarkGray	8
Blue	1	LightBlue	9
Green	2	LightGreen	10
Cyan	3	LightCyan	11
Red	4	LightRed	12
Magenta	5	LightMagenta	13
Br‘wn	6	ELlow	14
LightGray	7	White	15

Konkret pozitsiyadagi nuqta rangini bilish uchun `GetRixel(X, Y: Integer)` fuksiyasi ishlatiladi

`Line(X1, Y1, X2, Y2: Integer)` protsedurasi chiziilar chizish uchun ishlatiladi.,

bu erda X1, Y1 - chiziqning koordinatalar boshi , X2, Y2 - koordinatlar oxiri. **Masalan:**

```
Line(1,1,600,1);
```

Ko‘rinib turibdiki, `Line` protsedurasida rang o‘rnatish parametri qui, shuning uchun bu holatlarda rangni `SetColor (Color: word)` protsedurasi yordamida o‘rnatiladi. Bu erda Color - rang, uning qiymatlari 13.6 jad.dan olinadi.

Masalan:

```
SetColor(Cyan);
```

```
Line(1,1,600,1);
```

CHiziq chizish uchun yana ikkita protsedura qo‘llaniladi: `LineTo` i `LineRel`.

`LineTo(X,Y: Integer)`

`LineRel(dX,dY: Integer)` prsedurasi ko‘rsatgichning joriy nuqtasidan $CRx + dX$, $CRy + dY$ gacha bo‘lgan chiziqni chizadi, bu erda CRx va Cry lar SR ning joriy koordinatalari.

PascalABC ABC turli stildagi ranglar chizish imkoniyatini beradi: nozik, keng, shtrixli, punktirli va xokozolar. Stilni o‘rnatish `SetLi-neStyle (LineStyle: word; Rattern: word; Thickness: word)` protsedurasi yordamida amalga oshiriladi.

Parametr `LineStyle` parametri qator toifasini o‘rnatadi. `Rattern` - namuna, `Thickness`

Agarda foydalanuvchi o‘zining stilida ishlashni xoxlasa, unda `Rattern` ning qiymati 4 ga teng. **Masalan:**

```
SetLineStyle(UserBitLn,$5555,ThickWidth);
```

```
Rectangle(20,20,120,100);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

GetLineSettings (VAR LineInfo: SetLineStyle. Toifa LineSettingsTyre tomonidan o‘rnatilgan joriy stilini, chiziq obrazi va qalinligini qaytaradi va quyidagi standartga ega.

TRUE

```
LineSettingsTyre = RECORD
  LineStyle : word;      { stil }
  Rattern   : word;      { obraz }
  Thickness : word      { ialinlik }
END;
```

Stil, obraz va qalinlik qiymatlarini aniqlash uchun LineSettings Tyre yozuviga murojat etish kerak:

```
VAR
  LineInfo : LineSettingsTyre;
...
  SetLineStyle(DOttedLn,0,NormWidth);
Rectangle(15,15,150,130);
  GetLineSettings(LineInfo);
  WITH LineInfo DO BEGIN
    Writeln(oStil:o, Linestyle);
    Writeln(oObraz:o, rattern);
    Writeln(oIalinlik:o, thickness)
  END;
```

GetTextsettings(VAR Info: TextsettingsTyre) protsedurasini chaqirib, joriy matn haqida to‘la harakteristika olish mumkin.

Bu protsedura qiymatlarni quyidagi toifa o‘zgaruvchilariga qaytaradi:

TRUE

```
TextsettingsTyre = RECORD
  Font,          { shrift kodi }
  DiRection,     { yo‘nalish kodi}
  CHarSize,     { belgi o‘lchovi}
  Horiz,        { goriz. tekislash qiymati }
  Vert: word    { vert. tekislash qiymati }
END;
```

Maydonlarga shtish:

```
VAR
  InfVar : TextsettingsTyre;
...
  GetTextsettings(InfVar);
  Writeln(oShrift:o,InfVar.Font);
  Writeln(oYunalish:o,InfVar.DiRection);
...
```

Matnlar bilan ishlash

OutText, OutTextXY, SetTextstyle, SetUserCharSize, matnni tekislash.

Ekranga chiqarilgan tasvir odatda tushintirish xati orqali kuzatiladi. Buning uchun grafika rejimida OutText va OutTextXY proseduralari ishlatiladi

OutText(Textstring : string)

Bu posedura matn satrini, SR ning joriy holatidan boshlab ekranga chiqaradi. Masalan:

OutText(oBerilganlarni kiriting:o);

OutTextXY(X, Y: Integer; Textstring: string)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

bu erda X, Y - matnni kiritish boshlang‘ich nuqta koordinatasi, Textstring - string toifasidagi konstanta yoki o‘zgaruvchi. Masalan, 60,100 nuqtalardan boshlab, "Davom ettirish uchun xoxlagan tugmachani bosing", degan ma`lumotni ekranga chiqarish uchun, quyidagini yozish kerak:

```
OutTextXY(60,100,o Davom ettirish uchun xoxlagan tugmachani bosing ..o);
```

PascalABC dagi shriftlar ro‘yxati 13.9 jad. da keltirilgan. Kerakli shriftni SetTextstyle(Font : word; DiRection : word; CHarSize: word) protsedurasi yordamida o‘rnatiladi: bu erda Font - tanlangan shrift, DiRection – yo‘nalish (gorizontal yoki vertikal), CHarSize - chiqarilayotgan belgilar o‘lchovi. Ikkita birinchi parametrlar qiymati 13.9, 13.10 jadisalarida keltirilgan. Vertikal chiqarishni tashkil qilishda shuni nazarda tutish kerakki, agarda foydalanuvchi MoveTo yordamida boshlanich nuqta koordinatasini o‘rnatib olmasa, u holda matn ekranning pastki qatoridan boshlab tepaga davom etadi Chiqarilayotgan belgilar kattaligi CharSize koefitsienti yordamida o‘rnatiladi. Agarda i CHarSize = 1 bo‘lsa, u holda belgi 8x8 matrisasida ko‘riladi, agarda CHarSize = 2 bo‘lsa, u holda 16x16 matrisasi va hokazolarda ko‘riladi.

Misol tariqasida vertikal va gorizontal qatorarni DefaultFont shrift bilan chiqarishni ko‘rib chiqamiz:

```
SetTextstyle(0, 1, 1);      { standart o‘lchovdagi harflar }
Outtextxy(200, 200, oVertikal qatoro);
SetTextstyle(0, 0, 2);     { oarflar kattalashtirilgan }
Outtextxy(200, 220, oGorizontal qatoro);
```

Kerakli o‘lchovni SetUserCharSize(multX, divX, multY, divY: word) protsedurasi yordamida ham o‘rnatish mumkin.

SetUserCharSize chiqarilgunga qadar yoki undan keyin SetTextstyle yordamida shriftni va uni yo‘nalishini o‘rnatish zarur.

Masalan:

```
SetTextstyle(SansSerifFont,HorizDir,4);
OutText(oNormal o‘lchovo);
SetUserCharSize(1,3,1,1);
OutText(oBalANDligi bo‘yicha kamaytirsh);
SetUserCharSize(3,1,1,1);
OutText(oKenligi bo‘yicha katalashtirish);
```

Qator yoki belgining piksellardagi gorizontal va vertikal o‘lchovlarini bilish uchun

TextHeight(Textstring: string): word,

TextWidth(Textstring: string): word funksiyalaridan foydalanamiz.

Misol tariiasida matn atrofida ramka chizish dasturini keltiramiz.

Ramka ochii xavo rangda, matn atrofi oi rang bilan chiziladi: PROGRAM DemoTextFrame;

```
USES Srt, Grah;
```

```
VAR
```

```
Driver, Mode: Integer;
```

```
St: string;
```

```
Height, Width, cX, cY, x1, x2, y1, y2: Integer;
```

```
BEGIN
```

```
St := oRamkadagi matno;
```

```
Driver:=detect;
```

```
InitGrah(Driver,Mode,oo);
```

```
cX:=GetMaxX div 2;
```

```
cY:=GetMaxY div 2;
```

```
SetTextJustify(CenterText,CenterText);
```

```
SetTextstyle(DefaultFont,HorizDir,2);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Height:=TextHeight(St) + 4;  
Width:=TextWidth(St) + 4;  
x1:=cX - (width div 2);  
x2:=cX + (width div 2);  
y1:=cY - (Height div 2);  
y2:=cY + (Height div 2);  
SetColor(White);  
Rectangle(x1,y1,x2,y2);  
SetColor(LightCyan);  
OutTextXY(cX,cY,St);  
Readln;  
CloseGrah  
END.
```

Matnlarni tekkislash Horiz i Vert porametrleri orqali berilgan SetTextJustify(Horiz, Vert : word) protsedurasi yordamida bajariladi.

Rang va palitrani o‘rnatish

SetRalette, SetAllRalette, GetBkColor, GetColor, GetMaxColor, GetRalette, GetRaletteSize
EGA/VGA-adapterlari uchun EGAVGA.BGI drayveri 54 rang o‘rnatadi. Fon SetBkColor yordamida beriladi.

SetRalette va SetAllRalette yordamida foydalanuvchi ranglarning o‘rnashish ketma-ketligini o‘zgartirishi mumkin (13.15 va 13.16 jad.ga qarang) .

InitGrah ga standart berilgan ANDan so‘ng munosabatlar o‘rnatilishi haqidagi axborot RaletteTyre toifaidagi o‘zgaruvchida bo‘ladi:

```
TRUE  
RaletteTyre = RECORD  
  Size : byte;  
  Colors : ARRAY[0..MaxColors] OF shortint  
END;
```

Bu erda Size - palitradagi ranglar soni Colors - registrlardagi palitralar qiymati

Colors - massivning elementlari butun sonlar, ular konkret ranglarni aniqlaydi.

Masalan 4 indeksli massivning rangi Red, 14 - element EILow rangiga to‘g‘ri keladi va xoqazo.

Joriy palitra haqidagi axborotni GetDefaultRalette(VAR Ralette : RaletteTyre) protsedura yordamida olish mumkin.

GetDefaultRalette(VAR Ralette : RaletteTyre)

bu erda Ralette - RaletteTyre toifasidagi o‘zgaruvchi.

Turb’-PascalABC tilida ikkita muxim atributni o‘rnatish uchun SetColor protsedurasi qo‘llaniladi.

SetColor(Color : word) protsedura rang Color parametri bilan berilgan grafika protsedurasining rangini o‘rnatadi.

Bir va bir nechta ranglarni o‘zgartirish uchun SetRalette prose-durasida foydalanish mumkin, agarda

butun palitrani o‘zgartirish talab qilinsa SetAllRalette protsedurasidan foydalanamiz.

SetRalette (ColorNum : word; Color : shortint) protsedura rang va palitra tartib nomerlari orasidagi moslikni o‘zgartirishda qo‘llaniladi. ColorNum palitradagi rang nomeri, Color rangning yangi qiymati.

Ettinchi tartib nomer boshqa rangni berishi uchun, masalan EILow ni berishi uchun quyidagicha yozish kerak: SetRalette(7,EILow); Bundan so‘ng ekran rangi tezda o‘zgaradi. Protседura EGA- va VGA-adapterli SITEMalar uchungina ishlaydi.

SetAllRalette (VAR Ralette: RaletteTyre)

EGA- va VGA-adapterli SITEMalarda palitradagi ranglar joylashuvi tartibini

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

O‘zgartirish uchun ishlatiladi.

Misol:

```
TRUE
RaletteTyre = RECORD
Size : byte;
Colors : ARRAY [0..MaxColors] OF shortint
END;
VAR
Ralette: RaletteTyre;
...
Ralette.Colors[0]:=Br'wn;
...
```

Palitralar ishlashga misol sifatida quyidagi masalani ko‘rib chiqamiz. Berilgan rangdagi fonda to‘g‘ri burchaqni chizib, so‘ngra 15- tartib nomerli White politrasi Green. bilan almashtirish kerak, bo‘lsin. So‘ngara o‘z politrasi tashkil etib, berilgan fondagi to‘g‘ri turtburchaqni ekranga chiqarish dasturi quyida ko‘rsatilgan:

```
PROGRAM Dem'Ralettes;
USES Srt, Grah;
VAR
DriverVar, ModeVar: Integer;
Ralette : RaletteTyre;
BEGIN
DriverVar:=detect;
InitGrah(DriverVar,ModeVar,oo);
GetRalette(ralette);
Rectangle(10,10,100,200); { White rangli to‘g‘ri burchaq }
Delay(3000);
{ 15 (White) ni Green bilan almashtirish }
SetRalette(15,Green); {to‘g‘ri burchaqni boshqaltdan bo‘yash }
{ oq rangdan yashil rangga }
Delay(4000);
{ 15 tarangdan iborat o‘z palitramizni o‘rnatamiz }
Ralette.Colors[0] := Br'wn;
Ralette.Colors[1] := Red;
Ralette.Colors[2] := Green;
Ralette.Colors[3] := LightBlue;
...
Ralette.Colors[15] := LightRed;
SetAllRalette(ralette);
{ Black rang foni Br'wn ga o‘zgaradi }
{ birinchi to‘g‘ri burchak rangini LightRed ga o‘zgartiradi }
{ Yangi to‘g‘ri burchak oddiy holatda oxirgi rang bilan chiziladi }
{ foydalanuvchining palitrasi bilan - LightRed }
Rectangle(120,100,600,300);
Readln;
CloseGrah
END.
Sanab o‘tilganlardan tashqari yana qo‘shimcha vositalar mavjud: GetBkColor, GetColor,
GetMaxColor,
GetRalette, GetRaletteSize. Ularning vazifalari bilan qisqacha tanishtirib o‘tamiz:
GetBkColor: word - fon rangining joriy tartib nomerini qaytaradi
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

GetColor: word - chizish uchun joriy rang tartib nomerini qaytaradi

GetMaxColor: word - minus 1 palitradagi rang kodining maksimal qiymatni qaytaradi.

GetRalette (VAR Ralette: RaletteTyre) - joriy politra haqidagi axborotni RaletteTyre toifadagi o‘zgaruvchiga chiqaradi. U etDefaultRalette dan farqliroq foydalanuvchi palitralaridagi berilganlarni qaytaradi.

GetRaletteSize: Integer - joriy palitralardagi ranglar sonini qaytaradi.

Standart grafik proseduralar

Graph modulining protsedura va funksiyalari

1. **Arc** protsedurasi - aylana yoyini chizadi.

Aniqlanishi : Arc(x,y : integer; StAng, EndAng, Radius: Word);
x,y - aylana markazining koordinatasi;

StAng, EndAng - mos ravishda yoyning boshlang‘ich va oxirgi burchaklari;
Radius-aylana radiusi.

2. **Bar** protsedurasi - rangga bo‘yalgan to‘g‘ri to‘rtburchak chizadi.

Aniqlanishi: Bar(x1,y1,x2,y2:integer);

(x1,u1) va (x2,u2) mos ravishda to‘g‘ri to‘rtburchakning chetki nuqtalari koordinatalari.

3. **Bar3D** protsedurasi rangga bo‘yalgan parallelipiped chizadi.

Aniqlanishi :Bar3D(x1,y1,x2,y2:integer;Depth:word;Top:boolean);

(x1,y1) va (x2,y2) asosni tashkil etuvchi to‘g‘ri to‘rtburchak uchlarining koordinatalari;
Depth -parallelipiped chuqurligi;

Top- mantiqiy o‘zgaruvchi.

4. **Circle** protsedurasi - aylana chizadi;

Aniqlanishi: Circle(x,y:integer;Radius:word);

(x,y) aylana markazining koordinatasi;

Radius-aylana radiusi.

5. **CloseGraph** protsedurasi grafik rejimini uzadi.

Aniqlanishi :Closegraph;(parametrsiz protsedura)

6. **DrawPoly** protsedurasi - ko‘p burchak chizadi.

Aniqlanishi :DrawPoly(NumPoints:word; var PolyPoints);

NumPoints - ko‘pburchak tomonlari soni;

PolyPoints - ko‘pburchak uchlarining koordinatalaridan tuzilgan massiv.

7. **Ellipse** protsedurasi - ellips yoyini chizadi.

Aniqlanishi: Ellipse(x,y:integer;StAng,EndAng:word;XRradius,YRradius:word);

(x,y) – ellips markazning koordinatasi;

StAng va EndAng - yoyning boshlang‘ich va oxirgi burchaklari;

Xradius va Yradius mos ravishda ellips balandligi va eni.

8. **FillPoly** protsedurasi - rangli ko‘pburchak chizadi.

Aniqlanishi: FillPoly(NumPoints:word; var PolyPoints);

NumPoints - ko‘pburchakning uchlari soni;

PolyPoints - ko‘pburchak uchlari koordinatalaridan tuzilgan massiv.

9. **GetArcCoords** protsedurasi - oxirgi marta ishlatilgan **Arc** protsedurasining koordinatalarini aniqlaydi.

Aniqlanishi: GetArcCoords(var ArcCoords:ArcCoords Type);

10. **GetColor** funksiyasi - ekran rangini aniqlaydi.

Aniqlanishi: GetColor:word;

11. **GetGraphMode** funksiyasi - grafik ekranni qaytaradi.

Aniqlanishi: GetGraphMode:integer;

12. **GetImage** protsedurasi - ekranning berilgan sohasini Area da saqlaydi.

Aniqlanishi: GetImage(x1,y1,x2,y2:integer;var Area);

13. **GetMaxColor** funksiyasi - rangning eng katta qiymatini hisoblaydi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Aniqlanishi: GetMaxColor:word;

14. **GetPixel** funksiyasi - berilgan nuqta rangini aniqlaydi.

Aniqlanishi: GetPixel(x,y:integer):word;

15. **GraphErrorMsg** funksiyasi - berilgan kod bo‘yicha xato haqida satr ma’lumot beradi.

Aniqlanishi: GraphErrorMsg(Code:integer):string;

16. **LineTo** protsedurasi - oldingi aniqlangan nuqtadan berilgan nuqtagacha kesma chizadi.

Aniqlanishi: LineTo(x,y:integer);

17. **PieSlice** protsedurasi sektor chizadi.

Aniqlanishi: PieSlice(x,y:integer;StAng,EndAng,Raduis:word);

Savol va topshiriqlar.

1. Paskalda Graph moduli qanday maqsadda qo‘llaniladi?
2. Grafikli drayver nima va u qanday vaifani bajaradi?
3. Ekranni grafik holatga o‘tkazish uchun Pascalda qanday ko‘rsatma beriladi?
4. Grafik xolatdan chiqish uchun qaysi operatoridan foydalaniladi?
5. Ekranni grafik holatga o‘tkazilganda yurgichning ko‘rinishi qanday bo‘ladi?
6. Grafik xolatda ekranning qaysi nuqtasi joriy nuqta bo‘ladi?
7. GetMaxX va GetMaxY funksiyalarini vazifalarini aytib bering.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

FOYDALANILGAN ADABIYOTLAR

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. Introduction to algorithms. Massachusetts Institute of Technology. London 2009.
2. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Сер: Классические учебники: COMPUTER SCIENCE. М.: МЦНМО, –960с., 2004.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2010 y.
4. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
5. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyo”, 2011y.
6. T. X. Holmatov, N. I. Toyloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta’minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
7. М. Ашуров, М. Мирзахмедов, Ш. Сапаев. Замонавий дастурлаш тиллари фанидан лаборатория ишлари. Т. : ТДПУ, 2008 й.
8. M.J.Yunusova, A.B.Rahimov. Dasturlash bo‘yicha praktikum. T.: “Ilm ziyo”, 2006 y.
9. А.В.Файсман. Профессиональное программирование на Турбо – Паскале. Информ Экс - Корпорейшн, 1992 г.
10. B.Boltayev, M.Mahkamov, A.Azamatov. Paskal dasturlash tili. T.; 2007 y.
11. А.Н.Марченко. Программирование в среде *Turbo Pascal 7.0*. К., Век+, М., «ДЕСС», 1999 y.
12. Karimov R. va boshqalar. Dasturlash. T. “O‘zbekiston”. 2003 y. 206 b.
13. Матрос Д.Ш., Поднебесова Г.Б. Теория алгоритм. Учебник для педагогического образования. М.: Бином. Лаборатория знаний, - 2008. -202с.
14. Макконелл Дж. Основы современных алгоритмов. 2-доп.изд., М.: ТЕХНОСФЕРА, 366с., 2004.
15. Вирт Н. Алгоритмы и структуры данных. С примерами на Паскале. Санкт-Петербург, 352с., 2005.
16. Рейест Р. и др. Алгоритмы: построение и анализ. М., Мир, 1994.
17. Малышко В.В. Алгоритмы и алгоритмические языки. Конспект лекций для студентов Ташкентского филиала МГУ, 68с., 2006.
18. Пильшиков В.Н., Абрамов В.Г., Вылиток А.А., Горячая И.В. Машина Тьюринга и алгоритмы Маркова. Решение задач. (Учебно-методическое пособие) Московский государственный университет им. М.В.Ломоносова, Факультет вычислительной математики и кибернетики, Москва, 47с., 2006.
19. Yunusov A.S. Matematik mantiq va algoritmlar nazariyasi elementlari. Samarqand davlat universiteti nashriyoti, 182 b, 2012y.
20. Слинкин Д.А. Основы программирования на Турбо-Паскале: Учебно-методическое пособие для студентов вузов. Шадринск: Изд-во Шадринского пединститута, 2003. – 134-136 сс.
21. Фаронов В. В. Turbo Pascal. — СПб.: ВХВ- Санкт-Петербург, 2004. – 1056 с. (301-320сс)
22. М.У.Ашуров, Н.Д.Мирзахмедова. Turbo Pascal дастурлаш тили.(услубий кўлланма), Тошкент ТДПУ – 2011 (81-87)
23. A.U.Ashurov, N.D.Mirzahmedova, N.S.Haytullayeva. Algoritmash va dasturlash tillari. Informatika o‘qitish metodikasi ta’lim yo‘nalishi uchun uslubiy qo‘llanma. Toshkent – 2015 (113-115 B)
24. www.ziyonet.uz.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA‘LIM
VAZIRLIGI
NAVOIY DAVLAT PEDAGOGIKA INSTITUTI
«INFORMATIKA O‘QITISH METODIKASI» KAFEDRASI**

ALGORITMLAR FANIDAN

AMALIY MASHG'ULOTLAR TO'PLAMI

Navoiy-2019

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Ushbu amaliy mashg‘ulotlar to‘plami “Informatika o‘qitish metodikasi” yo‘nalishidagi bakalavrlar uchun mo‘ljallangan “Algoritmlar” fani bo‘yicha fan dastur asosida tuzilgan bo‘lib, bo‘lajak mutaxassis egallashi kerak bo‘lgan bilim va ko‘nikmalarni o‘z ichiga oladi.

Tuzuvchi: dots.A.A.Ibragimov
k.o‘q.D.N.Xamroyeva
o‘q. S.Q.Absobirov

Taqrizchi: dots.Ro‘ziyev R.A.

Amaliy mashg‘ulotlar to‘plami Navoiy davlat pedagogika instituti “Informatika o‘qitish metodikasi” kafedrası yig‘ilishida ko‘rib chiqilgan va tavsiya qilingan. 2019-yil ____-avgustdagi 1-sonli yig‘ilish bayonnomasi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

1-AMALIY MASHG‘ULOT

MAVZU: Algoritm tushunchasi va ulardan foydalanish.

Mashg‘ulotning maqsadi: Masalani EHMda yechishning algoritmlash bosqichlari. Algoritmning asosiy xossalari. Algoritmni tavsiflash usullari. Dasturlash tillari va ularni sinflash. Algoritmning asosiy turlarini o‘rganish hamda bilim, ko‘nikma va malakalar hosil qilishdan iborat.

Qisqacha nazariy ma‘lumot

Algoritm tushunchasi.

Qo‘yilgan biror masalani EHMda yechish uchun, avval uning matematik modelini, keyin algoritmini va dasturini tuzish kerak bo‘ladi. Bu uchlikda algoritm bloki muhim ahamiyatga ega. Algoritm bu oldimizga qo‘yilgan masalani yechish zarur bo‘lgan amallar ketma-ketligidir.

ALGORITMNING ASOSIY XOSSALARI.

Algoritm quyidagi asosiy xossalarga ega: uzluksizlik, aniqlik, natijaviylik va ommaviylik.

UZLUKLILIK. Dastlabki berilgan ma'lumotlarni natijaga aylantirish jarayoni uzluksiz ravishda amalga oshiriladiki, bunda vaqtning har bir keyingi keladigan dasisasidagi misdor (kattalik)larning siymati vaqtning shundan oldingi dasisasida bo‘lgan misdorlar siymatidan ma'lum bir qoidalar buyicha olinadi.

ANIQLIK. Algoritmning har bir qoidasi aniqva qiymatli bo‘lishi zarurki, bunda vaqtning biror dasisasida olingan misdorlar siymati vaqtning shundan oldingi dasisasida olingan misdorlar siymati bilan bir siymatli anislangan bo‘ladi.

NATIJAVIYLIK. Algoritm masalaning yechimiga chekli sondagi sadamlar ichida olib kelishi yoki masalani «yechib bo‘lmaydi» degan xabar bilan tugashi kerak.

OMMAVIYLIK. Masalaning yechish algoritmi shunday yaratilishi kerakki, uni fasat boshlang‘ich ma'lumotlar bilan farqlanadigan masalalarni yechish uchun ham so‘llanishi kerak. Bunda boshlang‘ich ma'lumotlar algoritmini so‘llash sohasi deb ataladigan birorta sohadan olinadi. Masalan, yuqoridagi 1 – misolda koptok o‘rniga boshqa narsani tik irhitilsa va uning boshlang‘ich tezligi ma'lum bo‘lsa, shu algoritm bilan u erishadigan balandlik anislangan.

ALGORITMNI TAVSIFLASH USULLARI VA ULARGA MISOLLAR.

Algoritmni ishlab chisishda uni bir necha xil usul bilan ifodalab bersa bo‘ladi. Shulardan uchtasi keng tarsalgan. Bo‘lar:

1. Algoritmni oddiy tilda tavsiflash;
2. Algoritmni tuzim ko‘rinishda ifodalash;
3. Algoritmni maxsus (algoritmik) tilda yozish.

Algoritmni oddiy tilda tavsiflash.

Algoritmni ifodalanshini eng keng tarsalgan shakli, bu oddiy tilda so‘zlar bilan bayen silishdir. Bu nafasat hisoblash algoritmlarida, balki hayotiy, turmushdagi «algoritm»ga ham tegishlidir.








Algoritmning jadval yordamida ifodalanishi

Algoritmning bu ko‘rinishda berilishi ham sizga tanish. Masalan, matematikada qo‘llanib kelinayotgan Bradis jadvali deb nomlangan to‘rt honali matematik jadval, lotareya yutuqlar jadvali, Mendeleyev kimyoviy elementlar jadvali. Bunday jadvallardan foydalanish ma‘lum bir algoritm qo‘llashni talab etadi. Biror funksiyaning grafigini chizish uchun ham funksiyaning argument qiymatlariga mos qiymatlar jadvalini hosil qilamiz. Bu ham algoritmning jadval ko‘rinishiga misol bo‘ladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Algoritmning grafik shaklda ifodalanishi

Algoritmning bu ko‘rinishda ifodalanishi matematikada chizilgan grafik, kerakli uyni oson topish uchun dahalarda o‘rnatilgan uylarning joylashish sxemasi, avtobuslarning yo‘nalish sxemasi orqali sizga tanish. Algoritmash asoslarini o‘rganishning yana bir qulay grafik shakli – blok-sxema usulidir. Blok-sxemalar bir yoki bir nechta buyruq yoki ko‘rsatmani aks ettiruvchi maxsus geometrik shakllar –bloklardan tashkil topadi. Bloklar yo‘nalish chiziqlari orqali tutashiriladi.

	algoritmning boshlanishini va tugallanganligini bildiradi
	ma'lumotlarni kiritish yoki chiqarishni bildiradi
	oddiy harakatni, ya'ni qiymat berish yoki tegishli ko'rsatmalar berishni bildiradi
	shart tekshirilishini bildiradi
	yordamchi algoritimga murojaatni bildiradi
	blok-sxemadagi harakat yo'nalishini bildiradi
	qiymat berish ko'rsatmasi

Informatikada masala yechish tushunchasi deganda axborotlarni qayta ishlab, natijani oldindan belgilangan ma'lum bir ko‘rinishga olib kelish tushuniladi.

EHMdan foydalanib masalani yechish – yaratilgan algoritimga asoslangan holda dastlabki ma'lumotlar ustida avtomatik tarzda amallar bajarilib izlangan natija (natijalar) ko‘rinishiga keltirish demakdir.

Masalalarni EHMda yechish bosqichlari.

EHMdan foydalanib, «ilmiy-texnik masalani yechish» tushunchasi keng ma'nodagi so‘z bo‘lib, quyidagi bosqichlarga bo‘linadi. Maqsadimiz bosqichlarni qaysi birlarini mutaxassis EHMdan foydalanmasdan va qaysi birlarini EHMdan foydalanib bajarishini aniqlash, hamda bosqichlarni to‘la o‘rganib chiqishdan iborat.

Ilmiy-texnik masalalarni EHMdan foydalanib yechish bosqichlari:

1. Masalaning qo‘yilishi va maqsadning aniqlanishi;
2. Masalani matematik ifodalash;
3. Masalani yechish uslubini ishlab chiqish, sonli usullarni tanlash;
4. Masalani yechish algoritmini ishlab chiqish;
5. Ma'lumotlarni tayyorlash va tarkibini aniqlash(tanlash);
6. Dasturlash;
7. Dastur matnini va ma'lumotlarni axborot tashuvchiga o‘tkazish;
8. Dastur xatolarini tuzatish;
9. Dasturni avtomatik tarzda EHMda bajarilishi;
10. Olingan natijalarni izohlash, tahlil qilish va dasturdan foydalanish uchun ko‘rsatma yozish.

Algoritmni tasvirlash usullariga misollar keltirib o‘tamiz:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Masalan, to‘g‘ri to‘rtburchakning tomonlariga ko‘ra uning perimetri, diagonal va yuzasini hisoblash.

5. So‘z bilan ifodalash:

5.1. boshlash;

5.2. tomonlar qiymatini kiritish (a, b);

5.3. perimetr qiymatini hisoblash (p);

5.4. diagonal qiymatini hisoblash (d);

5.5. yuzasini hisoblash (s);

5.6. perimetr, diagonal va yuzasini qiymatini chop

etish.

6. Formulalarda berish:

6.1. A va B to‘rtburchak tomonlari qiymatlari;

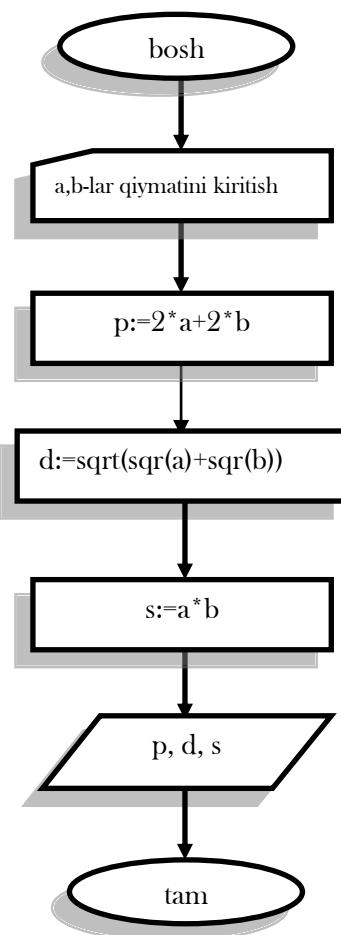
6.2. $P=2*a+2*b$;

6.3. $D = \sqrt{a^2 + b^2}$;

6.4. $S=a*b$;

6.5. P, D va S qiymatlarini chop etish

7. Blok-sxemalarda tasvirlash:



Amaliy mashg‘ulotni bajarish uchun topshiriqlar

1. Ikkita son berilgan. Bu sonlarning kublarini o‘rta arifmetigini va bu sonlarning modullari o‘rta geometrigini toping.
2. (X1,U1) va (X2,U2) koordinatalarga ega bo‘lgan ikkita nuqta orasidagi masofani toping.
3. Uchburchak uchlari (x1,u1),(x2,u2),(x3,u3) koordinatalar bilan berilgan. Shu uchburchak yuzasini va perimetrini toping.
4. a,b-sonlar berilgan. Ikkita tomoni a va b ga,ular orasidagi burchak esa α ga teng bo‘lgan uchburchak yuzasini toping.
5. a,b,c-haqiqiy sonlar berilgan.Bu sonlarning o‘rta arifmetik qiymati, summasi va ko‘paytmasini chop etuvchi dastur tuzing.
6. To‘g‘riburchakli uchburchakning katetlari berilgan bo‘lsa, gipotenuzasi va yuzasini hisoblovchi dastur tuzing.

O‘z-o‘zini tekshirish savollari:

1. Algoritmning asosiy xossalari nimalardan iborat?
2. Algoritm tuzishda asosiy xossalarni inobatga olish shartmi?
3. Algoritmni ishlab chisishda uni bir necha xil usul bilan ifodalab bersa bo‘ladi.
4. Masalani EHMda yechishni necha bosqichi mavjud?
5. Dasturlash qaysi bosqich va uning vazifasi?
6. Ilmiy-texnik masala nima?
7. Ilmiy-texnik masalalarni EHMdan foydalanib yechish bosqichlari bir-biri bilan o‘zaro bog‘langanmi? Agar bog‘langan bo‘lsa uni biror bir misol keltirish orqali ifodalab bering.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

2-AMALIY MASHG‘ULOT

MAVZU: ALGORITMLAR SAMARADORLIGINI BAHOLASH.

Mashg‘ulotning maqsadi: Amaliy masalalarga algoritmlar tuzish bo‘yicha yetarli ko‘nikma hosil qilish.

NAZARIY QISM

Algoritm tuzish jarayonida nazariy va amaliy nuqtai nazardan algoritmlash, dasturlash va EHM larni qo‘llash bilan bog‘liq bo‘lgan bilimlar kerak. Asosiy maqsad bu masalani qo‘yish, masalaning yechish algoritmini tuzish, algoritmi mashina dasturi ko‘rinishida amalga oshirish va algoritmni samaradorligini ko‘rsatish muammolarini o‘rganish. Bu jarayonlar algoritmni to‘liq yaratish tushunchasiga olib keladi va quyidagi bosqichlarni belgilaydi:

1. Masalaning qo‘yilishi.
2. Modelni yaratish.
3. Algoritmni ishlab chiqish.
4. Algoritm to‘g‘riligini tekshirish.
5. Algoritmni amalga oshirish.
6. Algoritmni va ularning murakkabligini tahlil qilish.
7. Dasturni tekshirish.
8. Hujjatlashtirish.

Masala qo‘yilishi

Masalani yechishdan oldin, uni berilishini aniq shakllantirib olish zarur. Bu jarayon to‘g‘ri savollarni aniqlash bo‘lib, savollar quyidagicha bo‘lishi mumkin:

1. Dastlabki berilgan masala shartlarida hamma iboralar tushunarlimi?
2. Nima berilgan?
3. Nimani topish kerak?
4. Yechimni qanday ta‘riflash kerak?
5. Qaysi berilganlar yetarli emas va hammasi kerakmi?
6. Qanaqa mumkinliklar qabul qilingan?

Albatta, bulardan tashqari boshqa savollarni ham ishlatish mumkin, yoki ayrim savollarni bir necha bor takror ishlatishga to‘g‘ri keladi.

Modelni yaratish

Akademik A. N. Tixonov fikri bo‘yicha matematik modellashtirish dunyoni bilish va o‘rganishda kuchli qurollardan (vositalardan) biridir.

Uning ta‘rifi bo‘yicha matematik model tashqi dunyoning xodisalar turkumini matematik belgilar yordamida taxminiy tavsifi.

Xodisani tavsiflash uchun uning muhim xususiyatlarini, qonuniyliklarini, ichki aloqalarini, ayrim xossalarning ahamiyatini aniqlash zarur. Eng muhim faktorlari aniqlanganda, ahamiyatlari kamroq bo‘lganlarini hisobdan chiqarish mumkin. Umuman, modelni tanlash fandan ko‘ra, ko‘proq san‘at ishi deb hisoblanadi, yahshi tuzilgan modellarni o‘rganish esa – modellashtirishda tajriba orttirishning eng yahshi usuli. Modelni yaratishda quyidagi savollarni aniqlash maqsadga muvofiq:

1. Masalani yechish uchun qaysi matematik struktura ko‘proq mos keladi?
2. O‘xshash masalaning yechimi bormi?
3. Masalaning barcha muhim ma‘lumotlari matematik ob‘yektlar orqali tavsiflanadimi?
4. Izlanayotgan natija biron bir matematik o‘lchamga mos keladimi?
5. Modelning ob‘yektlari orasidagi bog‘lanishlar aniqlanganmi?
6. Tuzilgan model bilan ishlash qulaymi?

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

Algoritmni ishlab chiqish

Algoritmshartirish jarayoni uslublari bo‘yicha matematik modellarni tuzish jarayoniga juda yaqin. Har bir algoritmni ishlab chiqish bevosita o‘ziga xos yondashishni talab qilishiga qaramasdan, bu faoliyatni umumiy uslub va bosqichlari ham mavjud. Ba‘zan dasturlarni tezroq yozib boshlashga hohish paydo bo‘ladi. Lekin bu xatoli, chunki aynan algoritmni ishlab chiqish bosqichiga va uning to‘g‘riligiga masalaning to‘liq yechimi bog‘liqdir. Algoritmshartirish turli xil uslublari mavjud.

Algoritmni to‘g‘riligini tekshirish

Dastur to‘g‘riligini isbotlashning eng keng tarqalgan turi – bu uni testlardan o‘tkazishdir.

Algoritmni tekshirishda nazoratchi boshlang‘ich ma‘lumotlarni majmui algoritmik test deb nomlanadi.

To‘g‘ri deb shunday algoritmga aytiladiki, u masalaning qo‘yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang‘ich ma‘lumotlar bilan ham shakllantirib biladi. Odatda, dastur bergan natijalar ma‘lum bo‘lgan yoki qo‘lda hisoblangan ma‘lumotlar bilan taqqoslanadi, va ular to‘g‘riligi aniqlansa dastur to‘g‘ri ishlaydi degan hulosaga kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo‘lmaydi, ya‘ni dastur ishlaymaydigan hamma holatlarni hisobga olib bo‘lmaydi.

Gudman va Xidetniyemi [2] lar tomonidan algoritm to‘g‘riligini isbotlash uchun quyidagi uslubiyat taklif qilingan.

Algoritm 0 dan m gacha bo‘lgan qadamlar ketma-ketligi ko‘rinishida tavsiflangan deb tahmin qilaylik. Har bir qadam uchun qandaydir asoslanishni taklif etamiz. Xususan, qadamdan oldin va keyin ishlaydigan shartlar haqida lemma kerak bo‘lishi mumkin. Shu bilan birgalikda, algoritm chekliligining isbotini ham taklif etamiz, va hamma ruxsat etilgan kiritish ma‘lumotlarini tekshirib, hamma mumkin bo‘lgan chiqarish ma‘lumotlarni olamiz. Algoritmni to‘g‘riligi bilan samaradorligi o‘rtasida hech qanday aloqa yo‘qligini ta‘kidlab o‘tamiz. Aslida hamma talablarga bir xil yahshi javob beradigan algoritm kamdan-kam ishlab chiqiladi.

Algoritmni amalga oshirish

Algoritmni amalga oshirish deganda, EHM uchun dasturni yozish deb tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

1. Asosiy o‘zgaruvchilarni aniqlash.
2. O‘zgaruvchilarning turlarini aniqlash.
3. Nechta massiv yoki fayllar va qanday kattalikda ular kerak bo‘ladi?
4. Bog‘lanilgan ro‘yhatlardan foydalanish ma‘nolimi?
5. Qanday dasturiy qismlar kerak bo‘lishi mumkin (tayyor bo‘lsa ham)?
6. Qaysi dasturlash tilini tanlash?

Dastur yozish yoki tuzishning hilma-hil usullari va uslublari mavjud.

Algoritmni va uning murakkabligini tahlil qilish

Algoritmni tahlil qilishdan maqsad – algoritmga ma‘lumotlarni aniq muvaffaqiyatli qayta ishlash uchun kerak bo‘ladigan xotira hajmi va ishlash vaqtining baholari va chegaralarini olish. Bir masalani yechadigan ikki algoritmni taqqoslash uchun qandaydirsonli mezon topish kerak.

Faraz qilaylik, A – qandaydir bir turkumdagi masalalarni yechadigan algoritm, n – esa shu turkumdagi alohida bir masalaning kattaligi. Umumiy holda, n – oddiy skalyar yoki massiv yoki kiritiladigan ketma – ketlikning uzunligi bo‘lishi mumkin. $f_A(n)$ - n kattalikdagi ixtiyoriy masalani yechadigan algoritm A bajarish kerak bo‘lgan asosiy amallarni (qo‘shish, ayirish, taqqoslash,...) yuqori chegarasini beradigan ishchi funksiya. Algoritmning sifatini baholash uchun quyidagi mezonni ishlatamiz.

Agar $f_A(n)$ o‘sish tartibi n dan bog‘liq bo‘lgan polinomdan katta bo‘lmasa, A algoritm polinomial deb aytiladi, aks holda algoritm A eksponensial hisoblanadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Shular bilan birgalikda tahlil jarayonida ko‘p matematik fanlarda standart bo‘lgan iboralar ishlatiladi.

$f_A(n)$ funksiya $O[g(n)]$ deb belgilanadi, va $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = const \neq 0$ bo‘lganda, uni tartibi katta n lar uchun $g(n)$ deb qabul qilinadi. Demak $f(n)=O[g(n)]$.

$f_A(n)$ funksiyasi $o[z(n)]$ deb katta n lar uchun belgilanadi, va unda $\lim_{n \rightarrow \infty} \frac{h(n)}{z(n)} = 0$ sharti bajariladi.

Bu begilar “katta O” va “kichik o” deb nomlanadi. Agar $f(n)=O[g(n)]$ bo‘lsa, ikkala funksiya ham $n \rightarrow \infty$ bo‘lganda bir xil tezlikda o‘sadi.

Agar $f(n)=O[g(n)]$ bo‘lsa, unda $g(n)$, $f(n)$ nisbatan ancha tez o‘sadi.

Demak, $P_k(n)$ - qandaydir n o‘zgaruvchidan bog‘liq va k darajadagi polinom uchun $f_A(n) = O[P_k(n)]$ yoki $f_A(n) = oP_k(n)$ bo‘lganda algoritm polynomial hisoblanadi, aks holda algoritm eksponensial.

Eksponensial algoritm yahshi ishlaydigan deb hisoblanadi. Agar algoritmlar eksponensial bo‘lsa, ular orasida eng samaralisini topish kerak, n kattalikdagi masalani $O(2^n)$ qadamda yechadigan algoritm $O(n!)$ yoki $O(n^n)$ qadamda masalani yechadigan algoritmdan afzalroq.

Dasturni tekshirish

Biz dasturni har bir qismini tekshiradigan kirituvchi ma’lumotlar to‘plamini tanlashimiz kerak. Ko‘p murakkab algoritmlarni matematik tomondan tadqiq qilish yoki juda qiyin yoki mumkin emas. Bunday holatlarda algoritmnii faoliyat jarayonida va qiyinligi bo‘yicha tekshiradi. Bundan tashqari dasturlarni hisoblash imkoniyatlarini aniqlash uchun ham testlash maqsadga muvofiq. Ko‘p dasturlar qandaydir kiritiladigan ma’lumotlar bilan yahshi ishlaydi, boshqalari bilan yomon ishlaydi. “Yahshi” lardan “yomon” larga o‘tish “mayin” bo‘lish kerak. Testlash uchun ma’lumotlar dasturning qiyinligiga, mavjud vaqt resurslariga, kiritish-chiqarishsoniga bog‘liq holda tanlanadi. Bu yerda analitik va eksperimental tahlil bir-birini to‘ldiradi.

Hujjatlashtirish

O‘zingiz yozmagan dastur kodini o‘qish juda qiyin. Bu muammoni hujjatlashtirish yordamida yechsa bo‘ladi. Hujjatlashtirish o‘z ichiga hamma yordamchi ma’lumotlarni oladi va dasturda nima bajarilishini tushuntirib beradi, xususan, blok-sxemalardagi boshqarishni uzatish, berilganlarni kiritish-chiqarish shaklini batafsil tavsif qilish, siklning parametrlari, yordamchi local va global proseduralarni bajarilishi va boshqalar.

Hujjatlashtirishning eng asosiy qoidasi bu “boshqalar yozgan dasturlarni qanday ko‘rishni istasangiz, o‘zingiz ham dasturni shunday ko‘rinishda rasmiylashtiring”.

TOPSHIRIQ

1. Uchburchak yuzasini uning tomonlari berilgan bo‘lsa, Geron formulasi bo‘yicha yuzasini xisoblovchi dastur tuzing.
2. Jismning tezlanishi doimiy bo‘lsa va dastlabki lahzada uning tezligi v_0 –teng bo‘lsa. t vaqtdan so‘ng uning bosib o‘tgan yo‘lini aniqlang.
3. N -balandlikdan erkin tushish vaqtini aniqlovchi dastur tuzing
4. R -qarshilikga ega o‘tkazgichdan t -vaqtda I -tok kuchi o‘tganda ajraluvchi issiqlik miqdorini aniqlang.
5. a, b, c –parallelopiped tomonlari bo‘lsa, uning xajmini aniqlang.
6. a, b kattaliklarning qiymatlari berilgan. r -kattalikdan foydalanib ularning qiymatlarini almashtiring.
7. Uchburchakning a, b, c tomolari berilgan. Uchburchak yuzasi topilsin.
8. x, y, z haqiqiy sonlari berilgan. Ularning maksimal topilsin.
9. x, y, z haqiqiy sonlari berilgan. Ushbu sonlarga mos uchburchak hosil qilish mumkinmi?
10. x, y, z haqiqiy sonlari berilgan. $x+y+z$ va $x*y*z$ miqdorlarning maksimali topilsin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

11. Haqiqiy a son berilgan. Quyidagi funktsiyaning nuqtadagi qiymati topilsin.
12. Berilgan x haqiqiy sonning butun qismi topilsin.
13. Kvadrat tenglama yechimi topilsin.

3-AMALIY MASHG‘ULOT

MAVZU: ALGORITMLAR TAHLILI. ALGORITMLARNI ISHLAB CHIQUISH METODLARI.

Mashg‘ulotning maqsadi: Talabalarga algoritm va hisoblash jarayoni orasidagi bog‘lanish mavzusida yetarli ko‘nikma hosil qilish.

NAZARIY QISM

1-misol. R radiusli doiraning yuzasini hisoblash algoritmi tuzilsin. Avval aytib o‘tilganidek, algoritmda boshlang‘ich qiymatlar o‘rniga ularning nomlari ishtirok etishi mumkin va bu algoritmning ommaviylik xossasiga aloqadorligini bildiradi. Bu masalada ham radiuslar guruhi R nomi bilan berilmoqda va uning joiz boshlang‘ich qiymati ixtiyoriy haqiqiy son bo‘lishi mumkin. Eslatib o‘tamiz, algoritmda turli nomlar ishtirok etishi va ular boshlang‘ich qiymatlar va natijalar nomi bo‘lishi ham mumkin. Masalaning quyida keltirilgan yechimidagi S nomi masalani yechimi bo‘ladigan natijalar guruhining nomidir.

1) **Boshlanish;**

2) **R ning qiymati aniqlan- R ning qiymatini kiritish**

3) **R ning R ga ko‘paytirib, S deb olinsin;**

4) **S ni 3,14 ga ko‘paytirib, S deb olinsin;**

5) **javob sifatida S yozilsin;**

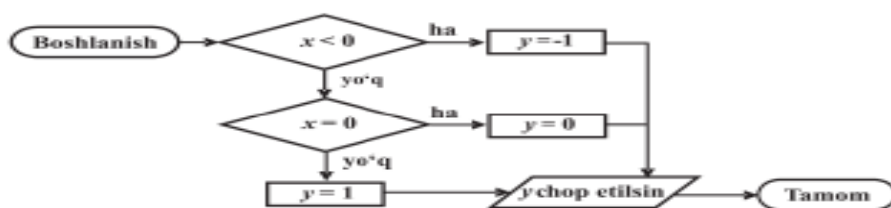
6) **tugallansin.**



2-misol. Algoritmi formula yordamida berilgan

$$y = \begin{cases} -1, & \text{agar } x < 0 \\ 0, & \text{agar } x = 0 \\ 1, & \text{agar } x > 0 \end{cases}$$

funksiyaning qiymatini hisoblashga doir tarmoqlanuvchi algoritmni blok-sxema yordamida tasvirlaymiz:



MISOLLAR

2.1 - Misol. Sonli massiv $A = (a_1, a_2, \dots, a_N)$ ning elementlarini yig'indisini hisoblang.

Test

Berilgan		Natija
N=5	A=(3, 5, -2, 6, 3)	S=15.0

Algoritmi:

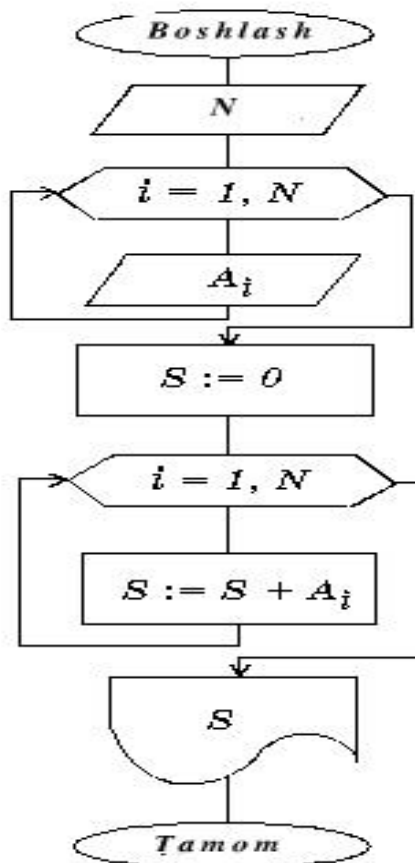
alg Summa (**but** N,
haqqid A[1:N], **haq** S)
arg N,A
boshlbut i
 S:=0
sb i uchun 1 dan N gacha
 S := S + A[i]
so
tamom

Blok sxemasi:

Algoritmning bajarilishi

i	S
	0
1	$0 + a_1 = 0 + 3 = 3$
2	$a_1 + a_2 = 3 + 5 = 8$
3	$a_1 + a_2 + a_3 = 8 - 2 = 6$
4	$a_1 + a_2 + a_3 + a_4 = 6 + 6 = 12$
5	$a_1 + a_2 + a_3 + a_4 + a_5 = 12 + 3 = 15$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



2.2 - Misol. Berilgan $A = (a_1, a_2, \dots, a_N)$ massivning eng katta qiymatini va uning nomerini aniqlang.

Test

Berilganlar		Natija	
N=4	A=(3, -1, 10, 1)	Amax=10	K=3

Algoritmi:

alg Maks (but N, haq jad A[1:N], haq Amax, but k)

arg N, A

natija Amax, k

boshlbut i

Amax := A[1]; k := 1

sb iuchun 2 dan N gacha

agar A[i] > Amax

u holda Amax:=A[i]; k := i

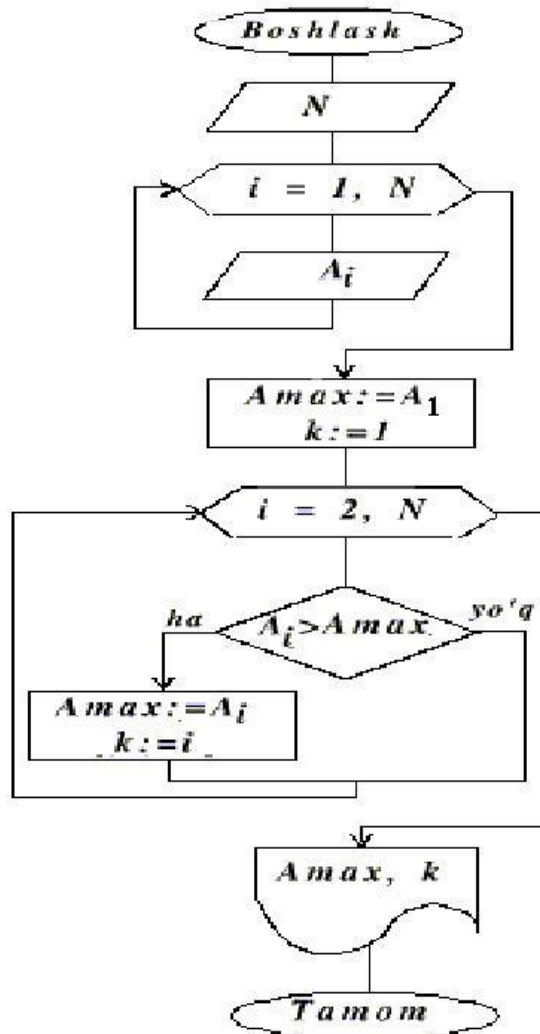
hal bo'ldi

so

tamom

blok-sxemasi:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



Algoritmning bajarilishi

I	A[I] > Amax	Amax	k
2	-	3	1
3	+	10	3
4	-		

2.3 - Misol. Sinfdagi o‘quvchilar ichidan basketbol komandasi tuziladi. Komandaga bo‘yi 170 dan yuqori bo‘lgan o‘quvchilar qabul qilinadi. Komandaga tushadigan davogarlarning ro‘yxatini aniqlang.

Test

Test tartibi	Tekshirish	O‘quvchilar soni	Familiyasi	Bo‘yi	Natija
1	Nomzodlar bor	3	Kulov Chexin Uvarov	171 165 178	Kulov Uvarov
2	Nomzodlar yo‘q	2	Ershov Ivanov	170 165	Nomzodlar yo‘q

Algoritmi:

alg Basketbol (**but** N, **lit jad** Fam[1:N], Nomz[1:N],

haq jad Bo‘yi[1:N])

argN, Fam, Bo‘yi

boshlbut i, k

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

k:=0

sb iuchun 1 dan N gacha

agar Bo‘yi[i]>170

u holda k:=k+1; Bo‘yi [k] := Fam [i]

hal bo‘ldi

so

agark=0

u holdachiqarish "SINFDA NOMZODLAR YO‘Q."

aks holda **sb** i uchun 1 dan k gacha

chiqarish Nomz[i]

so

hal bo‘ldi

tamom

Algoritmning bajarilishi

Test tartibi	i	Bo‘yi[i] > 170	K	Komandaga nomzodlar
1	1	+	0	Kulov Uvarov
	2	-	1	
	3	+	2	
2	1	-	0	-
	2	-		

2.4 - Misol. X(N) massiv berilgan. Yangi Y(N) massivi hosil qilingki unda X ning avval hamma musbat keyin nol undan keyin manfiy sonlari joylashgan bo‘lsin.

Test

Berilganlar	Natija
N=7 X=(-1, 2, 0, 4, -3,-2,0)	Y=(2, 4, 0, 0, -1, -3, -2)

Algoritmi

alg Yangi_Tartib (but N, haqjad X[1:N], Y[1:N])

arg N, X

natija Y

boshlbut i, k

k := 0

sb iuchun 1 dan N gacha

agar X[i] > 0

u holda k := k+1; Y[k] := X[i]

hal bo‘ldi

so

sb iuchun 1 dan N gacha

agar X[i] = 0

u holda k := k+1; Y[k] := X[i]

hal bo‘ldi

so

sb iuchun 1 dan N gacha

agar X[i] < 0

u holda k := k+1; Y[k] := X[i]

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

halbo‘ldi

so

tamom

Mustaqil ishlash uchun masalalar

1. Berilgan $X(N)$ massivda eng kichik va eng katta elementlar o‘rnini almashtiring.
2. Berilgan $X(N)$ massivdagi berilgan qiymatdan kichiklarning sonini aniqlang.
3. Berilgan $A(N)$ massivning komponentlarini chapga bir pozisiya siklik siljishni amalga oshiring, ya’ni $A = (a_2, a_3, \dots, a_N, a_1)$ vektorni hosil qiling.
4. Berilgan $A(N)$ massivning komponentlarini o‘ngga ikki pozisiya siklik siljishni amalga oshiring, ya’ni $A = (a_{N-1}, a_N, a_1, a_2, \dots, a_{N-2})$ vektorni hosil qiling.
5. Berilgan $A(N)$ massivelementlarini $B(N)$ massivga teskaritartib bilan yozing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

4-AMALIY MASHG‘ULOT

MAVZU: ALGORITMNING ASOSIY TIPLARI: CHIZIQLI, TARMOQLANUVCHI VA TAKRORLANUVCHI ALGORITMLAR VA ULARNING BLOK-SXEMALARI.

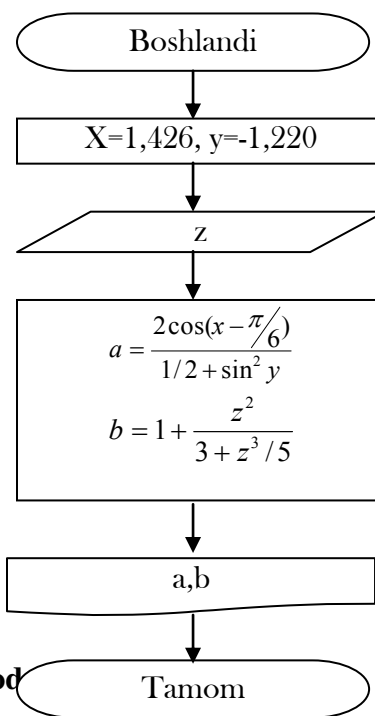
Mashg‘ulotning maqsadi: Algoritmning asosiy tiplari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar va ularning blok-sxemalarini tuzish bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

TOPSHIRIQ: Quyida berilgan ifodalarni blok sxemasi tuzing.

1-topshiriq uchun namuna.

Hisoblash kerak	Berilgan
$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y} \quad b = 1 + \frac{z^2}{3 + z^3/5}$	$x = 1,426; y = -1,220; z = 3,5.$

Blok-sxema



Topshiriq: Arifmetik ifod

№	Arifmetik ifoda
1.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y} \quad b = 1 + \frac{z^2}{3 + z^3/5}$
2.	$j = \left x^{y/x} - \sqrt[3]{y/x} \right \quad \psi = (y-x) \frac{y-z/(y-x)}{1+(y-x)^2}$
3.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}, \quad \varphi = x(\sin x^3 + \cos^2 y)$
4.	$y = e^{-bt} \sin(at + b) - \sqrt{bt + a}, \quad S = b \sin(at^2 \cos 2t) - 1$
5.	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x + a) / x$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

	$y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$
6.	$S = x^3 \operatorname{tg}^2(x+b)^2 + \frac{a}{\sqrt{x+b}}, Q = \frac{bx^2 - a}{Ax}$
7.	$R = x^2(x+1)/b - \sin^2(x+a), S = \sqrt{xb/a} + \cos^2(x+b)^3$
8.	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}, Z = \frac{x^2}{A} + \cos(x+b)^2$
9.	$f = \sqrt[3]{\operatorname{mtgt} + c \operatorname{sint} }, z = m \cos(bt \operatorname{sint}) + c$
10.	$y = btg^2 x - \frac{A}{\sin^2(x/a)}, S = b \sin(at^2 \cos 2t) - 1$
11.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, b = 1 + \frac{z^2}{3 + z^3/5}$
12.	$j = \left x^{y/x} - \sqrt[3]{y/x} \right , \psi = (y-x) \frac{y - z/(y-x)}{1 + (y-x)^2}$
13.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}, \varphi = x(\sin x^3 + \cos^2 y)$
14.	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }, S = b \sin(at^2 \cos 2t) - 1$
15.	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/x, y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$
16.	$S = x^3 \operatorname{tg}^2(x+b)^2 + \frac{a}{\sqrt{x+b}}, Q = \frac{bx^2 - a}{Ax}$
17.	$R = x^2(x+1)/b - \sin^2(x+a), S = \sqrt{xb/a} + \cos^2(x+b)^3$
18.	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}, Z = \frac{x^2}{A} + \cos(x+b)^2$
19.	$f = \sqrt[3]{\operatorname{mtgt} + c \operatorname{sint} }, z = m \cos(bt \operatorname{sint}) + c$
20.	$y = btg^2 x - \frac{A}{\sin^2(x/a)}, S = b \sin(at^2 \cos 2t) - 1$
21.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, b = 1 + \frac{z^2}{3 + z^3/5}$
22.	$j = \left x^{y/x} - \sqrt[3]{y/x} \right , \psi = (y-x) \frac{y - z/(y-x)}{1 + (y-x)^2}$
23.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}, \varphi = x(\sin x^3 + \cos^2 y)$
24.	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }, S = b \sin(at^2 \cos 2t) - 1$

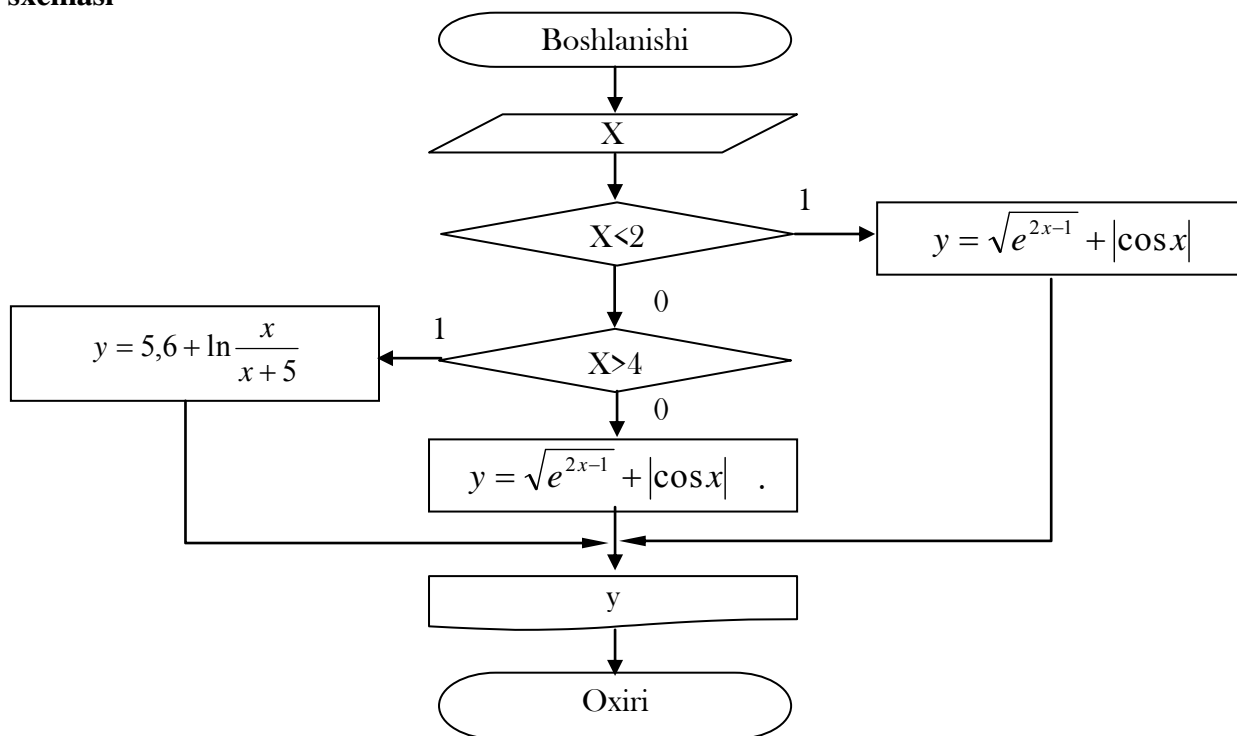
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

25.	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/x$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$
26.	$S = x^3 \operatorname{tg}^2(x+b)^2 + \frac{a}{\sqrt{x+b}}$, $Q = \frac{bx^2 - a}{Ax}$
27.	$R = x^2(x+1)/b - \sin^2(x+a)$, $S = \sqrt{\frac{xb}{a}} + \cos^2(x+b)^3$
28.	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x+b)^2$
29.	$f = \sqrt[3]{mtgt + c \sin t }$, $z = m \cos(bt \sin t) + c$
30.	$y = bt \operatorname{tg}^2 x - \frac{A}{\sin^2(x/a)}$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$

2-topshiriq uchun namuna

№	Ifoda	Shart
31	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$X < 2$ $2 \leq x \leq 4$ $x > 4$

Blok-sxemasi



“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

Quyidagi topshiriqlarni algoritm, blok-sxemasini tilida tuzing:

1	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} & x < 1 \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} & 1 \leq x \leq 2,5 \\ \operatorname{arctg}(\cos x) & x > 2,5 \end{cases}$
2	$y = \begin{cases} 3,2 + \ln x^2 + 1 & x \leq 2,1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) & 2,1 < x \leq 4 \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} & x > 4 \end{cases}$
3	$y = \begin{cases} 1 + \sin x^2 - 2 & x \leq 1,5 \\ \operatorname{tg}\sqrt{x^2 + 1} + \sin x & 1,5 \leq x \leq 10 \\ \operatorname{arcsin}(\cos(x^3 + 1)) & x > 10 \end{cases}$
4	$y = \begin{cases} \ln ax^2 + b & x \leq a \\ e^{\sqrt{x^3 + b}} + \lg x^3 & a < x \leq b \\ e^x + \lg(ab + 1) & x > b \end{cases}$
5	$y = \begin{cases} 3,2 + \ln x^2 + 1 & x \leq 2,1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) & 2,1 < x \leq 4 \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} & x > 4 \end{cases}$
6	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x & x < 1 \\ \sqrt{x^2 + 1} + \sin x & 1 \leq x \leq 2 \\ 5,6 + \ln \frac{x}{x+5} & x > 2 \end{cases}$
8	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } & x \leq 2,6 \\ \operatorname{tg}^2(x^3 + 3,5) & 2,6 \leq x < 6 \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\frac{\cos x}{x}} & x \geq 6 \end{cases}$
9	$y = \begin{cases} \operatorname{arcsin} \sqrt{x} + \cos x & x < a \\ e\sqrt{x^3 + ab} + \ln^2 x & a \leq x \leq b \\ \operatorname{tg}(ax^3 + bx) + \sqrt{\cos x} & x > b \end{cases}$
10	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} & x < 3 \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} & 3 \leq x \leq 4,5 \\ \operatorname{arctg}(\cos x) & x > 4,5 \end{cases}$
11	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} & x < 4 \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} & 4 \leq x \leq 5,5 \\ \operatorname{arctg}(\cos x) & x > 5,5 \end{cases}$
12	$y = \begin{cases} 3,2 + \ln x^2 + 1 & x \leq 3,1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) & 3,1 < x \leq 5 \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} & x > 5 \end{cases}$
13	$y = \begin{cases} 1 + \sin x^2 - 2 & x \leq 2,5 \\ \operatorname{tg}\sqrt{x^2 + 1} + \sin x & 2,5 \leq x \leq 12 \\ \operatorname{arcsin}(\cos(x^3 + 1)) & x > 12 \end{cases}$
14	$y = \begin{cases} \ln ax^2 + b & x \leq a \\ e^{\sqrt{x^3 + b}} + \lg x^3 & a < x \leq b \\ e^x + \lg(ab + 1) & x > b \end{cases}$
15	$y = \begin{cases} 3,2 + \ln x^2 + 1 & x \leq 3,1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) & 3,1 < x \leq 6 \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} & x > 6 \end{cases}$
16	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x & x < 2 \\ \sqrt{x^2 + 1} + \sin x & 2 \leq x \leq 4 \\ 5,6 + \ln \frac{x}{x+5} & x > 4 \end{cases}$
17	$y = \begin{cases} e^{x^2-1} + \ln(x^2 + 1) & x < 1 \\ 0,25\ln(x+1) + \frac{1}{x+1} & 1 \leq x \leq 2 \\ \frac{1}{1+ \cos x } + \sqrt{\cos x} & x > 2 \end{cases}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

18	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ \operatorname{tg}^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\cos x} \end{cases}$	$x \leq 3,6$ $3,6 \leq x < 7$ $x \geq 7$	25	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 4,1$ $4,1 < x \leq 6$ $x > 6$
19	$y = \begin{cases} \arcsin \sqrt{x} + \cos x \\ e\sqrt{x^3 + ab} + \ln^2 x \\ \operatorname{tg}(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	$x < a$ $a \leq x \leq b$ $x > b$	26	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$x < 2$ $2 \leq x \leq 4$ $x > 4$
20	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 2$ $2 \leq x \leq 3,5$ $x > 3,5$	27	$y = \begin{cases} e^{x^2-1} + \ln(x^2 + 1) \\ 0,25 \ln(x+1) + \frac{1}{x+1} \\ \frac{1}{1+ \cos x } + \sqrt{\cos x} \end{cases}$	$x < 0$ $0 \leq x \leq 2$ $x > 2$
21	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 6$ $6 \leq x \leq 7,5$ $x > 7,5$	28	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ \operatorname{tg}^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\cos x} \end{cases}$	$x \leq 4,6$ $4,6 \leq x < 8$ $x \geq 8$
22	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 6$ $x > 6$	29	$y = \begin{cases} \arcsin \sqrt{x} + \cos x \\ e\sqrt{x^3 + ab} + \ln^2 x \\ \operatorname{tg}(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	$x < a$ $a \leq x \leq b$ $x > b$
23	$y = \begin{cases} 1 + \sin x^2 - 2 \\ \operatorname{tg} \sqrt{x^2 + 1} + \sin x \\ \arcsin(\cos(x^3 + 1)) \end{cases}$	$x \leq 3,5$ $3,5 \leq x \leq 13$ $x > 13$	30	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 2$ $2 \leq x \leq 4,5$ $x > 4,5$
24	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3 + b}} + \lg x^3 \\ e^x + \lg(ab + 1) \end{cases}$	$x \leq a$ $a < x \leq b$ $x > b$			

3-topshiriqqa doir namuna

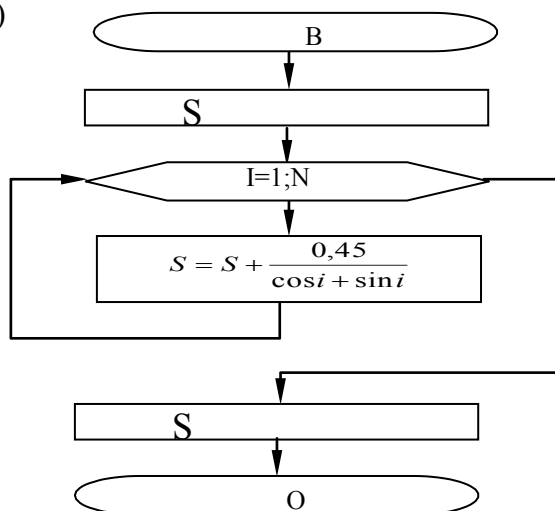
- 1) Yig‘indini hisoblash uchun algoritm va dastur tuzing. Dasturlashda funksiyaning foydalaning.

$$S = \sum_{i=1}^N \frac{0,45}{\cos i + \sin i}$$

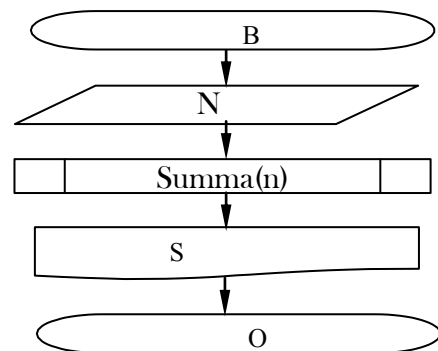
Funksiya blok-sxemasi (1) va funksiyaning asosiy blok-sxemada ishlatilishi (2)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

1)



2)



Topshiriq: a) Quyidagi topshiriqlar algoritmi, blok-sxemasini tuzing.

№	Funksiyani qiymatini hisoblash uchun algoritmi va dastur tuzing.	Yig'indini hisoblash uchun algoritmi va dastur tuzing.	Cheksiz qatorni 0,005 aniqlikda hisoblash algoritmi va dasturini tuzing.
1	$Y = a^3 \sin x^2$ $\Delta X = 0.1; -1 \leq X \leq N$	$S = \sum_{i=4}^N \frac{0.4}{i(i+1) + \sqrt{i}}$	$Z = \sum_{i=1}^{\infty} \frac{1}{i(i+1) + 2 \cos i}$
2	$Y = \sqrt{a} \cos x$ $\Delta X = 0.2; 5 \leq X \leq N$	$S = \sum_{i=1}^N \frac{\cos i}{i^{2.7} + 5}$	$Z = \sum_{i=2}^{\infty} \frac{1}{i^2 + 5 + \ln i}$
3	$Y = a^{\frac{1}{3}} \operatorname{tg} x$ $\Delta X = 0.1; 2 \leq X \leq N$	$S = \sum_{i=4}^N \frac{5 + e^j}{6j + 2.7}$	$Z = \sum_{i=1}^{\infty} \frac{5}{6i^4 + 2.7 + \sqrt{i}}$
4	$Y = 2 + a \ln x$ $\Delta X = 0.2; -3 \leq X \leq N$	$S = \sum_{i=5}^N \frac{1}{\ln i + 1.7}$	$Z = \sum_{i=3}^{\infty} \frac{1}{\ln i + i^2}$
5	$Y = 3.2a\sqrt{x+1}$ $\Delta X = 0.5; 1 < X < N$	$S = \sum_{i=1}^N \frac{1}{\sqrt{K} + 1.2}$	$Z = \sum_{K=1}^{\infty} \frac{4}{3k^3 + \sin K}$
6	$Y = 5e^{xa} + 2.3$ $\Delta X = 0.3; 0 \leq X \leq N$	$S = \sum_{i=1}^N \frac{1}{2e^j + 3.4}$	$Z = \sum_{i=1}^{\infty} \frac{1}{2e^j + \operatorname{tg}^j}$
7	$Y = ax^3 + 2.4e^x$ $\Delta X = 0.4; -5 \leq X \leq N$	$S = \sum_{j=2}^N \frac{\sqrt{5.4}}{3j^3 + j^{\frac{1}{3}}}$	$Z = \sum_{K=1}^{\infty} \frac{4}{3k^3 + \sin K}$
8	$Y = e^{ax} + 2.1x$ $\Delta X = 0.2; -2 \leq x \leq N$	$S = \sum_{i=5}^N \frac{1}{1 + \ln^2 i}$	$Z = \sum_{J=1}^{\infty} \frac{8}{\sqrt{J} + J^4}$
9	$Y = \operatorname{tg} ax + \sqrt{x}$ $\Delta X = 0.3; -0.3 \leq X \leq N$	$S = \sum_{l=2}^N \frac{1 + \sqrt{5}}{\sqrt{5} + L^2}$	$Z = \sum_{j=1}^{\infty} \frac{\ln j}{j^6}$
10	$Y = e^{ax} - e^{x^l}$ $\Delta X = 0.5; -0.5 \leq x \leq N$	$S = \sum_{i=1}^N \frac{\cos^2 i}{i^3}$	$Z = \sum_{i=1}^{\infty} \frac{3.4}{e^{2i}}$
11	$Y = \frac{a + 0.27}{x + \sqrt{1+x}}$ $\Delta X = 0.8; -0.9 \leq X \leq N$	$S = \sum_{i=1}^N \frac{(-1)^i}{i + \operatorname{tgi}}$	$Z = \sum_{li=3}^{\infty} \frac{1}{i^i + 2.7}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

12	$Y = \frac{\sqrt{1+x+0,3}}{a^2x}$ $\Delta X = 0,3; 0 \leq x \leq N$	$S = \sum_{i=3}^N \frac{3}{(-1)^i + 2i^2}$	$Z = \sum_{j=1}^{\infty} \frac{\sqrt{j}}{1+j^6}$
13	$Y = \sin ax + e^a$ $\Delta X = 0,9; -2 \leq X \leq N$	$S = \sum_{i=1}^N \frac{35}{\sqrt{i+\sqrt{i}}}$	$Z = \sum_{i=2}^{\infty} \frac{3}{i+2\sqrt{i}}$
14	$Y = a + e^x + \cos x$ $\Delta X = 0,4; -0,8 \leq x \leq N$	$S = \sum_{i=2}^N \frac{e^i + 1}{i}$	$Z = \sum_{k=1}^{\infty} \frac{k}{e^k + k^2}$
15	$Y = a^2 \sqrt{\sqrt{x+1} + 2}$ $\Delta X = 0,4; 0,3 \leq X \leq N$	$S = \sum_{i=1}^N \frac{1}{ \cos i _i}$	$Z = \sum_{j=1}^{\infty} \frac{0,7}{2 + \ln k }$
16	$Y = \cos ax + \sin a$ $\Delta X = 0,3; 0,1 \leq x \leq N$	$S = \sum_{i=5}^N \frac{1}{\cos i + \sin i}$	$Z = \sum_{i=1}^{\infty} \frac{67}{i^4 + \sqrt{3i}}$
17	$Y = 2,7a^4 + \ln x^2$ $\Delta X = 0,2; -1 \leq X \leq N$	$S = \sum_{j=1}^N \frac{2,7}{\ln j^3 + 1}$	$Z = \sum_{i=6}^{\infty} \frac{ \cos i }{i^2}$
18	$Y = 35,9a^{\frac{1}{4}} \sin x^2$ $\Delta X = 0,1; 0 \leq x \leq N$	$S = \sum_{k=3}^N \frac{k+0,4}{\sqrt{k+\sqrt{k}}}$	$Z = \sum_{j=1}^{\infty} \frac{0,4}{\sqrt{j+100}}$
19	$Y = \sqrt{a+2} \ln x^3$ $\Delta X = 0,4; 0,02 \leq X \leq N$	$S = \sum_{j=1}^N \frac{0,999k}{k^2 + k^{0,7}}$	$Z = \sum_{i=3}^{\infty} \frac{2,4}{ i + e^{i+1}}$
20	$Y = \sqrt{a+2,1} \ln x^3$ $\Delta X = 0,4; 0,02 \leq x \leq N$	$S = \sum_{j=1}^N \frac{k+1}{k^2 + 2}$	$Z = \sum_{L=1}^{\infty} \frac{100}{L^2 + \ln L }$
21	$Y = e^a \operatorname{tg} x + 3$ $\Delta X = 0,3; -0,05 \leq X \leq N$	$S = \sum_{i=1}^N \frac{0,4i}{i \cos i + 3i}$	$Z = \sum_{j=1}^{\infty} \frac{ \cos k + \sin k }{k}$
22	$Y = e^{ax} + 2,91x$ $\Delta X = 0,1; 1 \leq x \leq N$	$S = \sum_{i=2}^N \frac{\sqrt{i+2}}{3i}$	$Z = \sum_{j=1}^{\infty} \frac{0,9}{e^{j+2} + 1}$
23	$Y = \cos^2(2x+a)$ $\Delta X = 0,2; 0,04 \leq X \leq N$	$S = \sum_{j=1}^N \frac{\sqrt{ \cos k }}{k}$	$Z = \sum_{i=1}^{\infty} \frac{0,4}{\sqrt{i^3 + 1}}$
24	$Y = 3 \ln^2(ax^2 + 3)$ $\Delta X = 0,04; 0,4 \leq x \leq N$	$S = \sum_{j=1}^N \frac{0,9}{\sqrt{j^2 + \sqrt{1+j}}}$	$Z = \sum_{i=5}^{\infty} \frac{e^{0,4}}{\ln i^2}$
25	$Y = 0,2ax^2 \operatorname{tg} \frac{x}{a}$ $\Delta X = 0,1; 0,1 \leq X \leq N$	$S = \sum_{j=1}^N \frac{k^{\frac{1}{3}}}{k^3 + \sin k}$	$Z = \sum_{i=1}^{\infty} \frac{2,4+i}{i^i + 2}$
26	$Y = ax^2 + \sin \pi x$ $\Delta X = 0,03; 0,03 \leq x \leq N$	$S = \sum_{j=2}^N \frac{(-1)^{j+1}}{k + \sqrt{0,2k}}$	$Z = \sum_{j=1}^{\infty} \frac{k}{\ln k^5}$
27	$Y = \frac{ax^2 3x + 1}{x}$ $\Delta X = 0,04; 1 \leq X \leq N$	$S = \sum_{i=1}^N \frac{i(i+2)}{i + \sqrt{i+1}}$	$Z = \sum_{j=3}^{\infty} \frac{0,5k}{k^3 + e^j}$
28	$Y = \frac{\sin x}{a\pi x} + 0,4$ $\Delta X = 0,4; 0,08 \leq X \leq N$	$S = \sum_{i=1}^N \frac{\sin(\pi i)}{i^2}$	$Z = \sum_{j=5}^{\infty} \frac{0,9}{e^{j+3} + 3,4}$

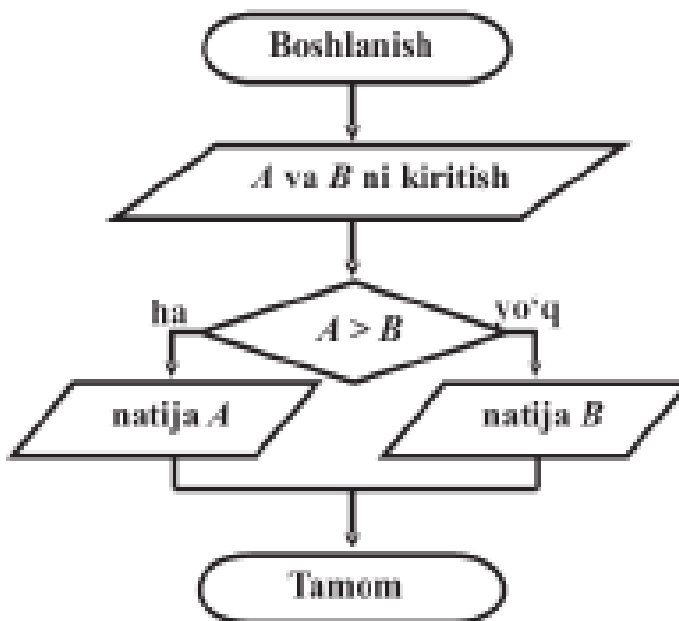
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

29	$Y = a^{\frac{1}{6}} e^{x^2}$ $\Delta X = 0,25; -2 \leq x \leq N$	$S = \sum_{i=3}^N \frac{\cos(\pi i)}{\pi i}$	$Z = \sum_{k=1}^{\infty} \frac{99k}{k^{\frac{1}{3}} + \sqrt{k+k^4}}$
30	$Y = \cos \pi a x + \sin \pi a x$ $\Delta X = 0,5; -3 \leq x \leq N$	$S = \sum_{i=1}^N \frac{0,45}{\cos i + \sin i}$	$Z = \sum_{k=1}^{\infty} \frac{0,49}{\sqrt[4]{k+0,27} + k}$

TOPSHIRIQ

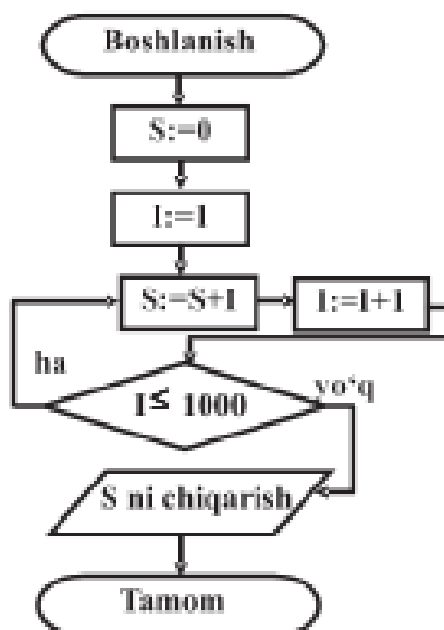
1-misol. Berilgan ikkita A va B sonlardan kattasini topish (IKT nomi bilan ataluvchi) algoritmini so‘zlar va blok-sxema yordamida tuzing.

- 1) Boshlanish;
- 2) A va B kiritilsin;
- 3) agar $A > B$ bo‘lsa
- 4-bandga o‘tilsin;
- aks holda
- 5-bandga o‘tilsin;
- 4) natija A deb olinsin va
- 6-bandga o‘tilsin;
- 5) natija B deb olinsin;
- 6) tugallansin.



2-misol. 1 dan 1000 gacha bo‘lgan sonlar yig‘indisini, ya’ni $S=1+2+3+\dots+1000$ ni hisoblash algoritmini tuzing.

- 1) Boshlansin;
- 2) $S = 0$ deb olinsin
(ya’ni $S := 0$);
- 3) I ning qiymati 1 deb olinsin
(ya’ni $I := 1$);
- 4) S ga I qo‘shilib, S deb olinsin
(ya’ni $S := S + I$);
- 5) I ga 1 qo‘shilib I deb olinsin
(ya’ni $I := I + 1$);
- 6) agar $I \leq 1000$ bo‘lsa
4-bandga o‘tilsin;
- 7) javob deb S olinsin;
- 8) tugallansin.



“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

5-AMALIY MASHG‘ULOT

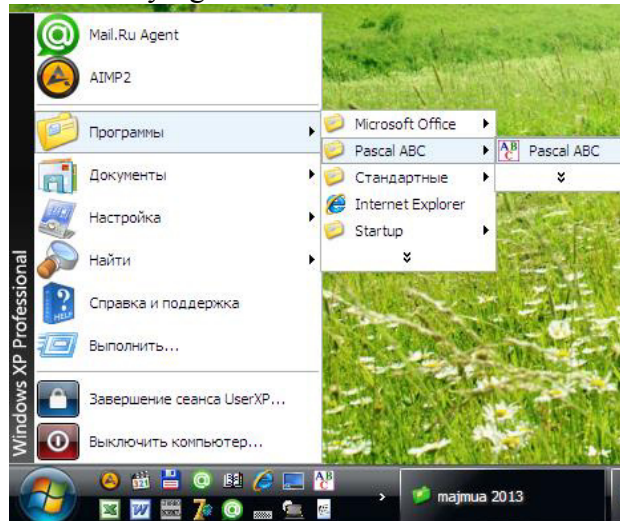
MAVZU. PASKAL TILI DASTURLASH TILINING ALIFBOSI, BUYRUQLAR TIZIMI VA OPERATORLARI.

Mashg‘ulotning maqsadi: Talabalarga Paskal algoritmik tilining alifbosi, Pascal abc muhiti va undagi standart funksiyalar. ifodalarni Pascal tilida tasvirlash. matematik munosabatlar va amallari haqida ko‘nkma hosil qilish.

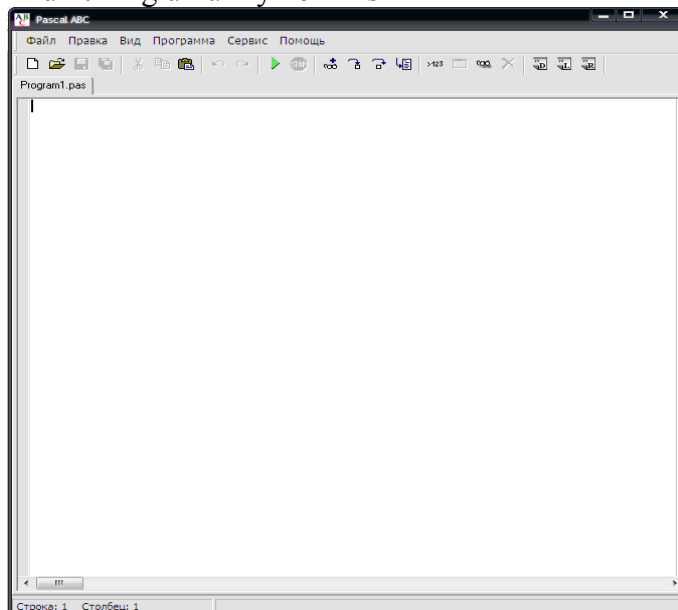
Vositalar: Pascal dasturlash tili o‘rnatilgan kompyuterlar.

NAZARIY QISM:

PaskalABC muhitiga kirish uchun quyidagi bosqichlarni amalga oshiramiz
Пуск-программы-PascalABC buyrug`i tanlanadi

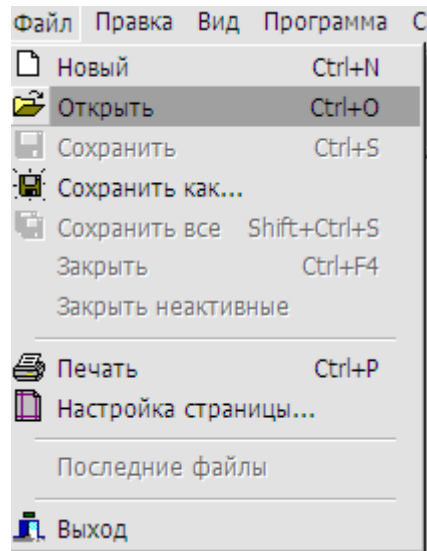


Paskal dasturlash tili muhitining umumiy ko‘rinishi

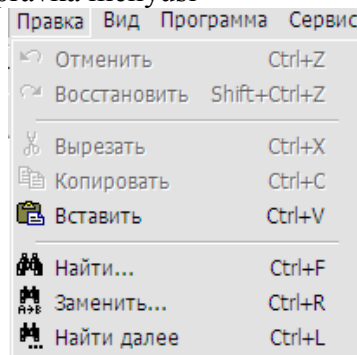


7. Paskal ABC muhitining fayl menyusi

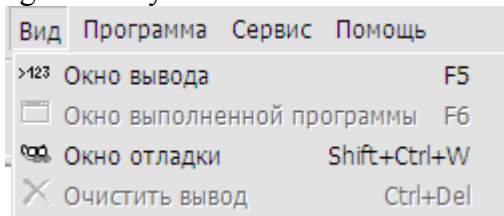
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



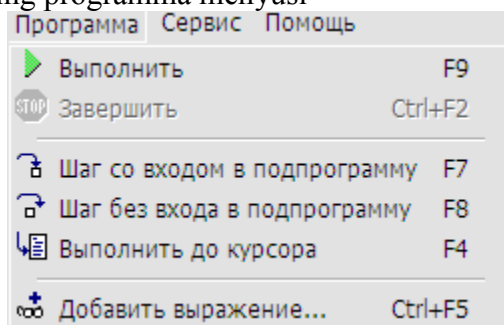
8. Paskal ABC muhitining pravka menyusi



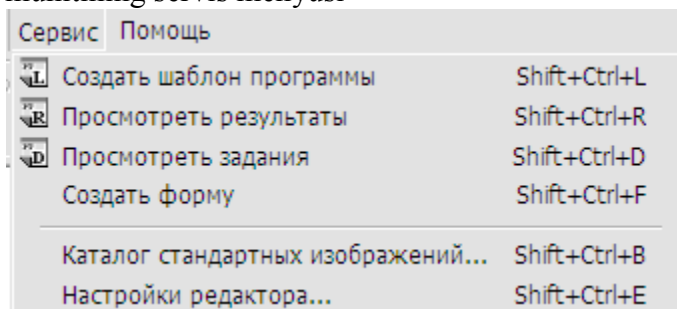
9. Paskal ABC muhitining vid menyusi



10. Paskal ABC muhitining programma menyusi

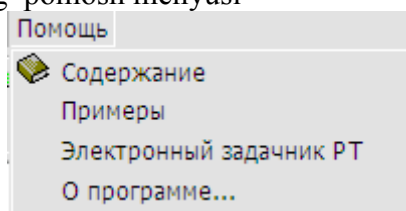


11. Paskal ABC muhitining servis menyusi



“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

12. Paskal ABC muhitining pomosh menyusi



Paskal dasturlash muhitining asosiy fayllarini **turbo.exe**, turbo.tpl, turbo.tph, turbo.tp lar tashkil qiladi. Windows NT, Windows 2000, Windows XP va undan yuqori operatsion sistemalari uchun esa bp.exe fayli ham mavjud bo‘lishi kerak. Undan tashqari *.tpu kengaytmali har xil prosedura va funktsiyalar to‘plamida iborat modul fayllari, grafikada ishlash uchun har xil shriftlarni qo‘llash imkonini beradigan *.chr kengaytmali fayllari bo‘ladi.

MS DOS operatsion sistemasida paskal dasturlash tili muhitiga kirish uchun dastur joylashgan papkaga kiritiladi. Masalan c:\tp_7\ katalogda joylashgan bo‘lsa kursor C: da turgan bo‘lsa c:\tp_7\turbo.exe buyrug‘i yoziladi va ENTER tugmasi bosiladi.

Windows 95, Windows 98 operatsion sistemalarida paskal dasturlash tili muhitiga kirish uchun turbo.exe fayli izlab topilib ishga tushiriladi.

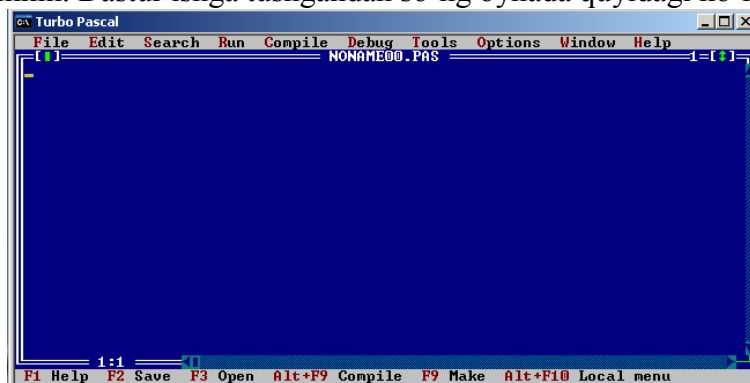
Windows NT, Windows 2000 va undan yuqori operatsion sistemalarida paskal dasturlash tili muhitiga kirish uchun turbo.exe fayli joylashgan papka izlab topiladi. Windows ning Выполнить buyrug‘ida cmd.exe buyrug‘i yoziladi. DOS buyruqlaridan foydalanib turbo.exe fayli ishga tushiriladi. Masalan c:\tp_7\turbo.exe.

Paskal dasturlash tili muhitiga kirgandan keyin avtomatik tarzda yangi noname00.pas nomli dastur kiritish oynasi hosil bo‘ladi. (1-rasm) Yangi oyna yaratish uchun F10→File→New tugma va menyuu buyruqlari ketma-ketligi bosiladi.

Turbo Paskal 7.0 dasturlash tilini muhiti bilan tanishish

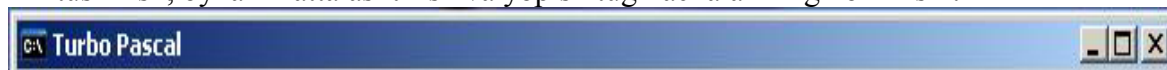


Turbo Paskal 7.0 dasturlash tilining yorlig‘i. Bu yorliq orqali dasturni ishga tushirishimiz mumkin. Dastur ishga tushgandan so‘ng oynada quyidagi ko‘rinish hosil bo‘ladi



Turbo Pascal 7.0 dasturlash tilining oynasi va uning tuzilishi.

1. Turbo Pascal 7.0 dasturlash tilining sarlavha sarti va oynani masalalar paneliga tushirish, oynani kattalashtirish va yopish tugmachalarining ko‘rinishi.

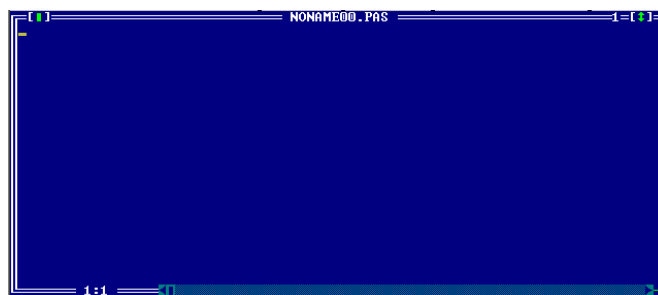


2. Turbo Pascal 7.0 dasturlash tilining menyuu satri va undagi bandlar.

File Edit Search Run Compile Debug Tools Options Window Help

3. Turbo Pascal 7.0 dasturlash tilining ish maydoni.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



Ish maydonda joylashgan kursorni joylashish o‘rnini (satri va ustunini) ko‘satib turuvchi sanagich, ish maydoni yuqori sartida joylashgan “Noname00.pas” nomli satr. Bu satr foydalanuvchi tomonidan dastur kiritilganda va ushbu dasturni kompyuter doimiy xotirasiga shu nom bilan yoki bosha nom bilan saqlaganda saqlangan nom shu satrda paydo bo‘ladi.

Dasturni xotiraga saqlash uchun F2 tugmasi bosiladi, hosil bo‘lgan oynaga dastur fayli nomi kiritiladi va ENTER tugmasi bosiladi.

Oynani yopish uchun Alt+F3 tugmasi bosiladi.

Dastur fayllarini oynaga ochish uchun F3→Enter tugmalari bosiladi va kerakli katalog ochilib yoki joriy katalogdagi kerakli fayl tanlanib ENTER tugmasi bosiladi.

Dasturni tekshirish uchun F9 tugmasi bosiladi. Keyin dasturdagi sintaksis xatolikka kursor kelib xatolik haqidagi ma’lumot yuqorida qizil rangda hosil bo‘ladi. Dastur to‘g‘ri bo‘lsa to‘g‘riligi haqida ma’lumot hosil bo‘ladi va “PRESS ANY KEY” (biror tugmani bosing) yozuvi lipillab turadi.

Dasturni bajartirish uchun Ctrl+F9 tugmalari birgalikda bosiladi. Natijasini ko‘rish uchun Alt+F5 tugmasi bosiladi.

Dasturni qadamma-qadam bajarilishi ta’minlash uchun F7 tugmasini bosib turamiz.

Dasturdan chiqish uchun Alt+X tugmalar ketma-ketligini bosamiz.

Turbo Pascal 7.0 dasturlash tilini kompyuterga yuklagan vaqtda biz ekranda bir necha vazifalarni bajaruvchi integrallashgan muhitga ko‘zimiz tushadi.

Muhitning birinchi satri – sarlavha satri deb yuritiladi

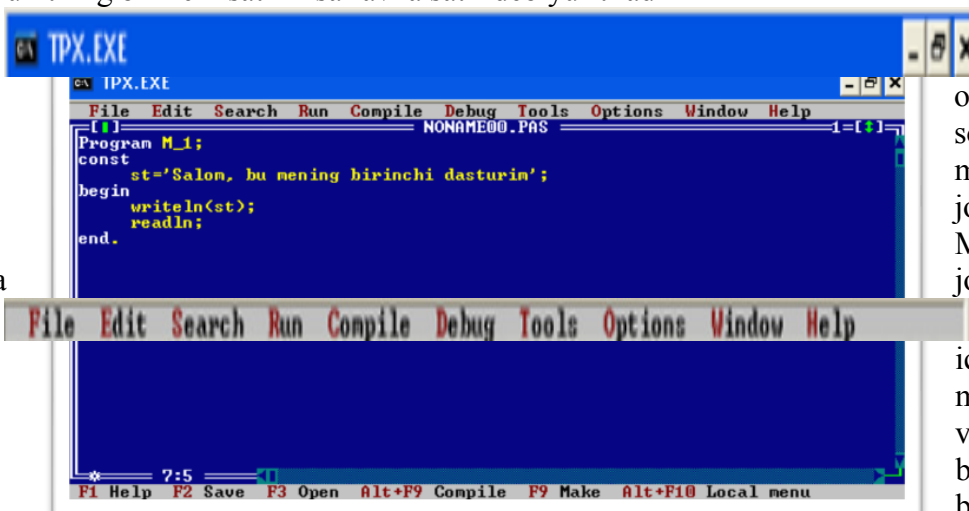
va
sattrdan

qatori

qatorida

har bir

qator



odatda shu
so‘ng
menyular
joylashadi.
Menyular
joylashgan

ichki
menyuda
vazifalarni
bajaruvchi
buyruqlar
joylashgan.

Undan pastda asosiy ish maydoni va holat satri joylashgan. Shuningdek, muhitning asosiy elementlaridan biri bu oynalar hisoblanadi.

Til alfavitining metalingvistik (Bekus - Naur) formulasi quyidagicha bo‘ladi:

<asosiy belgi>::=<harf>|<raqam>|<maxsus belgi>

Harf sifatida katta va kichik lotin harflari ishlatiladi. Ya’ni 26 ta **lotin alifbosi harflari**: A, a, B, b, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk, Ll, Mm, Nn, Oo, Pp, Qq, Rr, Tt, Uu, Vv, Ww, Xx, Yy, Zz. Lekin, matnlar va programmaga izohlar yozish uchun kirill alifbosining bosh va kichik harflarini ham alfavitga kiritilgan.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Kattaliklarning tiplari ular foydalanadigan qiymat va ular bilan bajariladigan amallar bilan aniqlanadilar. Butun va haqiqiy tiplarda kattaliklar uchun qo‘yidagi amallar qo‘llaniladi:

Amallar	Amalning vazifasi	Butun va haqiqiy tiplarda
+	qo‘shish	Qo‘shish amali bajariladi, amalning natijasi qo‘shiluvchilar tipiga mos
-	ayirish	Aayirish amali bajariladi, amalning natijasi qo‘shiluvchilar tipiga mos
*	ko‘paytirish	Ko‘paytirish amali bajariladi, amalning natijasi qo‘shiluvchilar tipiga mos
/	bo‘lish	Bo‘lish amali bajariladi, amalning natijasi faqat haqiqiy tip

Bundan tashqari faqat butun tiplar bilan bajariluvchi quyidagi amallar mavjud:

div	Butunga bo‘lish	amalning natijasi faqat butun tip
mod	Bo‘lish amalidagi qoldiq	

Div amalida bo‘lish paytida qoldiq tashlab yuboriladi;

Mod amalida bo‘lish paytidagi qoldiqni aniqlash.

Paskal tilidagi dastur strukturasi quyidagi ko‘rinishda bo‘ladi

Program Programma mavzusini berish

Uses Ishlatilayotgan kutubxona bo‘limlari (modullari)

Label Dasturning asosiy qismida ishlatilayotgan belgi (metka) larni e‘lon qilish

Const O‘zgarmlarni e‘lon qilish

Type Yangi o‘zgaruvchilarning turini muomalaga kiritish

Var Asosiy dasturda muomalada bo‘ladigan o‘zgaruvchilarni e‘lon qilish

Procedure, Function – Protsedura va Funksiyalarni e‘lon qilish.

Begin

Dasturning asosiy qismi

End.

Demak, har qanday dastur yuqorida berilgan asosiy tuzilmaning xususiy xoli bo‘lishi mumkin va ular o‘z navbatida Paskal tiliga xos bo‘lgan asosiy tushunchalar asosida hosil qilinadi.

Paskal dasturlash tilida uchraydigan standart funksiyalar jadvali.

Funksiyani paskaldagi ifodasi	Funksiyaning matematik ifodasi	Argumentning turi	Funksiyaning turi
abs(x)	x	REAL, INTEGER	REAL, INTEGER
sqr(x)	x^2	REAL, INTEGER	REAL, INTEGER
sqrt(x)	\sqrt{x}	REAL, INTEGER	REAL, INTEGER
exp(x)	e^x	REAL, INTEGER	REAL REAL
ln(x)	$\ln x$	REAL, INTEGER	REAL, REAL
sin(x)	$\sin x$	REAL,	REAL, REAL

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

		INTEGER	
cos(x)	cosx	REAL, INTEGER	REAL, REAL
arctan(x)	arctgx	REAL, INTEGER	REAL, REAL
round(x)	x ni yaxlitlash	REAL	INTEGER
trunc(x)	x ni butun qismini olish	REAL	INTEGER
pred(x)	x dan oldingi qiymatni olish.	INTEGER, CHAR BOOLEAN	INTEGER, CHAR BOOLEAN
succ(x)	x dan keyingi qiymatni olish.	INTEGER, CHAR BOOLEAN	INTEGER, CHAR BOOLEAN
a div b	A ni B ga bo‘lib butun qismini olish	INTEGER, INTEGER	INTEGER
a mod b	A ni B ga bo‘lib qoldig‘ini olish	INTEGER,INTE GER	INTEGER
chr(x)	x ni tartib nomeriga ko‘ra simvolini aniqlash	INTEGER	CHAR
ord(x)	x simvolini tartib nomerini aniqlash	CHAR	INTEGER
odd(x)	x ning toq yoki juftligini aniqlash	INTEGER x-toq, x-juft	BOOLEAN TRUE, FALSE

x^n ko‘rinishdagi ifodani ko‘p uchraydi, uni matematik almashtirishlardan foydalanib $x^{n \cdot \ln x}$ ko‘rinishda yozsa bo‘ladi. Bu ifodani paskaldagi ko‘rinishini $\exp(n \cdot \ln(x))$ shaklda yozish mumkin.

Ekran bilan ishlash operatorlari

TextMode prosedurasi. Bu prosedura adaptorning matn bilan ishlash rejmlaridan birini aniqlash imkoniyatini beradi.

TextColor prosedurasi Kiritiladigan matn belgilari rangini aniqlaydi. Undan foydalanish tartibi quyidagicha: TextColor(Color: Byte);

TextBackground prosedurasi. Fon rangi belgilash uchun ishlatiladi. Undan foydalanish tartibi quyidagicha :TextBackground(Color: Byte);

ClrScr prosedurasi. Bu prosedura ekran va Window prosedurasi yordamida yaratilgan oynani tozalaydi, berilgan fon rangi bilan to‘ldirib, kursorni ekranning yuqori chap nuqtasiga joylashtiradi. Undan foydalanish tartibi quyidagicha: clrscr;

Window prosedurasi. Bu prosedura matn uchun mo‘ljallangan oyna («kichik ekran» ni yaratish uchun mo‘ljallangan. Bu proseduradan foydalanganda kursor oynaning yuqori chap nuqtasiga joylashadi va oyna fon rangi bilan to‘ldiriladi.

GotoXY prosedurasi. Bu prosedura kursorni ekran yoki oynada yangi koordinatali nuqtaga o‘tkazish uchun ishlatiladi.

WhereX va WhereY funkuiyalari. Bu funkuiyalar yordamida kursorning joriy koordinatalari aniqlanadi: WhereX –uning gorizontal, WhereY - vertikal koordinatalari qiymatini aniqlashda qo‘llaniladi.

1. Tayyor tuzilgan dasturlarning tahlili.

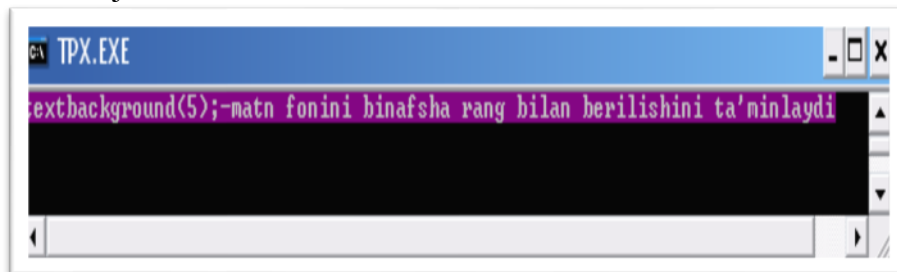
1- Topshiriq:

1- Misol. Textbackground prosedurasi yordamida binafsha rangli fonda matnni chop etuvchi dastur tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
uses crt;
const col80=3;
begin  textmode(col80+font8x8);
textbackground(5);
write(textbackground(5); -matn fonini binafsha rang bilan berilishini ta'minlaydi');
end.
```

Dastur natijasi:



2-Misol. Textcolor prosedurasi yordamida oq rangli fonda qizil rangli matnni chop etuvchi dastur tuzing.

```
uses crt;
const co80=3;
begin  textmode(co80+font8x8);
textbackground(15); textcolor(4);
write('oq fonda qizil rangli matn yozilgan');
end.
```

3-Misol. Ixtiyoriy tugma bosilguncha ekran rangi qizil bo‘lishi va tugma bosilgach, uning rangi qora bo‘lishini ta’minlovchi dastur yarating.

```
Uses CRT;
var
k: Char;
begin
TextBackground(red) ;
ClrScr;
WriteLn('ixtiyoriy tugmani bosing...');
k:= ReadKey;
TextBackground(Black) ;
ClrScr
end.
```

TOPSHIRIQLAR

2- Topshiriq:

1-Misol. Ekranda Window prosedurasi yordamida ikkita oyna ochib, bunda algoritm xossalarini to‘g‘risida chop etuvchi dastur yarating.

2-Misol. clreol prosedurasi vazifasini aniqlang

3-Misol. LowVideo, NormVideo, HighVideo proseduralari yordamida matn yorqinligini turli holatlarda ta’minlovchi dastur yarating.

4-Misol. Mustaqil ravishda TextMode prosedurasining qolgan rejimlarida matnning yozilishini kuzating.

3- Topshiriq:

Quyidagi ifodalarni Pascal tilida yozing:

1-variant
$$y = tg^2 3x + \sqrt{x+0,5} |x| + a \cdot \ln x^3$$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

2–variant $y = \arctg \frac{2x}{z} + ctg 3x^2$

3–variant $q = \frac{|x+y|}{x-\frac{2}{y}} - \arcsin \sqrt{x}$.

4–variant $z = 2^x + \sin(x+y) - \frac{\sqrt[3]{x+y}}{\ln x}, \quad y = 5|x|$

5–variant $k = c \operatorname{tg} \frac{x-4}{0,6y} + \ln e^{0,5xy} - \sqrt[3]{x-0,15},$

$w = \sin^2 \frac{x^3}{|b-a^2|} - a^{2x-\sqrt{b}},$

14–variant

6–variant

$w = \sin^2 \frac{x^3}{2,65} + \ln \arctg^2 x^2 - 3,5\sqrt{x},$

$y = \ln |x-0,6z^2| + \frac{\sqrt{x+y}}{0,5} - \operatorname{tg}^2 x^3,$

15–variant

$y = \arcsin x + 0,69 \cdot a \cdot \operatorname{tg}^3 x - 2^{\sqrt{x+0,4}},$ *by epda* $x = 0,6; \quad a = 5.$

7–variant

16–variant

$y = e^{\sin x} + 0,656 \cdot \operatorname{tg} \frac{x}{z} + 3\sqrt{a+1}, \quad z = x+a,$ *by epda* $x = 1,2; \quad a = 0,69.$

$h = 4,5^a - \cos 2x^2 - \frac{8,46 \cdot c}{5^{a-c}} + \operatorname{ctg} \ln^2 x$

8–variant

17–variant

$z = \arcsin x + \arccos x + \frac{2^{|x-5|}}{|x+y|}.$

$p = \ln e^x - \frac{\sqrt[3]{x+4}}{e^{x-5}} + \cos \frac{2x}{y}, \quad y = \sin x,$ *by epda* $x = 0,159.$

9–variant

18–variant

$c = \ln \operatorname{tg}^2 \sqrt{x} + \frac{|0,6x-y|}{e^{x+y}} - \operatorname{ctg} \sqrt{x},$

$y = e^{\sqrt{2x+5}} + \frac{\ln e^{x-5}}{|x-5|} - 3 \sin x$

10–variant

19–variant

$w = a^{2x-\sqrt{b}} + 6,13 \cdot a \cdot b^2 - \frac{\operatorname{arctg} x}{2,38 \cdot a \cdot b},$

11–variant

$z = 2,6^{\ell nx} + \ell^{\ell n^2 x} - \sin x \cdot \cos x,$

12–variant

$y = \sin \frac{x}{2,6} + \frac{|x+z|}{\sqrt{0,5x}} - 6 \sin x, \quad x = \sqrt[3]{z+5},$

13–variant

$$y = e^{\sqrt{x+\sin x}} + \ln x^3 + \arctg x^2, \quad x = 5z, \text{ by epda } z = 3,5.$$

20–variant $z = \ctg \sin x^2 + \frac{0,55xy}{e^x} + \sqrt{x+y^2}, \quad y = e^{x+5}, \text{ by epda } x = 0,5.$

21–variant $y = e^{\arctg x} + \ln \ctg^2 x - \frac{0,65x}{|x \cdot z|},$

22–variant $a = 3^x + \frac{b+c}{c+d} + \ln(x+b) + e^{c+a}, \quad x = b \cdot c \cdot d.$

23–variant $y = e^{\arcsin x} - 2^{a+b} - \sqrt[3]{x+5ab}, \quad a = x + b,$

24–variant $w = e^{\ln^2 x} - \sin 2,5x + \frac{|x-5|}{\sqrt{x}},$

$$y = \sin \frac{x-5}{0,6x} + \ln \arctg x + \frac{\sqrt{|x-5 \cdot a|}}{0,7x},$$

6- AMALIY MASHG‘ULOT

MAVZU: CHIZIQLI, TARMOQLANUVCHI VA TAKRORLANUVCHI DASTURLAR.

Mashg‘ulotning maqsadi: Munosabat amallari va ularni kattaliklar orasidagi shartlar sifatida qo‘llanilishi. Shartli o‘tish operatorlarini o‘rganish hamda bilim, ko‘nikma va malakalar hosil qilishdan iborat. Tanlash operatori CASE va takrorlanish jarayonlarining uch xil ko‘rinishlari va Paskalning ularga mos operatorlari. Avval sharti tekshiriladigan (old shartli – «hozircha») takrorlanish jarayoni. Sharti keyin tekshiriladigan takrorlanish jarayoni. Parametrlil takrorlanish jarayonlarini o‘rganish hamda bilim, ko‘nikma va malakalar hosil qilishdan iborat.

Qisqacha nazariy ma’lumot

Paskal - dasturning umumiy ko‘rinishini quyidagi ko‘rinishda yozib olaylikda, so‘ng har bir bo‘limni to‘laroq tahlil qilib chiqamiz:

Program <dastur ismi>;

label

<metkalar ro‘yxati>;

const

<o‘zgarmlar va ularning qiymatlari>;

type

<ma’lumotlarning yangi, nostandart turlarini aniqlash>;

var

<o‘zgaruvchilarni, protseduralar va funksiyalarni e’lon qilish>;

begin

<operatorlar bo‘limi>

end.

Arifmetik ifodalar va operatsiyalar

Arifmetik ifodalar, arifmetik operatsiyalar, butun sonli bo‘lish (DIV), (MOD) moduli bo‘yicha bo‘lish, arifmetik I (AND), chapga siljitish (ShL), unga siljitish (ShR), mantiqiy qo‘shish (OR), yoki (XOR), (+) saqlash xar operatsiyasi, (-) saqlash ular operatsiyasi, NOT operatsiyasi.

Arifmetik ifodalar butun va xaqiqiy qiymatlarni qaytaradi. Arifmetik ifodalarning eng sodda ko‘rinishlari quyidagicha:

belgisiz butun va xaqiqiy konstantalar;

butun va xaqiqiy o‘zgaruvchilar;

butun va xaqiqiy toifadagi massiv elementlari;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

butun va xaqiqiy qiymatlarni qabul qiladigan funktsiyalar.

Massiv elementi va o‘zgaruvchisining qiymati arifmetik ifodada paydo bo‘lgunga qadar aniqlangan bo‘lishi kerak.

Arifmetik operatsiyalar ifodalarda butun va xaqiqiy toifali operatsiyalarda qiymatlari ustida arifmetik amallarni bajaradi. PascalABC tilining arifmetik operatsiyalari 1-jad.da keltirilgan.

Arifmetik operatsiyalar

Operatsiya	Amallar	Operandalar toifasi	Natija toifasi
+	Qo‘shish	Butun	Butun xaqiqiy Butun
-	Ayirish	Xaqiqiy butun	xaqiqiy
*	Ko‘paytirish	Xaqiqiy butun	Butun xaqiqiy
/	Bo‘lish	Xaqiqiy butun Butun	Butun
Div	Butun sonli bo‘lish		Butun
Mod	sonli bo‘lish	Butun	Butun
AND	Qoldiq	Butun	Butun
Shl	Arifmetik Va	Butun	Butun
Shr	chapga Surish	Butun	Butun
Or	O‘ngga Surish	Butun	Butun
Xor	Arifmetik Yoki	Xaqiqiy Butun	Butun
Unar +	Dizyunksiya	Xaqiqiy Butun	Xaqiqiy Butun
-	Belgini Saqlash	Butun	Xaqiqiy Butun
Not	Belgini Inkori	Arifmetik Inkori	Butun

Mantiqiy operatsiyalar

Operatsiya	Amallar	Ifoda	A	B	Natija
NOT	Mantiqiy inkori	NOT A	True False		False True
AND	Mantiqiy VA	A AND B	True False	True False	True False
OR	Mantiqiy YOKI	A OR B	True False	True False	True True
XOR	YOKI bekor qilish	AxorB	True False	True False	False True

@ Operatsiyasi

@ operatsiyasi yordamida ko‘rsatkichni o‘zgaruvchiga aylantirish mumkin.

Shartsiz o‘tish operatori GOTO n; Bu buyruq o‘tishni n – belgiga borishni ta’minlaydi. Masalan: goto 20; 20-belgiga o‘tishni ta’minlaydi.

Paskal tilida **shartli o‘tish operatorining** ikki turi mavjuddir: to‘liq va qisqa. Shartli o‘tish operatorining to‘liq ko‘rinishi quyidagicha:

<to‘liq shartli operator> ::= IF <mantiqiy ifoda> THEN <operator> ELSE <operator>

Bu yerda IF (agar), THEN (u holda) ELSE (aks holda) degan xizmatchi so‘zlar.

Mantiqiy amallar(AND, OR, NOT) murakkab mantiqiy ifodalar yozishda ishlatiladi va qavslarga olib yoziladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Qisqa shartli operator:

< Qisqa shartli operator> ::= IF <mantiqiy ifoda> THEN <operator>

Ko‘p hollarda dasturchi uchun shartli operatorning umumlashgan shakli – tanlash operatori qulayroq bo‘ladi. Tanlash operatorining umumiy ko‘rinishini quyidagicha:

CASE S OF

M1: A1;

M2: A2;

MN: AN;

END;

ko‘rinishda yozish mumkin.

Bu yerda: CASE – xizmatchi (tanlash) so‘zi, S – selektor, OF-dan, Mi –operatorlar belgilari, Ai - operatorlar.

Tanlash operatori

Tanlash operatori, selektor, parametrlar ro‘yxati, tanlash o‘zgarmlari ro‘yxati.

Tanlash operatori CASE mavjud variantlardan birini tanlash imkoniyatini beradi. U tanlash o‘zgarmlari ro‘yxati (ro‘yxat bitta o‘zgarmsdan iborat bo‘lishi mumkin), selektor deb nomlangan ifodadan va parametrlar ro‘yxatidan iborat.

Yozilishi:

CASE <ifoda-selektor> OF

<ro‘yxat1>: <operator1>;

<ro‘yxat k2>: <operator2>;

...

<ro‘yxat N>: <operatorN>

ELSE <operator>

END;

O‘zgarmlar toifasi doimo selektor toifasiga to‘g‘ri kelishi kerak. Selektor uchun real va string toifalari man etilgan. CASE operatori quyidagicha ishlaydi. Birinchi navbatda selector ifoda qiymati xisoblanadi, keyingi navbatda joriy selektor qiymatiga teng bo‘lgan o‘zgarmlar qatnashgan operator bajariladi.

Agar xech qaysi o‘zgarmlar selektorning joriy qiymatiga teng bo‘lmasa ELSE so‘zidan keyingi operator bajariladi. Agar ELSE so‘zi bo‘lmasa END so‘zidan keyingi operator ishga tushadi, ya‘ni CASE chegarasidan keyingi operator.

Selektor butun sonli (-32768..32767 diapazonida bo‘lgan) Bulev, liter yoki foydalanuvchi toifasiga bog‘liq bo‘lishi kerak.

O‘zgarmlar qiymatlar ro‘yxati tasodifiy qiymat yoki diapazondan iborat, ular bir-biridan vergul orqali ajratiladi. Diapazon chegaralari biri-biridan ".." belgisi yordamida ajratilgan o‘zgarmlar sonlar yordamida yoziladi. O‘zgarmlar toifasi selektor toifasiga to‘g‘ri kelishi kerak. Quyida CASE operatorining yozilish tartibi ko‘rsatilgan:

Takrorlanishlar soni ma‘lum takrorlanish operatori quyidagicha:

FOR <o‘zgaruvchi>:=<boshlang‘ich qiymat> TO <oxirgi qiymat> DO

<sikl tanasi>;

Masalan: for i:=1 to 10 do

Write(I, ‘- qadam’);

Sharti avval tekshiriladigan takrorlanish operatori quyidagicha:

WHILE <mantiqiy ifoda > DO <operatorlar>;

yoki

WHILE L DO M;

Bu yerda: WHILE – hozircha, DO – bajarish ma‘nosini anglatuvchi xizmatchi so‘zlar, L – mantiqiy ifoda, M – operatorlar yoki operatorlar guruhi, u takrorlanish tanasini belgilaydi. Takrorlanish tanasida bitta yoki bir nechta operatorlar guruhi bo‘lishi mumkin. Bunda operatorlar guruhi albatta Begin va End operatorlari orasida yozilishi kerak.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Takrorlanishlar soni noma’lum bo‘lganda va sharti keyin takrorlanish operatori REPEAT quyidagicha ko‘rinishga ega:

REPEAT S UNTIL b;

Bu yerda: b – mantiqiy ifoda, S – sikl tanasi.

1-Misol $S = \sum_{i=1}^{10} i^3$ ni dasturini tuzing.

Program summa (input, output);

Var I: Integer, S: Real;

Begin

S: =0;

*For i=1 to 10 do s=s+i*i*i;*

Write ('S=', S);

End.

Ayrim paytlarda, sikl parametrini o‘tib borish emas, balki kamayish tartibida o‘zgartirish mumkin, bu holda sikl operatori quyidagi formada yoziladi:

for k:=k2 downto k1 do S;

bu yerda **down to** (gacha kamayib) – tilning xizmatchi so‘zi.

Bu operatorida k parametri k2 dan toki k1 gacha kamayish tartibida (agar k - butun qiymatli o‘zgaruvchi bo‘lsa sikl qadami - 1 ga teng) o‘zgaradi. Operatorning ishlash prinsipi oldingi operatornikiday qolaveradi.

Misol. Yuqorida ko‘rsatilgan misolni dasturini qaytadan tuzaylik. Bu holda dasturdagi sikl operatorigina o‘zgaradi xolos:

for i:=n downto 1 do

qolgan operatorlar esa o‘z o‘rnida o‘zgarmay qoladi.

Dasturda parametrli takrorlash operatoridan foydalanish jarayonida, sikl parametrining qiymatini sikl tanasi ichida o‘zgartirmaslik lozim, aks holda operatorning ish ritmi buzilishi mumkin. Buni quyidagi misollarda ko‘rish mumkin:

To‘g‘ri tuzilgan dastur qismi

for i:=1 to 10 do

Begin

*s:=i*i;*

writeln(s);

end;

Noto‘g‘ri tuzilgan dastur qismi

for i:=1 to 10 do

Begin

*s:=i*i;*

writeln(s);

i:=i+3

end;

2. Repeat takrorlash operatori

Yuqorida aytib o‘tganimizdek, sikldagi takrorlanishlar soni oldindan ma’lum bo‘lsa, parametrli (**for**) sikl operatori foydalanish uchun juda qulay. Lekin, ko‘pgina hollarda, takrorlanuvchi jarayonlardagi takrorlanishlar soni oldindan ma’lum bo‘lmaydi, sikldan chiqish esa ma’lum bir shartning bajarilishi yoki bajarilmasligiga bog‘lik holda bo‘ladi. Bu hollarda **repeat** yoki **while** sikl operatorlaridan foydalanish zarur. Agar sikldan chiqish sharti, takrorlanuvchi jarayonning oxirida joylashgan bo‘lsa **repeat** operatoridan, bosh qicmida joylashgan bo‘lsa **while** operatoridan foydalanish maqsadga muvofiqdir.

Repeat operatorining yozilish formasi quyidagicha bo‘ladi:

repeat S1; S2; ... SN **until** B;

bu yerda **repeat** (takrorlamoq), **until** (gacha) - xizmatchi so‘zlar;

S1, S2, ..., SN lar esa sikl tanasini tashkil etuvchi operatorlar;

B - sikldan chiqish sharti (mantiqiy ifoda).

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Operatorning ishlash prinsipi juda sodda, ya’ni siklning tanasi B mantiqiy ifoda rost qiymatli natija bermaguncha takror - takror hisoblanaveradi. Misol sifatida, yana yuqoridagi yig‘indi hisoblash misolini olaylik.

```
Program Sum2;  
var i, n: Byte;  
    S: real;  
begin  
    readln(n);  
    S:=0; i:=1;  
    repeat  
        S:= S+1/i;  
        i:=i+1;  
    until i>n;  
    writeln (S)  
end.
```

Ayrim takrorlanish jarayonlarida sikldan chiqish shartini ifodalovchi mantiqiy ifoda hech qachon True (rost) qiymatga erishmasligi mumkin. Bu xolda dasturning takrorlash qismi cheksiz marta qaytadan hisoblanishi mumkin, ya’ni dasturchilar tili bilan aytganda «**dastur osilib qoladi**» shuning uchun, operatordagi shartni tanlashda e’tiborli bo‘lish lozim.

E’tiboringizga ya’na bir, ismni qidirib topish dasturini xavola qilamiz:

```
Program BRV;
```

```
Var
```

```
a,b:String[20];
```

```
Begin
```

```
a:='Jamshid';
```

```
Repeat
```

```
Writeln("Tanlagan ismingizni kiriting");
```

```
Readln(B);
```

```
if a<>b Then writeln('Noto‘g‘ri') else writeln('YAshang to‘g‘ri topdingiz');
```

```
Until A=B;
```

```
End.
```

3. While takrorlash operatori

Ahamiyat bergan bo‘lsangiz, repeat operatorida siklning tana qismi kamida bir marta hisoblanadi. Lekin, ayrim paytlarda, shu bir marta hisoblash ham yechilayotgan masalaning mohiyatini buzib yuborishi mumkin. Bunday hollarda, quyidagi formada yoziluvchi while sikl operatoridan foydalanish maqsadga muvofiqdir:

```
while B do S;
```

bu yerda **while** (hozircha), **do** (bajarmoq) - xizmatchi so‘zlari;

B - sikldan chiqishni ifodalovchi mantiqiy ifoda;

S - siklning tanasini tashkil etuvchi operator.

Bu operatorida oldin V sharti tekshiriladi, agar u **false** (yolg‘on) qiymatli natijaga erishsagina sikl o‘z ishini tugatadi, aks holda siklni tana qismi qayta - qayta hisoblanaveradi.

While operatoriga misol sifatida, yana yuqorida berilgan yig‘indi hisoblash misolini ko‘rib chiqaylik:

```
program sum3;  
var i, n: byte;  
    S: real;  
begin  
    readln(n);  
    i:=1; S:= 0;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
    while i<=n do
    begin
        S:= S + 1/i;
        i:= i+1;
    end;
    writeln (S)
end.
```

TOPSHIRIQ

Misol 1. Har qanday ikkita sonni birini ikkinchisiga bo‘lganda qoldiq qismini olishni dasturini tuzing.

```
Program {qoldiqli butun bo‘lish};
Var i, k: integer;
Begin
Read (i, k);
Write (i mod k);
End.
```

Misol 2. R radiusli sharning hajmini hisoblash dasturini tuzing.

```
Program E 10 {Input, Output};
Const
Pi=3.14;
Var
R: Real; {Shar radiusi}
V: Real; {Shar hajmi}
Begin
Writeln ('R radius qiymatini kiriting: ');
Read (R);
V: =4*PI*R*R*R/3;
Writeln;
Writeln ('Natija: ');
Writeln ('SHar hajmi=', V: 8: 3);
End.
```

Misol 3. Parallelopiped qiralarining uzunliklari a,b,c berilgan bo‘lsa, uning hajmini va diogonalini hisoblash dasturini tuzing.

```
Var
    a, b, c, d, v: real
Begin
Write ('A='); Read ln (a);
Write ('B='); Read ln (b);
Write ('C='); Read ln (c);
V: =a*b*c;
D: =sqrt (sqr(a)+sqr(b)+sqr(c));
Write ('V=', V);
Write ('D=', D);
End.
```

TOPSHIRIQ

Tarmoqlanuvchi jarayonlarni dasturlashga doir misollar

1. Uchburchakning a,b,c tomlari berilgan. Uchburchak yuzasi topilsin.

```
Var
    a,b,c, p,s : real;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Begin

```
Write('Uchburchak tomonlarini kiriting:');  
readln(a); readln(b); readln(c);  
if (a>0) and (b>0) and (c>0) and (a+b>c) and (b+oa) and (a+ob) then  
  begin  
    p:=(a+b+c)/2;  
    s:=sqrt(p*(p-a)*(p-b)*(p-c));  
    writeln(s);  
  end  
else Writeln('Kiritilgan sonlar uchburchak tomonlari bo‘la olmaydi’);  
end.
```

2. x,y,z haqiqiy sonlari berilgan. Ularning maksimal topilsin.

var

x,y,z,max: real;

begin

write('x, y, z sonlarini kiriting);

readln(x,y,z);

if x>y then max:=x

else max:=y;

if z>max then max:=z;

writeln('max(x,y,z)=' ,max);

end.

3. x,y,z haqiqiy sonlari berilgan. Ushbu sonlarga mos uchburchak hosil qilish mumkinmi?

var

x,y,z,max,d: real;

begin

Write('Uchburchak tomonlarini kiriting:') readln(x,y,z);

if x>y then max:=x

else

max:=y;

if z>max then max:=z;

write('Учбурчак: ');

if $2*max < x+y+z$ then begin

d:=sqrt(x)+sqrt(y)+sqrt(z)-2*sqrt(max);

if d>0 then write('ўтқир бурчакли');

if d=0 then write('тўғри бурчакли');

if d<0 then write('ўтмас бурчакли')

end else write(' мавжуд эмас!');

end.

4. x,y,z haqiqiy sonlari berilgan. x+y+z va x*y*z miqdorlarning maksimali topilsin.

var

x,y,z,s,p: extended;

begin

write(' x, y, z sonlarni kiriting: '); readln(x,y,z);

s:=x+y+z;

p:=x*y*z;

if s>p then write(p, ' < ',s)

else if s<p then write(p, ' > ',s)

else write(s, ' = ',p);

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

end.

5. Haqiqiy a son berilgan. Quyidagi funktsiyaning nuqtadagi qiymati topilsin.

$$f(x) = \begin{cases} 0, & x < 0, \\ x^2 - x, & 0 < x \leq 1, \\ x^2 - \sin(\pi x^2), & x > 1 \end{cases}$$

var

x, f : real;

begin

write('Haqiqiy sonni kiriting:');

readln(x);

if x <= 0 then f := 0

else if x <= 1 then f := sqrt(x) - x

else f := sqrt(x) - sin(pi * x * x);

write('f(' , x, ') = ', f);

end.

6. Berilgan x haqiqiy sonning butun qismi topilsin.

var

x, y: real;

begin

write('x sonini kiriting: ');

readln(x);

if (x > 0) or (frac(x) = 0) then y := int(x)

else y := int(x) - 1;

writeln(['x, '], y); {butun qism}

writeln(['{x, '}, x - y); {kasr qism}

end.

7. Kvadrat tenglama yechimi topilsin.

var

a, b, c, d, x1, x2: real;

begin

write('a, b, c = ');

readln(a, b, c)

d := b * b - 4 * a * c;

if d > 0 then begin

x1 := (-b + sqrt(d)) / (2 * a);

x2 := (-b - sqrt(d)) / (2 * a);

writeln('x1 = ', x1, ' x2 = ', x2)

end

else if d = 0 then begin

x1 := -b / 2 * a;

writeln(x = ', x1)

end

else writeln('Yechim yo 'q!');

end.

Misol-8. $y = \frac{x^3 - 4x + 1}{|x| + 1}$ funktsiyaning x argumentini x_n dan x_k gacha h qadam bilan o'zgarishidagi hisoblash dasturini tuzing.

3.PROGRAM M (INPUT,OUTPUT);

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

LABEL 50;

VAR

X, U, XN, XK, Hx: REAL;

BEGIN

READ (XN, XK, HX);

X: =XN;

50: y: =(x*x*x-4*x+1)\(ABS(x)+1);

WRITELN (x: 4:2, ' ', y);

x: =x+HX;

IF x<=xk THEN GOTO 50 ;

END;

Misol-9.
$$z = \begin{cases} \ln \ln(x^3 + 3) - y, & \text{agar } y \geq 0 \\ 2x + 3y, & \text{akscholda} \end{cases} \quad y = (x + 1)e^{2x}; x = 1; -2$$

z ning qiymatini hisoblash algoritmi, blok-sxemasi va paskal dasturlash tilidagi dasturini tuzing.

3.PROGRAM M (INPUT,OUTPUT);

CONST PI=3,14

VAR Y, Z: REAL;

X: INTEGER;

BEGIN

X: INTEGER

READ (X);

Y: =(x+1)*EXP(PI*x);

IF y>=0 THEN

Z: =ln(ln(x*x*x+3))-y;

ELSE

Z: =2*x+3*y;

WRITELN ('z=', z:4:1);

END.

Topshiriq uchun namuna

No	Ifoda	Shart
31	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$X < 2$ $2 \leq x \leq 4$ $x > 4$

Dasturi

Program tarmoq;

Uses crt;

Var x,y:real;

begin

clrscr;

readln(x);

if x<2 then y:=sqrt(exp(2*x-1))+abs(cos(x))

else

if x>4 then y:=5.6+ln(x/(x+5))

else

y:=sqrt(sqr(x)+1)+abs(sin(x));

tekshirilmaydi}

writeln('x=',x:4:2,' da y=',y:4:2);

readln

end.

Dastur natijasi:

4

{kiritiladgan qiymat}

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

$x=4.00$ da $y=4.88$

{natija hosil bo‘ladi}

2) Quyidagi topshiriqlarni algoritm, blok-sxemasi va dasturini Paskal dasturlash tilida tuzing:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

1	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 1$ $1 \leq x \leq 2,5$ $x > 2,5$	11	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 4$ $4 \leq x \leq 5,5$ $x > 5,5$
2	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 2,1$ $2,1 < x \leq 4$ $x > 4$	12	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 5$ $x > 5$
3	$y = \begin{cases} 1 + \sin x^2 - 2 \\ \operatorname{tg}\sqrt{x^2 + 1} + \sin x \\ \operatorname{arcsin}(\cos(x^3 + 1)) \end{cases}$	$x \leq 1,5$ $1,5 \leq x \leq 10$ $x > 10$	13	$y = \begin{cases} 1 + \sin x^2 - 2 \\ \operatorname{tg}\sqrt{x^2 + 1} + \sin x \\ \operatorname{arcsin}(\cos(x^3 + 1)) \end{cases}$	$x \leq 2,5$ $2,5 \leq x \leq 12$ $x > 12$
4	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3 + b}} + \lg x^3 \\ e^x + \lg(ab + 1) \end{cases}$	$x \leq a$ $a < x \leq b$ $x > b$	14	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3 + b}} + \lg x^3 \\ e^x + \lg(ab + 1) \end{cases}$	$x \leq a$ $a < x \leq b$ $x > b$
5	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 2,1$ $2,1 < x \leq 4$ $x > 4$	15	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 6$ $x > 6$
6	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$x < 1$ $1 \leq x \leq 2$ $x > 2$	16	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$x < 2$ $2 \leq x \leq 4$ $x > 4$
8	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ \operatorname{tg}^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\cos x} \end{cases}$	$x \leq 2,6$ $2,6 \leq x < 6$ $x \geq 6$	17	$y = \begin{cases} e^{x^2-1} + \ln(x^2 + 1) \\ 0,25\ln(x+1) + \frac{1}{x+1} \\ \frac{1}{1+ \cos x } + \sqrt{\cos x} \end{cases}$	$x < 1$ $1 \leq x \leq 2$ $x > 2$
9	$y = \begin{cases} \operatorname{arcsin}\sqrt{x} + \cos x \\ e\sqrt{x^3 + ab} + \ln^2 x \\ \operatorname{tg}(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	$x < a$ $a \leq x \leq b$ $x > b$	18	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ \operatorname{tg}^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\cos x} \end{cases}$	$x \leq 3,6$ $3,6 \leq x < 7$ $x \geq 7$
10	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 3$ $3 \leq x \leq 4,5$ $x > 4,5$			

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

19	$y = \begin{cases} \arcsin\sqrt{x} + \cos x \\ e\sqrt{x^3 + ab} + \ln^2 x \\ tg(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	$x < a$ $a \leq x \leq b$ $x > b$	26	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$x < 2$ $2 \leq x \leq 4$ $x > 4$
20	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ tg(x^3 + 1) + \sqrt{\sin x} \\ arctg(\cos x) \end{cases}$	$x < 2$ $2 \leq x \leq 3,5$ $x > 3,5$	27	$y = \begin{cases} e^{x^2-1} + \ln(x^2 + 1) \\ 0,25\ln(x+1) + \frac{1}{x+1} \\ \frac{1}{1+ \cos x } + \sqrt{\cos x} \end{cases}$	$x < 0$ $0 \leq x \leq 2$ $x > 2$
21	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ tg(x^3 + 1) + \sqrt{\sin x} \\ arctg(\cos x) \end{cases}$	$x < 6$ $6 \leq x \leq 7,5$ $x > 7,5$	28	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ tg^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\cos x} \end{cases}$	$x \leq 4,6$ $4,6 \leq x < 8$ $x \geq 8$
22	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + tg(x^2 + 3,5) \\ 0,1 + arctg \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 6$ $x > 6$	29	$y = \begin{cases} \arcsin\sqrt{x} + \cos x \\ e\sqrt{x^3 + ab} + \ln^2 x \\ tg(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	$x < a$ $a \leq x \leq b$ $x > b$
23	$y = \begin{cases} 1 + \sin x^2 - 2 \\ tg\sqrt{x^2 + 1} + \sin x \\ \arcsin(\cos(x^3 + 1)) \end{cases}$	$x \leq 3,5$ $3,5 \leq x \leq 13$ $x > 13$	30	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ tg(x^3 + 1) + \sqrt{\sin x} \\ arctg(\cos x) \end{cases}$	$x < 2$ $2 \leq x \leq 4,5$ $x > 4,5$
24	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3+b}} + \lg x^3 \\ e^x + \lg(ab+1) \end{cases}$	$x \leq a$ $a < x \leq b$ $x > b$			
25	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + tg(x^2 + 3,5) \\ 0,1 + arctg \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 4,1$ $4,1 < x \leq 6$ $x > 6$			

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Topshiriq

1) Quyidagi funksiyani qiymatini hisoblash uchun algoritm va dastur tuzing.
 $Y = \cos \pi ax + \sin \pi ax$ bunda $\Delta X = 0,5$; qadam bilan $-3 \leq x \leq N$ gacha o‘zgaradi.

Dasturi

```
Program takror;  
Uses crt;  
Var  
  A,n,x,y:real;  
Begin  
  Clrscr;  
  Write('a=');Readln(a);  
  Write('x=-3, N ning qiymatini x ning boshlang`ich qiymatidan katta kiriting n=');Readln(n);  
  X:=-3;  
  Repeat                                     {sikl boshi}  
    Y:=cos(pi*a*x)+sin(pi*a*x);             {sikl tanasi ...}  
    Writeln('x=',x:3:1, ' y=',y:4:2);  
    X:=x+0.5;  
  Until x>N;                                {sikl oxiri}  
  Readln  
End.
```

Dastur natijasi:

a=0.1	{0.1 kiritilgan}
qiymat}	
x=-3, N ning qiymatini x ning boshlang`ich qiymatidan katta kiriting n=1	{1 kiritilgan}
qiymat}	
x=-3.0 y=-0.22	{natijalar}
x=-2.5 y=0.00	
x=-2.0 y=0.22	
x=-1.5 y=0.44	
x=-1.0 y=0.64	
x=-0.5 y=0.83	
x=0.0 y=1.00	
x=0.5 y=1.14	
x=1.0 y=1.26	

2) Yig‘indini hisoblash uchun algoritm va dastur tuzing.

$$S = \sum_{i=1}^N \frac{0,45}{\cos i + \sin i}$$

Dasturi

```
Program takror;  
Uses crt;  
Var  
  I,n:integer;   {sikl parametrlari doimo butun toifada bo‘lishi kerak, aks holda xato  
  bo‘ladi}  
  S:real;  
Begin  
  clrscr;  
  S:=0;  
  Write('Sikl sonini kiriting n= '); Readln(n);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```

For I:=1 to n do
    S:=S+0.45/(cos(i)+sin(i));
Writeln('S= ', S:6:4);
Readln
End.
    
```

{ sikl }
{ sikl tanasi }

Dastur natijasi:

Sikl sonini kiriting n= 100
S= 2.5267

{100 kiritilgan qiymat}
{natija}

3) Quyida berilgan cheksiz qatorni 0,005 aniqlikda hisoblash algoritmi va dasturi tuzilsin.

$$Z = \sum_{i=2}^{\infty} \frac{2,4+i}{i^i + 2}$$

Dasturi

```

Program takror;
Uses crt;
Var
    I:integer;
    Z,y:real;
Begin
    Clrscr;
    Z:=0; i:=2;
    Y:=(2.4+i)/(exp(i*ln(i))+2);
    While y>0.005 do
        Begin
            Y:=(2.4+i)/(exp(i*ln(i))+2);
            Z:=z+y;
            I:=i+1;
        End;
        Writeln(I-2, ' ta qadam yig'indi Z=',z:6:4);
        Readln
    End.
    
```

{ sikl }
{ sikl boshi }

{ sikl oxiri }

Dastur natijasi:

4 ta qadam yig'indi Z=0.9467

{ natija }

Topshiriq: a) Quyidagi topshiriqlar dasturini Paskal dasturlash tilida tuzing.

№	Funksiyani qiymatini hisoblash uchun algoritm va dastur tuzing.	Yig'indini hisoblash uchun algoritm va dastur tuzing.	Cheksiz qatorni 0,005 aniqlikda hisoblash algoritmi va dasturini tuzing
1	$Y = a^3 \sin x^2$ $\Delta X = 0.1 - 1 \leq X \leq N$	$S = \sum_{i=4}^N \frac{0.4}{i(i+1) + \sqrt{i}}$	$Z = \sum_{i=1}^{\infty} \frac{1}{i(i+1) + 2 \cos i}$
2	$Y = \sqrt{a} \cos x$ $\Delta X = 0.2; 5 \leq X \leq N$	$S = \sum_{i=1}^N \frac{\cos i}{i^2 + 5}$	$Z = \sum_{i=2}^{\infty} \frac{1}{i^2 + 5 + \ln i}$
3	$Y = a^3 \operatorname{tg} x$ $\Delta X = 0.1; 2 \leq X \leq N$	$S = \sum_{i=4}^N \frac{5 + e^j}{6j + 2,7}$	$Z = \sum_{i=1}^{\infty} \frac{5}{6i^4 + 2.7 + \sqrt{i}}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

4	$Y = 2 + a \ln x$ $\Delta X = 0.2; -3 \leq X \leq N$	$S = \sum_{i=5}^N \frac{1}{\ln i + 1.7}$	$Z = \sum_{i=3}^{\infty} \frac{1}{\ln i + i^2}$
5	$Y = 3.2a\sqrt{x+1}$ $\Delta X = 0.5; 1 < X < N$	$S = \sum_{k=1}^N \frac{1}{\sqrt{k} + 1.2}$	$Z = \sum_{K=1}^{\infty} \frac{4}{3k^3 + \sin K}$
6	$Y = 5e^{xa} + 2.3$ $\Delta X = 0.3; 0 \leq X \leq N$	$S = \sum_{i=1}^N \frac{1}{2e^i + 3.4}$	$Z = \sum_{i=1}^{\infty} \frac{1}{2e^i + tg^i}$
7	$Y = ax^3 + 2.4e^x$ $\Delta X = 0.4; -5 \leq X \leq N$	$S = \sum_{j=2}^N \frac{\sqrt{5.4}}{3j^3 + j^{\frac{1}{3}}}$	$Z = \sum_{K=1}^{\infty} \frac{4}{3k^3 + \sin K}$
8	$Y = e^{ax} + 2.1x$ $\Delta X = 0.2; -2 \leq x \leq N$	$S = \sum_{j=5}^N \frac{1}{1 + \ln^2 j}$	$Z = \sum_{J=1}^{\infty} \frac{8}{\sqrt{J} + J^4}$
9	$Y = tgax + \sqrt{x}$ $\Delta X = 0.3; -0.3 \leq X \leq N$	$S = \sum_{L=2}^N \frac{1 + \sqrt{5}}{L^2}$	$Z = \sum_{J=1}^{\infty} \frac{\ln j}{j^6}$
10	$Y = e^{ax} - e^{ x }$ $\Delta X = 0.5; -0.5 \leq x \leq N$	$S = \sum_{i=1}^N \frac{\cos^2 i}{i^3}$	$Z = \sum_{i=1}^{\infty} \frac{3.4}{e^{2i}}$
11	$Y = \frac{a + 0.27}{x + \sqrt{1+x}}$ $\Delta X = 0.8; -0.9 \leq X \leq N$	$S = \sum_{i=1}^N \frac{(-1)^i}{i + tgi}$	$Z = \sum_{i=3}^{\infty} \frac{1}{i^i + 2.7}$
12	$Y = \frac{\sqrt{1+x+0.3}}{a^2x}$ $\Delta X = 0.3; 0 \leq x \leq N$	$S = \sum_{i=3}^N \frac{3}{(-1)^i + 2i^2}$	$Z = \sum_{j=1}^{\infty} \frac{\sqrt{j}}{1 + j^6}$
13	$Y = \sin ax + e^a$ $\Delta X = 0.9; -2 \leq X \leq N$	$S = \sum_{i=1}^N \frac{35}{\sqrt{i} + \sqrt{i}}$	$Z = \sum_{i=2}^{\infty} \frac{3}{i + 2\sqrt{i}}$
14	$Y = a + e^x + \cos x$ $\Delta X = 0.4; -0.8 \leq x \leq N$	$S = \sum_{i=2}^N \frac{e^i + 1}{i}$	$Z = \sum_{K=1}^{\infty} \frac{k}{e^k + k^2}$
15	$Y = a^2\sqrt{\sqrt{x+1}} + 2$ $\Delta X = 0.4; 0.3 \leq X \leq N$	$S = \sum_{i=1}^N \frac{1}{ \cos i \cdot i}$	$Z = \sum_{j=1}^{\infty} \frac{0.7}{2 + \ln k }$
16	$Y = \cos ax + \sin a$ $\Delta X = 0.3; 0.1 \leq x \leq N$	$S = \sum_{i=5}^N \frac{1}{\cos i + \sin i}$	$Z = \sum_{i=1}^{\infty} \frac{67}{i^4 + \sqrt{3i}}$
17	$Y = 2.7a^4 + \ln x^2$ $\Delta X = 0.2; -1 \leq X \leq N$	$S = \sum_{j=1}^N \frac{2.7}{\ln j^3 + 1}$	$Z = \sum_{i=6}^{\infty} \frac{ \cos i }{i^2}$
18	$Y = 35.9a^4 \sin x^2$ $\Delta X = 0.1; 0 \leq x \leq N$	$S = \sum_{k=3}^N \frac{k + 0.4}{\sqrt{k} + \sqrt{k}}$	$Z = \sum_{j=1}^{\infty} \frac{0.4}{\sqrt{j+100}}$
19	$Y = \sqrt{a+2} \ln x^3$ $\Delta X = 0.4; 0.02 \leq X \leq N$	$S = \sum_{k=1}^N \frac{0.999k}{k^2 + k^{0.7}}$	$Z = \sum_{i=3}^{\infty} \frac{2.4}{ i + e^{i+1}}$
20	$Y = \sqrt{a+2.1} \ln x^3$ $\Delta X = 0.4; 0.02 \leq x \leq N$	$S = \sum_{k=1}^N \frac{k+1}{k^2+2}$	$Z = \sum_{L=1}^{\infty} \frac{100}{L^2 + \ln L }$
21	$Y = e^a tgx + 3$ $\Delta X = 0.3; -0.05 \leq X \leq N$	$S = \sum_{i=1}^N \frac{0.4i}{i \cos i + 3i}$	$Z = \sum_{k=1}^{\infty} \frac{ \cos k + \sin k }{k}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

22	$Y = e^{ax} + 2,91x$ $\Delta X = 0,1; 1 \leq x \leq N$	$S = \sum_{i=2}^N \frac{\sqrt{i+2}}{3i}$	$Z = \sum_{j=1}^{\infty} \frac{0,9}{e^{j+2} + 1}$
23	$Y = \cos^2(2x+a)$ $\Delta X = 0,2; 0,04 \leq X \leq N$	$S = \sum_{k=1}^N \frac{\sqrt{ \cos k }}{k}$	$Z = \sum_{i=1}^{\infty} \frac{0,4}{\sqrt{i^3 + 1}}$
24	$Y = 3\ln^2(ax^2 + 3)$ $\Delta X = 0,04; 0,4 \leq x \leq N$	$S = \sum_{j=1}^N \frac{0,9}{\sqrt{j^2 + \sqrt{1+j}}}$	$Z = \sum_{i=5}^{\infty} \frac{e^{0,4}}{\ln i^2}$
25	$Y = 0,2ax^2 \operatorname{tg} \frac{x}{a}$ $\Delta X = 0,1; 0,1 \leq X \leq N$	$S = \sum_{k=1}^N \frac{k^{\frac{1}{3}}}{k^3 + \sin k}$	$Z = \sum_{i=1}^{\infty} \frac{2,4+i}{i^i + 2}$
26	$Y = ax^2 + \sin \pi x$ $\Delta X = 0,03; 0,03 \leq x \leq N$	$S = \sum_{k=2}^N \frac{(-1)^{k+1}}{k + \sqrt{0,2k}}$	$Z = \sum_{k=1}^{\infty} \frac{k}{\ln k^5}$
27	$Y = \frac{ax^2 3x + 1}{x}$ $\Delta X = 0,04; 1 \leq X \leq N$	$S = \sum_{i=1}^N \frac{i(i+2)}{i + \sqrt{i+1}}$	$Z = \sum_{k=3}^{\infty} \frac{0,5k}{k^3 + e^k}$
28	$Y = \frac{\sin x}{a\pi x} + 0,4$ $\Delta X = 0,4; 0,08 \leq X \leq N$	$S = \sum_{i=1}^N \frac{\sin(\pi i)}{i^2}$	$Z = \sum_{j=5}^{\infty} \frac{0,9}{e^{j+3} + 3,4}$
29	$Y = a^{\frac{1}{6}} e^{x^2}$ $\Delta X = 0,25; -2 \leq x \leq N$	$S = \sum_{i=3}^N \frac{\cos(\pi i)}{\pi i}$	$Z = \sum_{k=1}^{\infty} \frac{99k}{k^{\frac{1}{3}} + \sqrt{k+k^4}}$
30	$Y = \cos \pi ax + \sin \pi ax$ $\Delta X = 0,5; -3 \leq x \leq N$	$S = \sum_{i=1}^N \frac{0,45}{\cos i + \sin i}$	$Z = \sum_{k=1}^{\infty} \frac{0,49}{\sqrt[4]{k+0,27+k}}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

7-AMALIY MASHG‘ULOT

Mavzu: Pascalda massivlar.

Mashg‘ulotning maqsadi: Talabalarda Paskal dasturlash tilida massivlar va to‘plamlar bilan ishlash bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

Qisqacha nazariy ma‘lumot

Massivlar umumiy holda quyidagicha tavsiflanadi:

ARRAY[<indeks toifasi>] OF ARRAY [<indeks toifasi>] OF <elementlar toifasi>; yoki
ARRAY[<indeks toifasi, indeks toifasi>] OF <elementlar toifasi>;

Masalan, Bir o‘lchamli massiv:

Type mas1=array[1..100] of real;

Var a1:mas;

yoki

Var a1:array[1..100] of real;

ko‘rinishda, ko‘p o‘lchamli massiv:

Type mas2=array[1..10,1..10] of real;

Var a2:mas;

yoki

Var a2:array[1..10,1..10] of real;

ko‘rinishda tavsiflanadi.

Dastur ichida a1[1]:=5; a2[2,3]:=10; shaklda foydalanish mumkin.

To‘plamlar ustida amallar

1. Ikkita A va V to‘plamlari teng (bir xil) deyiladi, agarda ularning barcha hadlari o‘zaro bir xil bo‘lsa (A=V).

Misol: A=[4,1,3], V=[4,1,3].

2. A to‘plam V to‘plamining to‘plam ostisi deyiladi, agarda A ning barcha hadlari V ning ham hadlari bo‘lsa (A<V).

Misol: A=[1, 2, 3], B=[1, 2, 3, 4, 5, 6].

3. A va V to‘plamlarining kesishmasidan yana to‘plam hosil bo‘ladi va natijaviy to‘plamning hadlari A to‘plamga ham, V to‘plamga ham tegishli bo‘ladi (S=A*V).

Misol: [1,2,3,4,5] * [2,5,6,7,8] = [2,5]. Agar A va V to‘plamlari bir xil hadlarga ega bo‘lmasa, S = A*V natija – bo‘sh to‘plam bo‘ladi S = [].

4. A va V to‘plamlarning birlashmasi (S=A+V) ularning hech bo‘lmasa birortasiga tegishli bo‘lgan hadlardan tashkil topadi.

Misol: [1,2,3,4,5] + [2,5,6,7,8] = [1,2,3,4,5,6,7,8].

5. A to‘plamdan V to‘plamni ayirmasi deb, shunday to‘plamga aytiladiki, natijaviy to‘plamning hadlari A to‘plamga tegishli lekin V to‘plamga tegishli bo‘lmaydi (S = A - V).

Misol: [1,2,3,4,5] - [2,5,6,7,8] = [1,3,4].

6. x qiymatni A to‘plamiga tegishligini aniqlash uchun «in» amaldan foydalaniladi. x **in** A = **true**, agar x qiymat A to‘plamiga tegishli bo‘lsa; x **in** A = **false**, agar x qiymat A to‘plamiga tegishli bo‘lmasa.

Misol: A=[2, 3, 4, 7] bo‘lsa, 6 **in** A ifodasining natijasi **false**; 3 **in** A ifodani natijasi esa **True**.
taylik.

Kiritilgan qatorning faqat son, lotin harflari va bo‘sh joylardan tashkil topganligini aniqlash dasturini tuzing.

Program Tuplamlar;

Var

Str:string;

L:Byte;

Tru:Boolean;

Begin

Writeln(‘Qatorni kiriting’);

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Readln(Str);
L:=Length(Str);           {Kiritilgan simvollar soni}
Tru:=L>0;                 {True, agar bo‘sh qator bo‘lmasa}
While Tru and (L>0) do   {Qator oxirigacha tekshirish }
Begin
    Tru:=Str[L] in ['0'..'9', 'A'..'Z', 'a'..'z', ' '];
    {Simvollar to‘g‘riligini tekshirish }
    Dec(L);                {Oldingi simvol }
End;
If Tru Then Writeln('To‘g‘ri yozilgan qator')
Else Writeln('Noto‘g‘ri yozilgan qator');
```

End.

Quyida faqat haqiqiy sonlarni qabul qila oladigan funksiyani yaratish dasturi keltirilgan:

```
Program Set_Of;
Uses Crt;
Var
    R:Real;
Function Input_R:Real;   {Haqiqiy sonlarni kiritish funksiyasi}
Var
    S:String[15];        {S o‘zgaruvchiga ko‘pi bilan 15 ta belgi sig‘adi}
    S1:Set Of Char;     {Belgilarni tekshirish uchun to‘plam}
    Ch:Char;            {Belgilarni qabul qiladigan o‘zgaruvchi}
    Code:Integer;
    R1:Real;
Begin
    S:='';
    S1:['0'..'9','.',',','-'];
    Repeat
        Ch:=Readkey;
        If Ch in S1 Then {Ch ga kiritilgan belgini S1 to‘plam ichidan tekshirish}
        Begin
            S:=S+Ch;
            Write(Ch);
        End;
    Until Ch=#13;       {Enter klavishi bosilguncha sikl aylanadi}
    Val(S,R1,Code);
    Input_R:=R1;
End;
Begin
    R:=Input_R;        {Input_R funksiyadan foydalanish}
    Writeln;
    Writeln('1-natija= ',R:10:4);
    Writeln('2-natija= ',Input_R:10:4); {Input_R funksiyadan foydalanish}
    Readln;
End.
```

TOPSHIRIQ

1-misol. Bir o‘lchamli, n ta hadli (n=30) massiv hadlarini yig‘indisini hisoblash dasturi.

```
Program L1;
const n=30;
var
    i: integer;
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
x: array [1..n] of real;
S: real;
begin
  for i:=1 to n do readln (x[i]); { massiv hadlarini
                                kiritish}
  S:=0;
  for i:=1 to n do S:=S+x[i];
  writeln ('natija=', S)
end.
```

2-misol. Bir o‘lchamli, n ta hadli (n=30) massiv hadlarining eng kattasini topish va uning joylashgan joyini aniqlash.

```
Program L2;
const n=30;
type
  gran = 1..30;
  vector = array [gran] of real;
var
  x: vector;
  S: real;
  i, k: integer;
begin
  writeln (' x - massivi hadlarini kiriting');
  for i: =1 to n do readln (x[i]);
  S: =x[1]; k: =1;
  for i: =2 to n do
    if x[i] > S then
      begin
        S: =x[i]; k: =i
      end;
  writeln ('x massivininig eng katta hadi');
  writeln (S);
  writeln ('max(x) ning o‘rni', k)
end.
```

3-misol. n ta hadli (n = 15) vektorlarning skalyar ko‘paytmasini aniqlash.

```
Program L3;
const n=15;
type
  gran = 1..n;
  mas = array [gran] of real;
var
  i: byte;
  S: real;
  x, y: mas;
begin
  writeln ('x va u massiv hadlarini kiriting');
  for i: =1 to n do readln (x[i]);
  for i: =1 to n do readln (y[i]);
  S:=0;
  for i:=1 to n do S:=S + x[i] * y[i];
  writeln ('natija=', S)
end.
```

4-misol. Matritsalarini qo‘shish.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Program L2;
const n = 3; m = 4;
    { n - matritsa satrlari soni,
      m - ustunlar soni }
var   i, j: integer;
      A, B, C: array [1..n, 1..m] of real;

begin {A, V matritsa hadlarini kiritish}
    for i := 1 to n do
        for j := 1 to m do
            readln (A[i,j], B[i,j]);
        for i := 1 to n do
            for j := 1 to m do
                begin
                    C[i,j] := A[i,j] + B[i,j];
                    writeln (C[i,j])
                end
            end
        end.
end.
```

5-misol. Matritsani vektorga ko‘paytirish.

```
Program L3;
const n = 3; m = 4;
type matr = array [1..n, 1..m] of real;
     vect = array [1..m] of real;
var   i, j: byte;
      A: matr;
      B, C: vect;

begin
    writeln ('A matritsa hadlarini kiriting');
    for i := 1 to n do
        for j := 1 to m do
            readln (A[i,j]);
        writeln ('B vektor hadlarini kiriting');
        for i := 1 to n do readln (B[i]);
        for i := 1 to n do
            begin
                C[i] := 0;
                for j := 1 to m do
                    C[i] := C[i] + A[i,j] * B[j];
                writeln (C[i]);
            end end.
end end.
```

6-misol. Matritsa hadlarining eng kattasini topish va uning joylashgan joyini aniqlash.

```
Program L4;
const n = 3; m = 4;
var   A: array [1..n, 1..m] of real;
      R: real;
      i, j: byte; K, L: byte;

begin {A matritsa hadlarini kiritish}
    for i := 1 to n do
        for j := 1 to m do
            readln (A[i,j]);
        R := A[1,1]; L := 1; K := 1;

```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
for i: =1 to n do
  for j: =1 to m do
    begin
      if R < A[i,j] then
        begin
          R: =A[i,j];
          L: = i; K: =j;
        end;
      end;
    writeln ('max A=', R);
    writeln ('satr=',L,'ustun =',K);
  end.
```

Topshiriq

1. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa elementlarining o‘rtacha qiymati topilsin. Massivda o‘rtacha qiymatdan kichik va katta elementlari son topilsin.
2. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa ustunlarida 4-ga karrali element bor yo‘qligini aniqlansin. Agar bor bo‘lsa ustunga mos $C(n)$ massiv elementiga 1 qiymat aks holda 0 qiymat berilsin, yani $C(n)$ massiv elementlari qiymati topilsin.
3. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa ustunlari 2-ga karralilarining yig‘indisini mos $B(n)$ massiv elementiga joylashtirilsin.
4. Haqiqiy elementlardan tashkil topgan kvadrat matrisa $A = \{a_{ij}\}$ berilgan. Nol va birlardan tashkil topgan b_1, b_2, \dots, b_n ketma-ketlik tuzilsin, unda $b_i = 1$ agar matrisa satrlarida hech bo‘lmaganda 1 bilan 10 orasida yotuvchi bitta element mavjud bo‘lsa.
5. Haqiqiy elementlardan tahkil topgan kvadrat matrisa berilgan. Nol va birlardan tashkil topgan b_1, b_2, \dots, b_n ketma-ketlik tuzilsin, unda $b_i = 1$ agar matrisa ustunlarida hech bo‘lmaganda bitta 2-ga karrali element mavjud bo‘lsa.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

8-AMALIY MASHG‘ULOT

MAVZU: PASKALDA SATRIY KATTALIKLAR BILAN ISHLASH.

Mashg‘ulotning maqsadi: Talabalarga simvolli kattaliklar va satrli ma’lumotlar bilan ishlash bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

NAZARIY QISM

1-Misol. Kiritilgan ixtiyoriy jumla tarkibidagi “A” belgisini “O” belgisiga almashtirish chop etuvchi dastur tuzing.

1-usul	2-usul
<pre>PROGRAM sim1; VAR jumla:string;i,d: INTEGER; SIM:STRING; BEGIN read(Jumla);d:=length(jumla); writeln(d); FOR I:=1 TO D DO BEGIN IF COPY(JUMLA,I,1)='A' THEN JUMLA[I]:='O'; END; WRITELN(JUMLA); readln; END.</pre>	<pre>PROGRAM sim1; VAR jumla:string;i,d: INTEGER; SIM:STRING; BEGIN read(Jumla);d:=ord(jumla[0]); writeln(d); FOR I:=1 TO D DO BEGIN IF JUMLA[I]='A' THEN JUMLA[I]:='O'; END; WRITELN(JUMLA); readln; END.</pre>

2-Misol. Burchak qiymati gradus o‘lchovida kiritilgan holat uchun sinus, kosinus,tangens funksiyalari qiymatini mos belgini tanlash yo‘li bilan hisoblovchi dastur tuzing.

```
PROGRAM char1;
uses crt;
VAR sim:char; i,grad:INTEGER; javob,radian:real; label 15,20;
BEGIN
writeln('burchak qiymatini gradus o‘lchovida kiriting');
readln(grad); radian:=(pi*grad)/180;
writeln('sinus ', grad,' ni hisoblash uchun "s" tugmasini tanlang');
writeln('kosinus ', grad,' ni hisoblash uchun "c" tugmasini tanlang');
writeln('tangens ', grad,' ni hisoblash uchun "t" tugmasini tanlang');
sim:=readkey;
case sim of
's': javob:=sin(radian);
'c': javob:=cos(radian);
't': if cos(radian)=0 then goto 15 else javob:=sin(radian)/cos(radian);
end;
WRITELN(javob:5:2);goto 20;
15:writeln ('qiymatimavjud emas');
20: readln;
END.
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

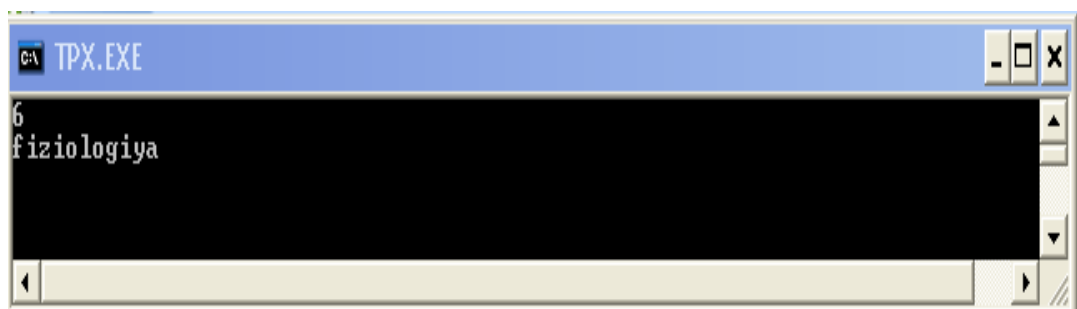
3-Misol. Tarkibida raqamlar bo‘lgan s1-satriy kattalikning barcha raqamlaridan yangi s2 –satriy kattalik yaratuvchi dastur tuzing.

```
Program Satriy_kattaliklar;  
Var s1, s2: string; i: byte;  
begin  
  writeln("Tarkibida raqamlar bo‘lgan satriy kattalikni kiriting");  
  readln(s1);  
  s2:= "";  
  for i:=1 to length(s1) do  
    if (s1[i]>='0') and (s1[i]<='9')  
      then s2:=s2+s1[i];  
  writeln('natija ',s2);  
end.
```

4-Misol.

```
uses crt;  
var  
s1,s2,s3,s4,s5:string;  
i:integer;  
begin  clrscr;  
s1:='SAMARQAND';  
i:=pos('QAND',s1);  
writeln(i);  
s2:='biologiya';  
s3:='fizika';  
delete(s2,1,2);  
delete(s3,5,2);  
insert(s2,s3,5);  
writeln(s3);  
readln;  
  
end.
```

Dastur natijasi:



TOPSHIRIQ

1-misol Har bir harfni “Enter” tugmasi yordamida chiqaruvchi dastur

```
program klm;  
var  
str:string[10];  
i:integer;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
begin
str:='uzbekiston';
for i:=1 to 10 do
begin
readln;
writeln(i,'-harf ', str[i]);
end;
readln;
end.
```

2-misol. Matnli satrlarni birlashtiruvchi dastur

```
program klm;
var
ch:char;
i:integer;
s,str:string;
begin
write('ismini kiriting ');
readln(str);
write('jinsini qaniqa(1/0) ');
readln(ch);
if ch='1' then
s:=concat(str,'jon')
else
s:=concat(str,'xon');
writeln(s);
end.
```

3-misol So`zda harflar sonini aniqlovchi dastur

```
program klm;
var
str:string;
begin
write('suzni kiriting ');
readln(str);
writeln('suzda ',length(str), ' ta harf mavjud'); end.
```

4-misol. Kichik simvolni katta simvolga aylantirish

```
program klm;
var
s:string;
i:integer;
begin
write('suzni kiriting ');
readln(s);
for i:=1 to length(s) do
s[i]:=upcase(s[i]);
writeln(s);
readln end.
```

5-misol matnning kerakli qismini qirqib olish.

```
program klm;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
var
s,str:string;
m,n:integer;
begin
  write('suzni kiriting ');
  readln(s);
  write('fragment boshini kirit: ');
  readln(n);
  write('nusxa sonini kiriting: ');
  readln(m);
  str:=copy(s,n,m);
  writeln(str);
end.
```

Mustaqil bajarish uchun amaliyot topshiriqlari

1. Biror soʻzda qatnashuvchi belgilar sonini aniqlash va ustun koʻrinishida chop etuvchi dastur tuzing
2. Berilgan literli kattalikdagi “a” va “b” harflar soni yigʻindisini hisoblovchi dastur tuzing.
3. Berilgan literli kattalikdagi hamma “a” harflarini “b” harfga almashtiruvchi dastur tuzing.
4. Berilgan matndagi har bir harfni ikkilantiruvchi dastur tuzing
5. NS soʻzi KS soʻzida necha marta uchrashini aniqlash dasturi tuzilsin.
6. “bol” soʻzi berilgan soʻzning qismligini aniqlash dasturi tuzilsin. Javob “ha” yoki “yoʻq” boʻlishi kerak. Misol uchun “futbol” va “bolalar” soʻzlari uchun “ha”, “kitob” soʻzi uchun “yoʻq”.
7. Berilgan gapda uchraydigan xar bir soʻzning «dan» qoʻshimchasini «ning» qoʻshimchasi bilan almashtiring.
8. Satrda uchraydigan barcha ‘+’ belgilari sonini aniqlash va ‘-’ belgisiga oʻzgartirish dasturini tuzing.
9. Berilgan gap nechta soʻzdan iborat ekanligini topish dasturini tuzing.
10. Kiritilgan soʻzni teskarisiga oʻgirib beruvchi dastur tuzing.
11. Berilgan soʻzning belgilari orasiga bittadan probel chiquvchi dastur tuzing.
12. Berilgan satrdagi barcha “a” harflarini oʻchirib beruvchi dastur tuzing.
13. Berilgan butun sonning raqamlari koʻpaymasini hisoblovchi dastur tuzing.
14. Ixtiyoriy kiritilgan s satriy kattalikni teskarilovchi dastur tuzing.
15. Berilgan satriy kattalik tarkibidagi 'A' belgisini 'B' belgiga almashtiruvchi dastur tuzing.
16. Ixtiyoriy berilgan jumla tarkibidagi soʻzlar sonini aniqlovchi dastur tuzing.
17. Ixtiyoriy koʻp xonali son tarkibiga 6 raqamining ishtirok etish sonini aniqlovchi dastur tuzing.
18. Ixtiyoriy jumla kiritish natijasida undagi har bir belgisi sonini aniqlovchi dastur tuzing.
19. Satriy kattalik bir necha soʻzdan iborat. Undagi har bir soʻzning bosh harfini berilgan simvolga oʻzgartirish dasturini tuzing.
20. Kiritilgan har bir soʻz probellar bilan ajratilgan. Satrdagi birinchi k ta soʻzni oʻchirish dasturini tuzing.
21. Berilgan satriy kattalikdagi matnning barcha 'dan' qoʻshimchalarini qoʻshimchalarini 'da' qoʻshimchasiga oʻzgartirish dasturini tuzing.
22. Berilgan satriy kattalikdagi matnning barcha soʻzlarining uzunligini hisoblovchi dasturini tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

9,10-AMALIY MASHG‘ULOT

Mavzu: QISM DASTURLAR-FUNKSIYALAR VA PROTSEDURALAR VA ULARGA DOIR DASTURLAR TUZISH.

Mashg‘ulotning maqsadi: Talabalarga funksiya va proseduralar bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

NAZARIY QISM

Parametrlri protseduralar

Protsedura bilan asosiy dasturni bog‘laydigan asosiy faktor bu – protsedura parametrlaridir. Parametrlarni ikkita turga ajratiladi: qiymatli parametrlar (parametr-qiymat), o‘zgaruvchili parametrlar (para-metr - o‘zgaruvchi).

Parametr - qiymat bu protsedurani ishlash jarayonini ta‘minlovchi parametrlar hisoblanadi, ya‘ni asosiy dastur qiymatlarini protseduraga uzatadigan parametrlardir.

Endi, yuqorida ko‘rib chiqilgan sonlarni eng kattasini topish algoritmining dasturcini qiymatli parametr bilan yozilgan protseduralar orqali amalga oshiraylik:

```
Program max;  
var x, y, u, v: real; S: real;  
procedure max2 ( a, b: real );  
begin  
    if a>b then S:=a else S:=b;  
end;  
begin  
    read (x, y);  
    max2 (x + y, x * y); u:=S;  
    max2 (0.5 , u); v:=S;  
    writeln (u, v)  
end.
```

bu yerda a, b - protseduraning qiymatli formal parametrlari.

1- Misol: $f(n) = n!$ ($n! = 1 * 2 * 3 * \dots * n$ - faktorial) funksiyadan foydalanib, - ifodani hisoblashni tashkil qiling:

```
Program L1;  
var  
    k, m, i :integer; y: real;  
function fact (n: integer): integer;  
var  
    j: integer; P: byte;  
begin  
    j:q1;  
    for p:=1 to n do j:=j * p;  
    fact :=j;  
end;  
begin  
    readln (k, m);  
    y:=(fact (20) + fact(3))/(fact(5) + fact (31)) * fact(k+1) / fact(m);  
    writeln( y);  
end.
```

Funksiyalarni aniqlashda doim shunday harakat qilish lozimki, uning tana qismida formal parametrlar va funksiyani aniqlash uchun zarur bo‘lgan lokal o‘zgaruvchilargina qatnashsin. Dasturning global o‘zgaruvchisiga iloji boricha protsedura yoki funksiya ichidan turib qiymat bermaslik kerak, aks holda dastur xato natija berishi mumkin.

Misol: Program m1;

```
Var x,y: integer;  
Funktion f(t: integer): integer;
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
Begin  
  f:=t*t;  
  x:=7;  
end;  
begin  
  x:=5; writeln(x);  
  y:=f(2)+x  
  writeln(x,y)  
end.
```

Bu dasturning ishlashi natijasida $x=5$, $y=11$ va $x=7$ qiymatlar ekranga chiqariladi, ya'ni funksiyaning ichki qismidagi $x=7$ qiymati asosiy dasturdagi natijaviy qiymatlarga o'z ta'sirini o'tkazmoqda.

Rekursiv funksiyalar

Paskal tilida protsedura – funksiyalar bilan ishlashda, funksiyalarning rekursivlik xossasidan foydalanish imkoniyati yaratilgan.

Rekursiya tushunchasiga misol qilib oddiy faktorial hisoblashni keltirish mumkin:

$$y = \int_b^a f(x)dx$$

bu yerda ko'rinib turibdiki $n!$ qiymati $(n-1)!$ orqali aniqlanayapti, ya'ni rekursiya degani o'zi orqali o'zini aniqlash ma'nosini anglatadi.

Paskal tili ham funksiyalarni rekursiv aniqlash imkoniyatini beradi. Funksiyani rekursiv aniqlash uning tana qismida o'ziga - o'zi murojaat qilish orqali amalga oshiriladi.

Yuqoridagi faktorial hisoblashni rekursiv funksiyalar orqali amalga oshiraylik:

```
program L1;  
var  
  n: integer; y: integer;  
function fact(m: integer): integer;  
var  
  k: integer;  
begin  
  if m=0 then fact:=1 else fact:=fact(m-1) * m;  
end;  
begin  
  readln (n);  
  y:= fact (n);  
  writeln(y);  
end.
```

TOPSHIRIQ

1-Misol: $u = \max(x + y, x * y)$, $v = \max(0.5, u)$ – berilgan x va y haqiqiy sonlardan foydalanib u va v qiymatlarni aniqlash. Bu yerda x , u - qiymatlari kiritiladigan haqiqiy turli o'zgaruvchilar.

1. Masalani yechish dasturining protseduradan foydalanmay tuzilgan holi:

```
Program max;  
var  
  x, y, u, v: real;  
  a, b, s: real;  
begin  
  {x, u - miqdorlarni kiritish};  
  readln (x,y);  
  a:= x +y; b:= x*y;  
  if a > b then S:= a else S:=b;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
u := S;  
a:= 0.5; b:=u;  
if a > b then S:= a else S:=b;  
v:=S;  
{olingan natijalar};  
writeln (u, v)
```

end.

Ahamiyat bersangiz, dasturdagi shartli operator ikki marta takrorlanib, bir xil ish bajardi.

2. Masalani yechish dasturini parametrsiz protseduradan foydalanib tuzilgan holi (endi yuqoridagi dasturda yo‘l qo‘yilgan kamchilikni protseduralar orqali tuzatishga harakat qilamiz):

Program max;

var x, y, u, v: real; a, b, S: real;

procedure max1;

begin if a>b then S:=a else S:=b;

end;

begin readln (x, y);

*a:= x + y; b:=x * y;*

max1; {max1 protsedurasiga 1-marta murojaat qilinmoqda}

u:=S;

a:=0.5; b:= u;

max1; {max1 protsedurasiga 2-marta murojaat qilinmoqda}

v:=S;

writeln (u,v); end.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

11-AMALIY MASHG‘LOT

MAVZU: PASCALDA FAYLLAR BILAN ISHLASH

Mashg‘ulotning maqsadi: Talabalarga Pascal dasturlash tilida fayllar bilan ishlash bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

Nazariy qism:

Faylli turlarni hosil qilish

Faylli turdagi o‘zgaruvchilarni diskdan ma’lumot o‘qib oluvchi yoki diskka ma’lumot yozib qo‘yuvchi dasturlarda ishlatish mumkin. Faylli turdagi o‘zgaruvchilarni e’lon qilishda *file* va *text* xizmatchi so‘zlari ishlatiladi:

```
var   mfile 1, mfile 2: file;  
      afile: file;  
      Prima: text;
```

text xizmatchi so‘zi faylning matnli ekanligini anglatadi. Matnli fayllar maxsus belgilar bilan ajratilgan, uzunligi noma’lum bo‘lgan qatorlardan tashkil topadi.

Ayrim paytlarda fayllarni bir xil turli hadlar ketma-ketligi ko‘rinishida qarash qulayrok bo‘ladi. Bu ketma-ketlik qatorlar, butun sonlar yoki yozuvlardan tashkil topishi ham mumkin:

```
var   A1: file of byte;  
      {A1 fayli baytlar ketma - ketligidan tashkil topgan}  
      A2: file of integer;  
      {A2 fayli butun sonlar ketma-ketligidan tashkil topgan}  
      A3: file of string;  
      {A3 fayli katorlar ketma-ketligidan tashkil topgan}  
      A4: file of string[20];  
      {A4 fayli 20ta belgili qatorlarning ketma-ketligidan tashkil topgan}  
      A5: text;  
      {A5 fayli matnli fayl hisoblanadi}
```

Agar faylning hadlari uchun tur aniqlangan bo‘lsa, bunday fayllarni turlashtirilgan, aks holda turlashtirilmagan deb ataladi:

```
var   A: file ; { turlashtirilmagan fayl}  
      B: file of char; { turlashtirilgan fayl}
```

Fayllar bilan ishlaydigan quyidagi dasturni ko‘rib chiqaylik.

Var

```
mydata: file of integer;  
i, j, sum: integer;
```

begin

```
assign (mydata, 'd:\tp\myfile.dat');  
      {mydata fayl o‘zgaruvchisi bilan faylning  
      ismini myfile.dat va uning aniq yo‘li  
      aniqlanmoqda}
```

```
rewrite (mydata);      {fayl yozuv uchun ochiq}
```

```
writeln ('Salom noma’lum o‘rtoq...');
```

```
writeln ('Birinchi sonni kiriting');
```

```
readln (i);
```

```
writeln ('Kiritilgan sonni diskdagi myfile.dat  
fayliga yozilmokda');
```

```
write (mydata, i); {bu operator yordamida  
diskdagi myfile.dat fayliga i sonini yoziladi}
```

```
writeln ('Ikkinchi sonni kiriting');
```

```
readln (j);
```

```
writeln ('Kiritilgan ikkinchi sonni diskdagi myfile.dat  
fayliga yozilmoqda');
```

```
write (mydata, j);      {Diskka yozish bajarilmoqda}
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
sum := i + j;  
writeln ('Yig'indi =', sum);  
writeln ('Yig'indi diskdagi myfile.dat  
        fayliga yozilmokda');  
write (mydata, sum);           {Diskka yozish bajarilmoqda}  
close (mydata);                {mydata fayli yopildi}  
writeln ('Xayr noma'lum o'rtoq...');
```

end.

2. Faylli turlar bilan ishlash protseduralari va ularning vazifalari

E'tiboringizga havola etilgan dasturda *Assign*, *Rewrite*, *Write* va *Close* protseduralaridan foydalanildi. Endi shu protseduralarning va keyingi dasturda ishlatiluvchi *Reset* va *Read* protseduralarning vazifalari va qanday aniqlanganligi haqida qisqacha ma'lumot berib o'taylik:

Assign protsedurasi.

Vazifasi:

Faylli o'zgaruvchiga tashqi fayl ismini o'zlashtiradi.

Assign (f; name: string);

bu yerda f - ixtiyoriy turli faylli o'zgaruvchi;

Aniqlanishi:

name - qatorli turdagi ifoda yoki qator, fayl ismi (agar faylning to'liq yo'li ko'rsatilmagan bo'lsa fayl ishlanayotgan katalogda joylashgan bo'ladi).

Close protsedurasi.

Vazifasi: ochiq faylni yopadi.

Close (f);

Aniqlanishi:

bu yerda f - oldindan ochilgan faylga mos keluvchi faylli o'zgaruvchi.

Read protsedurasi.

Vazifasi: fayl hadini o'zgaruvchiga o'qiydi.

Read (f, v);

Aniqlanishi:

bu yerda f - faylning ixtiyoriy turiga mos faylli o'zgaruvchi (faqat matnli turli emas);

v - fayl hadining turi bilan bir xil turli o'zgaruvchi.

Reset protsedurasi.

Vazifasi: mavjud faylni ochadi.

Reset (f: file);

Aniqlanishi:

bu yerda f – faylning ixtiyoriy turiga mos faylli o'zgaruvchi va bu o'zgaruvchi fayl bilan *Assign* protsedurasi orqali bog'langan bo'lishi kerak. *Reset* protsedurasi mazkur faylni ochadi.

Rewrite protsedurasi.

Vazifasi: yangi faylni yaratadi va ochadi.

Rewrite (f: file);

Aniqlanishi:

bu yerda f – ixtiyoriy faylli turdagi faylli o'zgaruvchi. *Rewrite* protsedurasini ishlatishdan oldin f o'zgaruvchi *Assign* protsedurasi yordamida diskdagi fayl bilan bog'lanishi kerak. *Rewrite* protsedurasi yangi fayl tashkil qiladi.

Write protsedurasi.

Vazifasi: fayl hadiga o'zgaruvchini yozib qo'yadi.

Write (f, v);

Aniqlanishi:

bu yerda f – faylli o'zgaruvchi;

v - f faylining hadi bilan bir xil turli o'zgaruvchi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Oldingi tuzgan dasturmiz «d:» diskdagi tp katalogida myfile.dat faylini tashkil qildi. Endi shu fayldan qanday qilib ma'lumotlarni o'qishni ko'rib chiqaylik.

```
Var
    mydata: file of integer;
    i, j, sum: integer;
begin
    assign (mydata, 'd:\tp\myfile.dat');
    reset (mydata);                {fayl o'qish uchun ochilmoqda}
    writeln ('Salom noma'lum o'rtoq...');
    read (mydata, i);
    writeln ('myfile.dat faylidan birinchi son o'qildi');
    read (mydata, j);
    writeln ('diskdagi myfile.dat faylidan
            ikkinchi son o'qildi');
    read (mydata, sum);
    writeln ('myfile.dat faylidan uchinchi son o'qildi');
    close (mydata);                {mydata fayli yopiladi}
    writeln ('Xayr noma'lum o'rtoq...');
end.
```

Text standart faylli tur matnli fayllarni aniqlaydi. Matnli fayllar o'zaro yangi qatorga o'tish belgilari bilan ajratilgan qatorlardan tashkil topadi.

Matnli fayllar bilan ishlash uchun maxsus kiritish (**Readln**) chop etish (**Writeln**) protseduralari ko'zda tutilgan. Bu protseduralar uzunligi noma'lum katorlarni fayllardan o'qish va fayllarga yozish uchun ishlatiladi.

Endi matnli fayllar bilan ishlashga doir quyidagi dastur bilan tanishib chiqaylik:

```
var
    mytext: text;
    s: string;
begin
    assign (mytext, 'd:\tp\mytext.txt');
        {mytext faylli o'zgaruvchi orkali fayl
         ismi va yo'li aniqlanmoqda}
    rewrite (mytext);
        {fayl yozish uchun ochiq}
    writeln ('Sizning ismingiz?');
    readln (s);
    writeln ('Ismingizni diskdagi mytext.txt fayliga yozilmoqda');
    writeln (mytext, s);
        {s - qatori mytext.txt fayliga yozilmoqda}
    close ( mytext);
        {mytext fayli yopildi}
end.
```

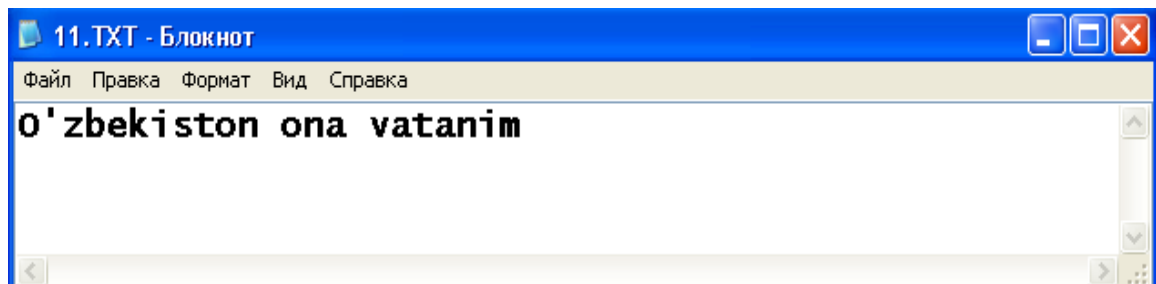
TOPSHIRIQ

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

1-misol. S-mantiqiy diskda “11.txt” faylini yaratish va unda 'O‘zbekiston ona vatanim' matnini joylashtiruvchi dastur yarating.

```
var f:text;
begin
assign ( f,'c:/11.txt');
rewrite(f);
writeln(f,'O‘zbekiston ona vatanim');
close(f);
end.
```

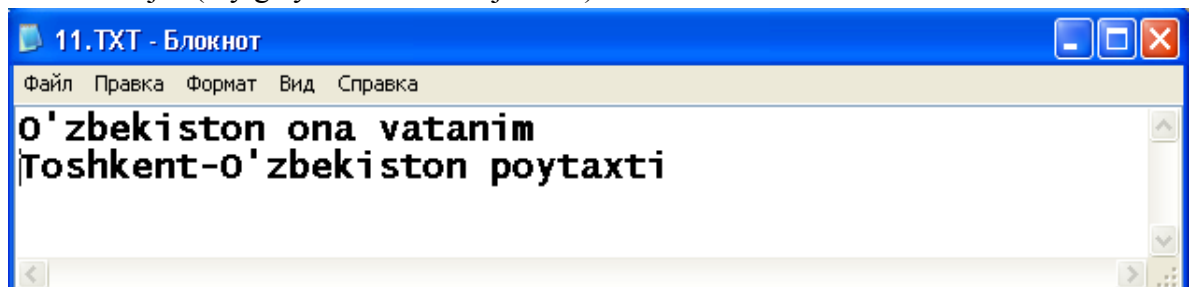
Dastur natijasi(faylga yozish amali bajariladi)::



2-misol. S-mantiqiy diskda joylashgan “11.txt” faylidagi 'O‘zbekiston ona vatanim' matnini “ Toshkent- O‘zbekiston poytaxti” matni bilan to‘ldirish.

```
var f:text;
begin
assign ( f,'c:/11.txt');
rewrite(f);
writeln(f,'O‘zbekiston ona vatanim');
close(f);
end.
```

Dastur natijasi(faylga yozish amali bajariladi):



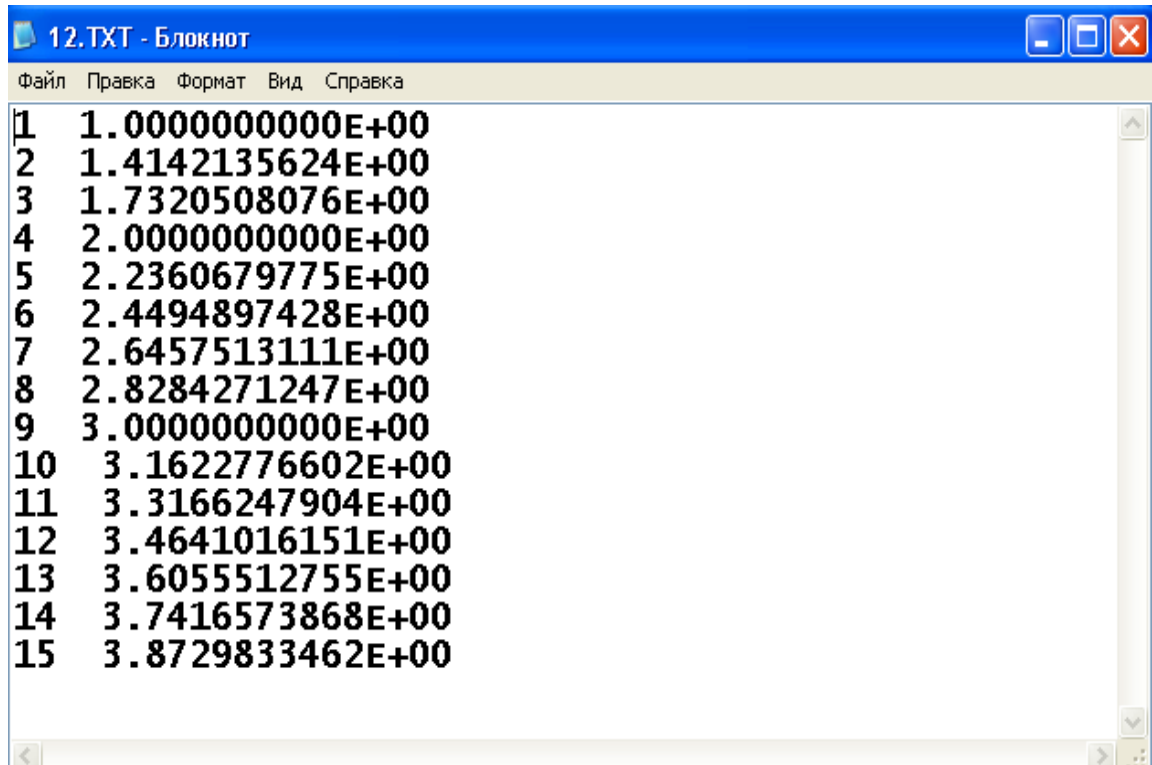
3-misol.

```
var f:text;
x,i:integer;
begin
assign ( f,'c:/12.txt');
rewrite(f);
for i:=1 to 15 do
begin
write(f,i);
write(f,' ');
write(f,sqrt(i));
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

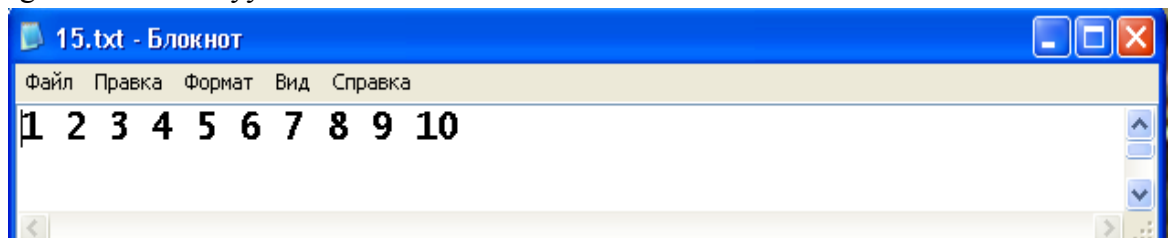
```
writeln(f);  
end;  
close(f);  
end.
```

Dastur natijasi(faylga yozish amali bajariladi):



```
1 1.0000000000E+00  
2 1.4142135624E+00  
3 1.7320508076E+00  
4 2.0000000000E+00  
5 2.2360679775E+00  
6 2.4494897428E+00  
7 2.6457513111E+00  
8 2.8284271247E+00  
9 3.0000000000E+00  
10 3.1622776602E+00  
11 3.3166247904E+00  
12 3.4641016151E+00  
13 3.6055512755E+00  
14 3.7416573868E+00  
15 3.8729833462E+00
```

Fayldan o‘qish amali bajarilishi kuzatish uchun avval S mantiqiy diskda (yoki boshqa manbada) o‘qish uchun faylni tayyorlaymiz. Buning uchun «Bloknot» dasturida «15.pas» faylini quyidagi ko‘rinishda tayyorlab olamiz:



```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

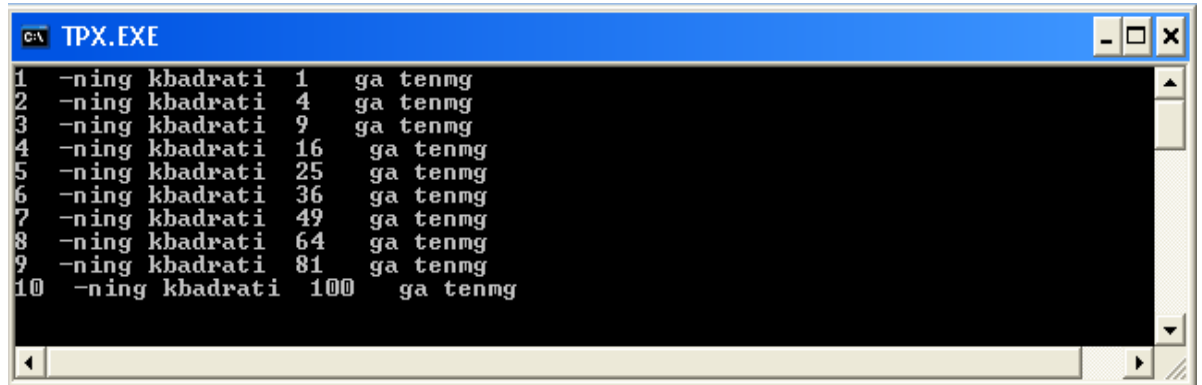
4-misol. S mantiqiy diskdagi «15.pas» fayldan 10 ta soni o‘qib ularning kvadratlarini monitorida aks ettiruvchi dastur yarating.

```
uses crt;  
var f:text;  
x,i:integer;  
begin  
assign ( f,'c:/15.txt');  
reset(f); clrscr;  
for i:=1 to 10 do  
begin  
read (f,x);  
writeln(i,'-ning kvadtati ',sqr(x),' ga teng');  
end;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

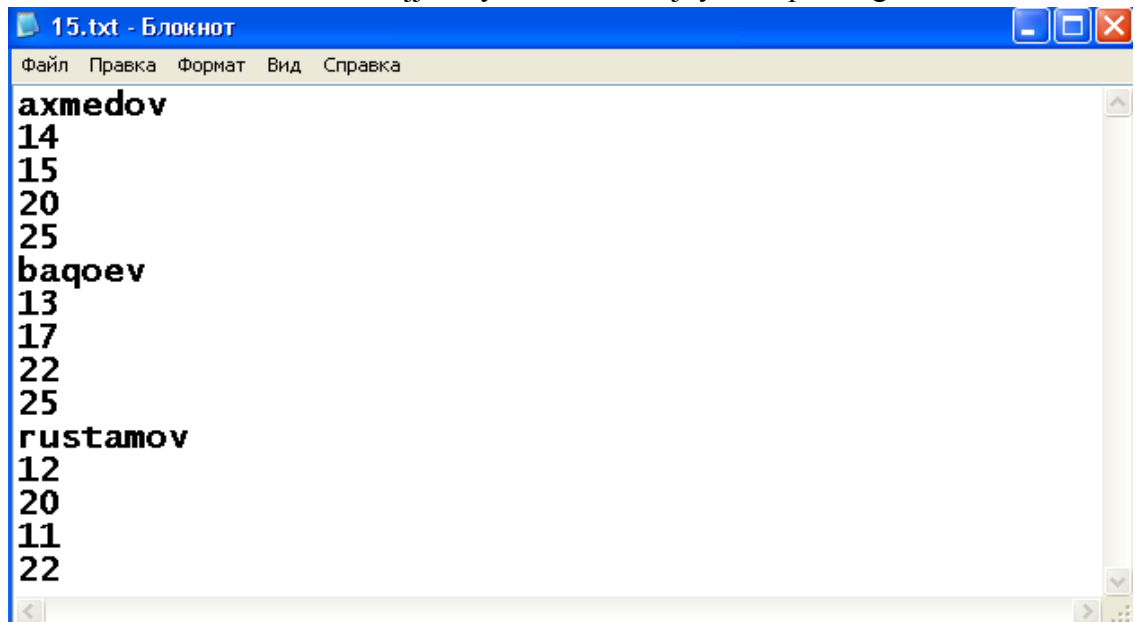
```
close(f);  
end.
```

Dastur natijasi:



```
1 -ning kbadrati 1 ga tenmg  
2 -ning kbadrati 4 ga tenmg  
3 -ning kbadrati 9 ga tenmg  
4 -ning kbadrati 16 ga tenmg  
5 -ning kbadrati 25 ga tenmg  
6 -ning kbadrati 36 ga tenmg  
7 -ning kbadrati 49 ga tenmg  
8 -ning kbadrati 64 ga tenmg  
9 -ning kbadrati 81 ga tenmg  
10 -ning kbadrati 100 ga tenmg
```

Fayldan o‘qishda faylning oxirini aniqlovchi EOF(f) mantiqiy funksiyasidan foydalanishga misol keltiramiz. Buning uchun siz quyidagi kabi sinfdoshlaringizni 4 fandan olgan test natijalarini aks ettiruvchi «15.txt» hujjatni yaratib kerakli joyda saqlashingiz zarur.



```
axmedov  
14  
15  
20  
25  
baqoev  
13  
17  
22  
25  
rustamov  
12  
20  
11  
22
```

5-misol. O‘quvchilarning 4 fandan olgan natijalarini fayldan o‘qib, har bir o‘kuvchining familiyasi va ballari yig‘indisini ekranda chop etuvchi dastur yarating.

Dastur ko‘rinishi:

```
uses crt;  
var f:text;  
fam:string;  
a,b,c,d:integer;  
begin  
assign ( f:'c:/15.txt');  
reset(f); clrscr;  
while not eof(f) do  
  
begin  
readln (f,fam, a,b,c,d);  
writeln(fam,a+b+c+d);
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
end;  
close(f);  
end.
```

(dasturni mustaqil bajarib, xulosa chiqaring)

Mustaqil bajarib ko‘ring.

```
1.  
var f:text;  
x,i:integer;  
begin  
mkdir ('c:/v1');{yangi katalog ochish uchun}  
assign ( f,'c:/v1/12.txt');  
rewrite(f);  
writeln(f,sqrt(2011));  
close(f);  
end.
```

```
2.  
var f:text;  
x,i:integer;  
begin  
assign ( f,'c:/v1/12.txt');  
rewrite(f);  
writeln(f,sqrt(2011));  
close(f);  
rename(f,'c:/v1/y1.txt');  
end.
```

Savol va topshiriqlar.

1. Fayl turdagi to‘zgaruvchi deganda nimani tushinasiz?
2. Paskalda matnli faylni ifodalovchi xizmatchi so‘zni ayting.
3. Assign operatori vazifasini aytib bering.
4. Rewrite operatori vazifasini aytib bering.
5. Rewrite operatori bilan ochilayotgan fayl tashqi xotirada avvaldan mavjud bo‘lsa qanday xodisa yuz beradi?
6. Close operator nima uchun qo‘llaniladi?
7. Append protsedurasini vazifasini aytib bering.
8. Fayldagi ma’lumotlarni ochish uchun qaysi operator yordamida ochiladi.
9. EOF funksiyasini vazifasini aytib bering.

12-AMALIY MASHG‘ULOT

Mavzu: GRAFIKA MODULI VA ULAR BILAN ISHLASH.

Mashg‘ulotning maqsadi: Talabalarga PascalABC da modular va standart modullar bilan ishlash bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

NAZARIY QISM

X va Y bu sonlarining maksimumi va minimumini aniqlovchi modulni yarataylik:

Unit Stud;

Interface {ochiq e’lonlar bo‘limi – interfeys seksiyasi}

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
function min(x,y:integer):integer;
function max(x,y:integer):integer;
Implementation {yopiq e'lonlar bo'limi}
function min(x,y:integer):integer;
Begin
    if x<=y then min:=x else min:=y;
End;
function max(x,y:integer):integer;
Begin
    if x>y then max:=x else max:=y;
End;
Begin
    {Initsializatsiya seksiyasi yo'q}
End.
```

Biz zarur modulni hosil qildik, endi uni kompilyatsiya qilishimiz lozim. Kompilyatsiya natijasida **Stud.tpu** ismli fayl hosil qilinishi kerak. Kompilyatsiya qilinmagan modulning ismi esa shunga mos holda **Stud.pas** bo'lishi kerak.

Bu moduldan foydalanish dasturi quyidagicha bo'lishi mumkin:

```
Uses Stud;
Var
    A,b,c,d:integer;
Begin
    Write('A va B larni kiriting > ');
    Readln(a,b);
    C:=max(a,b);
    Writeln('Max= ',C);
    C:=min(a,b);
    Writeln('Min= ',C);
    D:=max(a,b)+min(a,b);
    Writeln('Max+Min=',D);
End.
```

Quyida esa ekran rangini tanlash moduli misol sifatida ko'rsatilgan:

```
Unit Colors;
Interfase
Type
    Colortype =Array[0..15] of Byte;
Const
    Black:byte=0;blue:byte=1;
    Green:byte=2;cyan:byte=3;
    Red:byte=4;magenta:byte=5;
    Brown:byte=6;lightgray:byte=7;
    Darkgray:byte=8;lightblue:byte=9;
    Lightgreen:byte=10;lightcyan:byte=11;
    Lightred:byte=12;lightmagenta:byte=13;
    Yellow:byte=14;white:byte=15;
Var
    Currcolors:colortype absolute Black;
    Procedure setMonoColors;
    Procedure setColorColors;
Implementation
Const
    ColorColors:Colortype=(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
MonoColors:ColorType=(0,1,7,7,7,7,7,7,7,7,7,7,7,7,15,15);
Procedure SetMonoColors;
Begin
    CurrColors:=MonoColors;
End;
Procedure SetColorColors;
Begin
    CurrColors:=ColorColors;
End;
Var
    Ch:Char;
Begin
    Write(
        Readln(ch);
        If ch in ['M', 'm', 'M', 'm'] then SetMonoColors;
    End.
```

Graph.tpu faylidan iboratdir. Grafik tartibotida ishlash uchun bu fayl kompilyator uchun ishchi fayl bo‘lishi kerak. Dastur boshida Graph.tpu fayli joylashgan katalogga yo‘l ko‘rsatilishi kerak.

a) Grafik holatini o‘rnatish va undan chiqish tartiblari.

Komputer ekranining oddiy ish tartiboti matnli hisoblanadi. Matn tartibotidan grafik tartibotiga o‘tish uchun Graph modulining InitGraph proseduralari ishlatiladi. Uning umumiy ko‘rinishi quyidagicha:

InitGraph (D, M, F); - ekranni grafik holatga o‘tkazish. Bu yerda D- drayver nomeri, M- tartibot nomeri, F- esa kerakli drayver mavjud bo‘lgan faylga yo‘l. Agar F o‘zgaruvchi bo‘sh (F='') satrdan iborat bo‘lsa, drayver joriy katalogdan izlanadi. D va M lar o‘zgaruvchi parametrlardir. InitGraph proseduralari ishga tushirayotgan paytda D ning qiymati 0 ga teng bo‘lsa kerakli drayver va shu drayver uchun optimal grafik tartibot avtomatik tarzda aniqlanadi.

Graph modulida 0 ga teng bo‘lgan Detect o‘zgarmasi ishlatiladi.

Grafik holatidan chiqish yoki monitorni boshlang‘ich ish holatiga o‘tkazish uchun CloseGraph proseduralari ishlatiladi.

CloseGraph- grafik holatidan chiqish.

b) Graph modulining funktsiya va proseduralari

Quyida Graph moduliga tegishli prosekura va funktsiyalarni qarab chiqamiz.

Koordinatalarni o‘rnatish prosedura va funktsiyalari

Ko‘plab grafik prosedura va funktsiyalarda ekranda joriy o‘rinni ko‘rsatuvchi ko‘rsatkichdan foydalaniladi. Ko‘rsatkichning matn kursoridan farqi shundaki, u ekranda ko‘rinmaydi. Ana shu ko‘rsatkichning holati, shu bilan birga umuman grafik holatda har qanday koordinata ekranning yuqori chap burchagi (0,0) koordinataga nisbatan beriladi. Shunday qilib, ekranning gorizontal koordinatasi chapdan o‘ngga qarab, vertikal koordinatasi esa yuqoridan pastga qarab o‘sib boradi.

GetMaxX va GetMaxY funktsiyalari. Joriy ish tartibotida mos ravishda maksimal gorizontal va vertikal koordinatalarini aniqlaydi.

GetX va GetY funktsiyalari. Integer turidagi qiymatlar bo‘yicha ekranning gorizontal va vertikal koordinatalarini o‘rnatadi. Agar oyna o‘rnatilmagan bo‘lsa, koordinatalar ekranning chap yuqori burchagiga nisbatan olinadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

SetViewPort prosedurasi - grafik ekranda to‘g‘ri burchakli oyna hosil qiladi. Uning umumiy ko‘rinishi:

Procedure SetViewPort (X1, Y1, X2, Y2: integer; ClipOn: boolean);

Bu yerda X1...Y2 lar oyna burchagining koordinatalari. (X1, Y1)- yuqori chap va (X2, Y2) o‘ng quyi burchak koordinatalari, ClipOn – Boolean tipidagi ifoda.

MoveTo prosedurasi - ko‘rsatkichni yangi joriy holatga o‘rnatadi. Uning umumiy ko‘rinishi:

Procedure MoveTo(X,Y:integer); Bu yerda X,Y ko‘rsatkich-ning gorizontal va vertikal yo‘nalishlar bo‘yicha yangi koordinatalari. Koordinata ekranning yuqori chap burchagiga asosan olinadi.

ClearDevice prosedurasi – grafik oynani tozalaydi.

Uning umumiy ko‘rinishi:

Procedure ClearDevice; Bu prosedura bajarilganda ekran tozalanadi, ko‘rsatkich ekranning yuqori chap burchagiga joylashadi va ekran fon rangi bilan bo‘yaladi.

Ko‘pburchaklar hosil qilish funksiya va proseduralar Rectangle prosedurasi – ko‘rsatilgan burchak koordinatalari bo‘yicha to‘g‘ri to‘rtburchak hosil qilish. Uning umumiy ko‘rinishi:

Procedure Rectangle (X1, Y1, X2, Y2: integer); bu yerda X2 – X1... to‘g‘ri to‘rtburchak burchaklarining koordinatalari. (X1, Y1) – yuqori chap to‘rtburchak va (X2, Y1) quyi o‘ng burchak koordinatalari. To‘g‘ri burchak joriy rang va joriy chiziq qalinligi bo‘yicha hosil qilinadi.

DrawPoly prosedurasi – egilish nuqtalarining koordinatalari bo‘yicha ixtiyoriy siniq chiziqlarni chizadi. Uning umumiy ko‘rinishi:

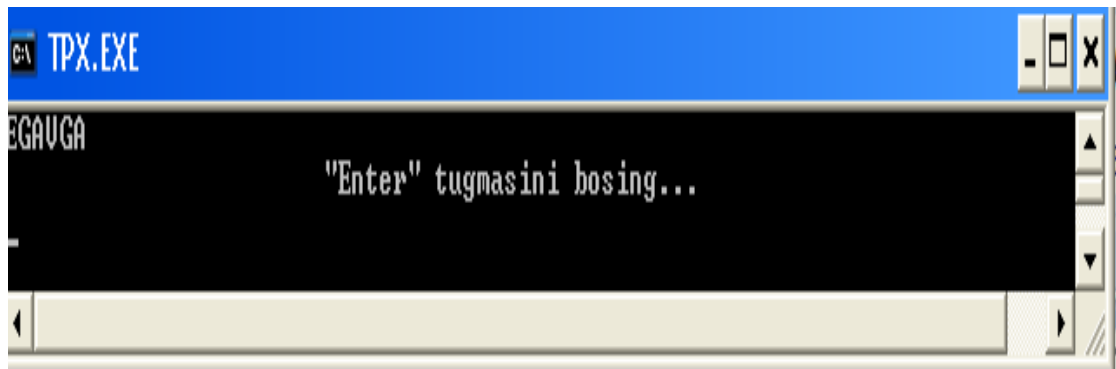
Procedure DrawPoly(N:Word;Var Points); Bu yerda N – egilish nuqtala-rining soni; Points – nuqta koordinatalarini tashkil etuvchilar. Chizish jarayonida joriy rang va joriy chiziq qalinligi tanlanadi.

1-masala. Dastur tarkibida GetDriverName funksiyasidan foydalanib, yuklangan grafikali drayver nomini aniqlang.

```
uses Graph;
var
Driver, Mode, Error: integer;  dr: string;
begin
Driver:= Detect;
InitGraph(Driver, Mode, 'c:\tp\bgi');
dr:=GetdriverName;
Writeln(dr, ' ');
Writeln("'Enter" tugmasini bosing...':50);
Readln;
CloseGraph ;
end.
```

Dastur natijasi yuklangan grafikali drayverga bog‘liq va u quyidagicha bo‘lishi mumkin:

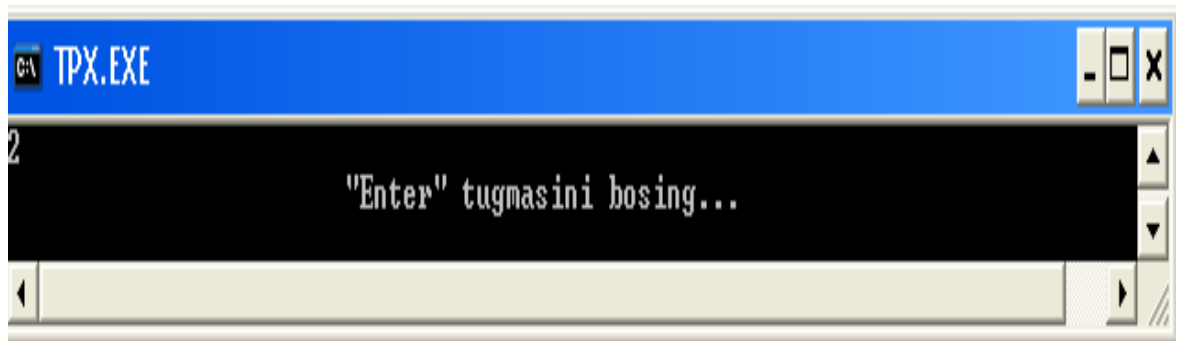
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



2-masala. GetMaxMode funksiyasidan foydalanib, adapterning grafikali holatlarining maksimal miqdorini aniqlovchi dastur yarating.

```
uses Graph;
var
  Driver, Mode, Error: integer;  d: integer;
begin
  Driver:= Detect;
  InitGraph(Driver, Mode, 'c:\tp\bgi');
  d:=GetMaxMode;
  restorecrtMode;
  Writeln(d);
  Writeln(""Enter" tugmasini bosing...":50);
  Readln;
  CloseGraph ;
end.
```

Dastur natijasi:



3-masala. GetModName funksiyasi yordamida ekranning yechim holatlarini va adapterning ish holati nomini parametr bo‘yicha aniqlang.

Dastur ko‘rinishi:

```
uses Graph;
var
  Driver, Mode, Error, i: integer;  ds: string;
begin
  Driver:= Detect;
  InitGraph(Driver, Mode, 'c:\tp\bgi');
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
writeln (getdrivename);
for i:=0 to 3 do
begin
ds:=GetModeName(i);
Writeln(ds);
end;
Writeln("Enter" tugmasini bosing...':50);
Readln;
CloseGraph;
end.
```

4-masala.

$y = 2 \sin(x) + 3 \cos(x)$ -funksiyaning grafigini chizish dasturini tuzing.

Program n6;

```
Uses Graph;
Var Gd, Gm : integer;
    dx, x1, x2, x, y: real;
Begin
x1 := - 3 * pi;    x2 := 4 * pi;    Gd := Detect;
InitGraph(Gd, Gm, "");
ErrorCode := GraphResult;
if ErrorCode <> grOk
then
begin
WriteLn (Графическая ошибка: ', GraphErrorMsg(ErrorCode));
Halt;
end;
SetColor (1);
SetViewPort (100, 50, 470, 400, True);
Rectangle (0, 80, 275, 270);
SetColor (10);
x := x1;
dx := (x2 - x1) / 300;
y := 2 * Sin (x) + 3 * Cos (x);
MoveTo (Round ((x - x1) / (x2 - x1)), Round (175 - 20 * y));
x := x + dx;
SetColor (11);
while x <= x2 do
begin
y := 2 * Sin (x) + 3 * Cos (x);
LineTo (Round ((x - x1) * 275 / (x2 - x1)), Round (175 - 20 * y));
x := x + dx;
end;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
SetColor (6);
Line (Round (0), Round (350 / 2), Round (273), Round (350 / 2));
Line (Round (273), Round (350 / 2), Round (268), Round (170));
Line (Round (273), Round (350 / 2), Round (268), Round (180));
Line (Round (125), Round (81), Round (125), Round (270));
Line (Round (125), Round (81), Round (120), Round (86));
Line (Round (125), Round (81), Round (130), Round (86));
SetColor (4);
SetViewPort (0, 0, GetMaxX, GetMaxY, True);
SetTextStyle (TriplexFont, HorizDir, 3);
SetColor (15);
OutTextXY (222, 100, 'y');
OutTextXY (229, 220, '0');
OutTextXY (377, 207, 'x');
SetColor (4);
OutTextXY (80, 50, 'y = 2 sin (x) + 3 cos (x);');
Readln;
CloseGraph;
End.
```

5-masala. Keltirilgan dastur tahlilini keltirning.

Program Ellipses;

```
Uses CRT, Graph;
Var d,m,x,y:integer;
xx,yy,t,a:real;
Const xo=320;yo=240;
Begin d:=detect; m:=detect;
InitGraph(d,m, ""); a:=0;
while a<=2*pi do begin t:=0;
while t<=2*pi do begin
xx:=200*cos(t); yy:=80*sin(t);
x:=round(xo+(xx*cos(a)-yy*sin(a)));
y:=round(yo-(xx*sin(a)+yy*cos(a)));
PutPixel(x,y,white);
t:=t+0.001; end;
a:=a+pi/20; end;
repeat
until KeyPressed;
CloseGraph
End.
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

TOPSHIRIQ

1. Asosi besh burchak bo‘lgan og‘ma prizma va uning kesimlarini chizish dasturini tuzing.
2. Ekranda televizor tasvirini hosil qilish dasturini tuzing.
3. Ekranda to‘rtburchak va uning ichida joylashgan ochiq konvert tasvirini hosil qilish dasturini tuzing.
4. Ichmaich joylashagan ellips, aylana, kvadrat va uchburchakni hosil qilish dasturini tuzing.
5. Ekranda stol tasvirini hosil qilish dasturini tuzing.
6. Shar va uning kesimlarini chizish dasturini tuzing.
7. Ekranda konus va silindr tasvirini hosil qilish dasturini tuzing.
8. Stol va uning ustida gul vazasini chizish dasturini tuzing.
9. Ikkita vagondan iborat poyezd tasvirini hosil qilish dasturini tuzing.
10. Yengil avtomabil tasvirini hosil qilish dasturini tuzing.
11. Monitor tasvirini hosil qilish dasturini tuzing.
12. Uch o‘lchovli koordinatalar sistemasini chizish, yo‘nalishlarini ko‘rsatish va o‘qlarini x, y, z bilan belgilash dasturini tuzing.
13. $y=\cos x$ funksiyani grafigini chizish dasturini tuzing.
14. To‘g‘ri chiziq bo‘yicha harakatlanuvchi shar tasvirini chizing.

13-AMALIY MASHG‘ULOT

MAVZU: PASKAL TILIDA SODDA SHAKLLAR CHIZISH.

Mashg‘ulotning maqsadi: Talabalarda Paskal dasturlash tilining grafik imkoniyatlaridan foydalanib dastur tuzish bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

Qisqacha nazariy ma‘lumot

Graph modulining protsedura va funksiyalari. YUqoridagi mavzuda ko‘rib chiqilgan funksiya va protseduralar yordamida faqat chiziqlar chizish mumkin. Endi boshqa turli xil ranglar bilan to‘ldirilgan figuralar chizishni tashkil etishga yordam beruvchi yana bir nechta protsedura va funksiyalar bilan tanishib chiqamiz.

1. **SetFillStyle(Style,Color)** protsedurasi - Color rangi bilan sohalarni to‘ldirish va ularni ko‘rsatilgan uslubda to‘ldirish (shtrixovka qilish) uchun ishlatiladi;

Sohani turli ranglar bilan to‘ldirish o‘zgarmlari:

const

EmptyFill=0; {sohani ekran fonining rangiga bo‘yaydi}

SolidFill=1; {sohani belgilangan rangda uzluksiz to‘ldirish}

LineFill=2; {sohani qalin gorizonttal (-----) chiziqlar bilan to‘ldiradi}

LtSlashFill=3; {sohani ingichka"/// " belgilari bilan to‘ldiradi}

SlashFill=4; {sohani qalin "/// " belgilari bilan to‘ldiradi}

BkSlashFill=5; {sohani qalin "\\\\" belgilari bilan to‘ldiradi}

LtBkSlashFill=6; {sohani "\\\\" qatlami bilan to‘ldiradi}

HatChFill=7; {sohani to‘r bilan to‘ldiradi}

XHatChFill=8; {sohani egri to‘r bilan to‘ldiradi}

InterLeaveFill=9; {sohani zich egri shtrixovka bilan to‘ldiradi}

WideDotFill=10; {sohani kam uchrovchi nuqtalar bilan to‘ldiradi}

CloseDotFill=11; {sohani zich nuqtalar bilan to‘ldiradi}

UserFill=12; {sohani dasturchi aniqlagan shtrixovka bilan to‘ldiradi}

2. **Bar(x1,y1,x2,y2)** protsedurasi ekrandagi rang va shtrixovka ustiga to‘g‘ri to‘rt burchak quradi;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3. **Bar3D**($x1,y1,x2,y2,Depth,Top$) protsedurasi ham shunday rang va shtrixovka bilan to‘ldirilgan parallelepiped chizadi. **Depth** o‘zgaruvchisi parallelepiped balandligini anglatadi. **Top** mantiqiy o‘zgaruvchisi **true** qiymatli bo‘lsa parallelepipedning yuqori asosi chiziladi aks xolda, chizilmay ochiq qoladi;

4. **FillEllipse**($x,y,XRadius,YRadius$) protsedurasi oldin o‘rnatilgan rangga to‘ldirilgan ellips chizadi. Ellips o‘qlari koordinata o‘qlariga paralel deb olinadi. **XRadius** - ellips eni, **YRadius** - ellips balandligi.

5. Grafik rejimdagi matnlar

Grafik rejimda matnlarni yozish uchun ikki xil turdagi shriftdan foydalanish mumkin: nuqtalar matritsasi va simbolni tashkil etuvchi vektorlar qatori orqali.

SHrifltlar fayllari .CHR kengaytmasiga ega bo‘ladi va shriftni ishga sozlaganda kerakli fayllar ishchi katalogda yoki .BGI grafik drayveri joylashgan katalogda bo‘lishi kerak.

SHriftni tanlash va masshtab o‘rnatish **SetTextStyle** protsedurasi yordamida amalga oshiriladi:

SetTextStyle(*Font, Direction, Size*) - kerakli shriftni o‘rnatadi, matnni chiqarish yo‘nalishini aniqlaydi va belgilar o‘lchovini belgilab beradi. **Font** - shriftni aniqlovchi o‘zgaruvchi, **Direction**-matnni chop etish yo‘nalishini ko‘rsatuvchi o‘zgaruvchi (chapdan o‘ngga yoki pastdan yukoriga), **Size**-shrift o‘lchovini aniqlovchi o‘zgaruvchi. Matritsali shriftda o‘lchov **Sizeq1**, vektor shriftida esa **Sizeq4** qiymatlarida erishiladi.

Turli xil shriftlarni ko‘rsatish va matnlarni chop etish yo‘nalishlarini tanlash uchun quyidagi o‘zgarmaslar aniqlangan:

Const {shrifltlar}

DefaultFont=0; {8x8 nuqtali standart matritsali shrift}

TriplexFont=1; {vektorli shrift}

SmallFont=2; {vektorli shrift}

SansSerifFont=3; {vektorli shrift}

GothicFont=4; {vektorli shrift}

{matn yo‘nalishi}

HorizDir=0; {chapdan o‘ngga}

VertDir=1; {pastdan yuqoriga}

OutTextXY($x,y,TextString$) protsedurasi - oldindan aniqlangan shriftda, yo‘nalishda va belgi o‘lchovida **TextString** qatorini (x,u) nuqtadan boshlab chop etadi.

SetTextJustify(*Horiz,Vert*) protsedurasi **OutTextXY** protsedurasi chop etadigan matnni avtomatik tarzda tekislab beradi. **Horiz** - gorizontol, **Vert** - vertikal tekislashlar.

Matnlarni tekislash uchun quyidagi o‘zgarmaslar aniqlangan:

const

{gorizontol tekislash uchun}

LeftText=0; {chap tomonga nisbatan tekislash}

CenterText=1; {markazga nisbatan tekislash}

RightText=2; {o‘ng tomonga nisbatan tekislash}

{vertikal tekislash uchun}

BottomText=0; {pastgi tomonga nisbatan tekislash}

CenterText=1; {markazga nisbatan tekislash}

TopText=2; {yuqori tomonga nisbatan tekislash}

6. Ekran sohalari

GetImage, PutImage protseduralari va **ImageSize** funksiyasi yordamida tasvirlarning to‘g‘ri to‘rtburchakli sohasini hotirada eslab qolish va ularni ekranga chiqarishimiz mumkin.

1.**ImageSize**($x1,y1,x2,y2$) funksiyasi – ekranning to‘g‘ri to‘rtburchakli sohasini saqlash uchun zarur bo‘lgan xotira o‘lchovini (baytlarda) beradi.($x1,y1$) to‘g‘ri to‘rtburchakli ko‘rinishning chapdan yuqoridagi, ($x2,y2$) - esa pastdan o‘ngdagi burchak nuqtalari uchun koordinatalar.

2.**GetImage**($x1,y1,x2,y2,Area$) protsedurasi xotiraning **Area** sohasida to‘g‘ri to‘rtburchakli ekran tasvirini saqlaydi. ($x1,y1$) va ($x2,y2$) lar yuqoridagi ma’noda qayta ishlatilmoqda.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3. **PutImage**($x,y,Area,Mode$) protsedurasi ekranning ko‘rsatilgan joyiga tasvir ko‘rinishini chop etadi. (x,y) – xotiraning **Area** sohasidagi tasvir ko‘rinishi nusxasini chop etiladigan, ekranning chapdan yuqoridagi nuqtasining koordinatasi. **Mode**-tasvirni ekranga chiqarish rejimi. Tasvirlarni ekranga chiqarish rejimini aniqlash uchun foydalani-ladigan o‘zgarmlar:

```
const
{PutImage protsedurasi uchun o‘zgarmlar}
NormalPut=0; { mavjud tasvirni almashtirish}
XorPut=1; {XOR mantiqiy amali}
OrPut=2; {OR mantiqiy amali}
AndPut=3; {AND mantiqiy ko‘paytirish amali}
NotPut=4; {NOT mantiqiy rad etmoq amali}
```

7. Xatolar tahlili

Grafik rejimni o‘rnatish mavzusida yo‘l qo‘yilgan xatolar diagnostikasi uchun **GraphResult** funksiyasidan foydalangan edik. Hozir shu funksiya beradigan xatolar kodi bilan to‘liqroq tanishib chiqaylik.

Quyida **GraphResult** funksiyasi beradigan kodlarga mos o‘zgarmlar ro‘yhati keltirilgan:

```
const
grOk=0; {xatolar yo‘q}
grNoInitGraph=-1; {Grafik rejim o‘rnatilmagan(InitGraph
protsedura-sini ishga tushiring )}
grNotDetected=-2; {Grafik plata o‘rnatilmagan}
```

- sohalarni ko‘chirishda xotira chegarasidan chiqish;
- zarur sohani bo‘yash paytida xotira chegarasidan chiqish;
- shrift fayli topilmagan;
- shrift faylini ishga tushirish uchun xotira
- yetishmayapti;
- tanlangan drayver uchun noto‘g‘ri grafik rejim.

8. Graph modulining protsedura va funksiyalari

1. **Arc** protsedurasi - aylana yoyini chizadi.

Aniqlanishi : Arc(x,y : integer; StAng, EndAng, Radius: Word);
 x,y - aylana markazining koordinatasi;

StAng, EndAng - mos ravishda yoyning boshlang‘ich va oxirgi burchaklari;
Radius-aylana radiusi.

2. **Bar** protsedurasi - rangga bo‘yalgan to‘g‘ri to‘rtburchak chizadi.

Aniqlanishi: Bar($x1,y1,x2,y2$:integer);

($x1,y1$) va ($x2,y2$) mos ravishda to‘g‘ri to‘rtburchakning chetki nuqtalari koordinatalari.

3. **Bar3D** protsedurasi rangga bo‘yalgan parallelepiped chizadi.

Aniqlanishi :Bar3D($x1,y1,x2,y2$:integer;Depth:word;Top:boolean);

($x1,y1$) va ($x2,y2$) asosni tashkil etuvchi to‘g‘ri to‘rtburchak uchlarining koordinatalari;
Depth -parallelepiped chuqurligi;

Top- mantiqiy o‘zgaruvchi.

4. **Circle** protsedurasi - aylana chizadi;

Aniqlanishi: Circle(x,y :integer;Radius:word);

(x,y) aylana markazining koordinatasi;

Radius-aylana radiusi.

5. **CloseGraph** protsedurasi grafik rejimini uzadi.

Aniqlanishi :Closegraph;(parametrsiz protsedura)

6. **DrawPoly** protsedurasi - ko‘p burchak chizadi.

Aniqlanishi :DrawPoly(NumPoints:word; var PolyPoints);

NumPoints - ko‘pburchak tomonlari soni;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

PolyPoints - ko‘pburchak uchlarning koordinatalaridan tuzilgan massiv.

7. **Ellipse** protsedurasi - ellips yoyini chizadi.

Aniqlanishi: Ellipse(x,y:integer;StAng,EndAng:word;XRadius,YRadius:word);

(x,y) – ellips markazning koordinatasi;

StAng va EndAng - yoyning boshlang‘ich va oxirgi burchaklari;

Xradius va Yradius mos ravishda ellips balandligi va eni.

8. **FillPoly** protsedurasi - rangli ko‘pburchak chizadi.

Aniqlanishi: FillPoly(NumPoints:word; var PolyPoints);

NumPoints - ko‘pburchakning uchlari soni;

PolyPoints - ko‘pburchak uchlari koordinatalaridan tuzilgan massiv.

9. **GetArcCoords** protsedurasi - oxirgi marta ishlatilgan **Arc** protsedurasining koordinatalarini aniqlaydi.

Aniqlanishi: GetArcCoords(var ArcCoords:ArcCoords Type);

10. **GetColor** funksiyasi - ekran rangini aniqlaydi.

Aniqlanishi: GetColor:word;

11. **GetGraphMode** funksiyasi - grafik ekranni qaytaradi.

Aniqlanishi: GetGraphMode:integer;

12. **GetImage** protsedurasi - ekranning berilgan sohasini Area da saqlaydi.

Aniqlanishi: GetImage(x1,y1,x2,y2:integer;var Area);

13. **GetMaxColor** funksiyasi - rangning eng katta qiymatini hisoblaydi.

Aniqlanishi: GetMaxColor:word;

14. **GetPixel** funksiyasi - berilgan nuqta rangini aniqlaydi.

Aniqlanishi: GetPixel(x,y:integer):word;

15. **GraphErrorMsg** funksiyasi - berilgan kod bo‘yicha xato haqida satr ma’lumot beradi.

Aniqlanishi: GraphErrorMsg(Code:integer):string;

16. **LineTo** protsedurasi - oldingi aniqlangan nuqtadan berilgan nuqttagacha kesma chizadi.

Aniqlanishi: LineTo(x,y:integer);

17. **PieSlice** protsedurasi sektor chizadi.

Aniqlanishi: PieSlice(x,y:integer;StAng,EndAng,Radius:word);

1- **MISOL.** $y = x^2$ funksiyani grafigini quring.

```
program grafik_funk;
  Uses Graph;
  var
    a,b,c,d,h,x,y:real;
    de,re,n,k,u,v:integer;
    const um=639;vm=469;
  function f(x:real):real;
  begin
    f:=SQR(x);
  end;
  procedure kiritish(var a,b,c,d:real;
                    var n:integer);
  begin
    writeln('Grafik parametrlari');
    write('Chapdan: a='); readln(a);
    write('O‘ngdan: b='); readln(b);
    write('Pastdan: c='); readln(c);
    write('Yuqoridan: d='); readln(d);
    write('n='); readln(n); {grafikdagi nuqtalar soni}
  end;
  begin
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
kiritish(a,b,c,d,n);
de:=detect;Initgraph(de,re,' ');
h:=(b-a)/n; {qadam}
for k:=0 to n do
begin
x:=a+h*k;
y:=f(x);
u:=round((x-a)*um/(b-a)); {nuqtalarning koordinatalarini chiqarish uchun}
v:=round((d-y)*vm/(d-c)); {"integer" tipiga o'tkazish}
putpixel(u,v,7);
end;
readln {tugmacha bosilgunga qadar grafik ekranda saqlanadi}
end.
Arc –protsedurasi.
```

Markazi x,y bo‘lgan yoy chizishda foydalaniladi.

_StAngle – yoyning boshini aniqlovchi gradus o‘lchovli burchak.(soat strelkasi yo‘nalishiga qarshi-chapdan o‘ngga);

EndAngle - yoyning oxirini aniqlovchi gradus o‘lchovli burchak.(soat strelkasi yo‘nalishiga qarshi-chapdan o‘ngga);;

Protseduradan foydalanishga misol:

Uses Graph;

Var Gd, Gm: Integer;

Radius: Integer;

begin

Gd := Detect; InitGraph(Gd, Gm, ’’);

for Radius := 1 to 5 do

Arc(100, 100, 0, 90, Radius* 10);

ReadLn; CloseGraph; end.



Izoh:Radius – yoy radiusi.

Bar-protsedurasi.

sintaksisi: **Bar(X1, Y1, X2, Y2: Integer); SetFillStyle** или **SetFillPattern** protseduralari orqali aniqlangan rang yoki shablona ko‘ra ichi bo‘yalgan to‘g‘ri to‘rtburchak chizadi.

X1,Y1,X2,Y2 – to‘rtburchakning diagonalida joylashgan huqtalar koordinatalari.

Protseduradan foydalanishga misol:

uses Graph;

var Gd, Gm, I, Width: Integer;

begin

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
Gd := Detect; InitGraph(Gd, Gm, ”);  
Width := 10;  
for I := 1 to 15 do  
Bar(I*Width, I* 10, Succ(I)*Width, 200);  
ReadLn; CloseGraph;end.
```



Bar3D protsedurasi berilgan parameterlarga ko‘ra parallelepiped chizadi.

sintaksisi: Bar3D(x1, y1, x2, y2: Integer; Depth: Word; Top: Boolean);

top kattalik qiymatiga ko‘ra yuqori qirralar aks ettiriladi yoki aks ettirilmaydi)

Protseduradan foydalanishga misol:

```
uses Graph; var  
Gd, Gm: Integer;  
y0, y1, y2, x1, x2: Integer;  
begin  
Gd := Detect; InitGraph(Gd, Gm, ”);  
y0 := 10; y1 := 60; y2 := 110;  
x1 := 10; x2 := 50;  
Bar3D(x1, y0, x2, y1, 10, TopOn);  
Bar3D(x1, y1, x2, y2, 10, TopOff);  
ReadLn; CloseGraph; end.
```

Circle-protsedurasi

Sintaksisi: Circle(X,Y: Integer; Radius: Word);

X,Y – aylana markazi koordinatalari;

```
Uses Graph;  
Var Gd, Gm: Integer;  
Radius: Integer;  
begin  
Gd := Detect; InitGraph(Gd, Gm, ”);  
for Radius := 1 to 5 do Circle(100, 100, Radius* 10);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

ReadLn; CloseGraph; end.



FloodFill-protsedurasi.

Sintaksisi: FloodFill(X, Y: Integer; Border: Word);

SetFillStyle или **SetFillPattern** protseduralari orqali aniqlangan rang yoki shablona ko‘ra uzluksiz kontur bilan aniqlangan sohani ichini to‘ldiradi. x,y- sohaga tegishli nuqta koordinatalari. **Border** -kontur chiziqning rangi.

Protseduradan foydalanishga misol:

```
uses Graph; var Gd, Gm : Integer;
```

```
begin
```

```
Gd := Detect; InitGraph(Gd, Gm, ’’);
```

```
SetColor(GetMaxColor);Circle(50, 50, 20);
```

```
FloodFill(50,50,GetMaxColor);ReadLn; CloseGraph; end.
```



Topshiriq

1. Uchlarining koordinatalari (10,10) va (630, 470) bo‘lgan ko‘k rangli kesma chizing. Fon sariq rangda bo‘lsin.
2. Chap yuqori va o‘ng pastki burchaklari mos ravishda (10,10) va (200,100) koordinatali nuqtalarda bo‘lgan ko‘k rangli to‘g‘ri to‘rtburchak chizing.
3. Uchlarining koordinatalari berilgan qizil rangli oltiburchak chizing.
4. Ekranning to‘rtta burchagida eni 60 va bo‘yi 40 ga teng qizil rangli to‘rtburchaklar chizing.
5. Svetofor rasmini chizuvchi dastur tuzing.
6. Oy va yulduzlar tasvirlangan kechki osmon manzarasini chizing.
7. Qizil rangli muntazam beshburchak chizing.
8. Velosiped rasmini chizing.
9. Ekranni teng to‘rt bo‘lakka bo‘lib, ularni mos ravishda qizil, sariq, yashil va ko‘k ranglarga bo‘yang.
10. Mos ravishda qizil, sariq va yashil chiroqlari yonib turgan uchta svetofor rasmini chizing.
11. Dengiz uchra nur taratayotgan Quyosh rasmini chizng. Dengizni chizishda yoy chizish operatoridan foydalaning.
12. O‘n ikki xil bo‘yash usulini namoyish etuvchi 40x40 o‘lchamli 12 ta kvadrat chizing.
13. Ichi siyrak qizil nuqtalar bilan to‘ldirilgan muntazam oltiburchak chizing.

Mustaqil ishlash uchun vazifalar.

1. «Broun harakati»ni tasvirlovchi dastur yarating.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

2. Turli ranglarda bo‘lgan 20-ta konsentrik aylalar chizuvchi dastur yarating.
3. $A[1..10]$ massiv elementlari qiymatlariga mos diagramma tuzuvchi dastur yarating.
4. Ixtiyoriy shakl (primitiv)ni ekranda «harakatlanuvchi» holatda tasvirlang.
5. Asosiy menyuda sodda geometrik figuralar nomlari bilan menyular yarating va ularni tanlashda mos figurani chizib beruvchi hamda uning nomini aks ettiruvchi dastur tuzing.
6. Forma uchun tarkibida sodda geometrik figuralar nomlari bilan kontekstli menyu yarating va ulardan birini tanlashda mazkur figura elementlarini va ular orasidagi bog‘liqlikni matn shaklida ko‘rsatuvchi dastur tuzing.
7. Protsedura va funksiyadan foydalanib maydonning turli nitalarida o‘zingizni ism va familiyangizni turli ranglarda chop etuvchi dastur tuzing.
8. Ekranda kitob do‘konida mavjud kitoblar, jurnallar, kanselyariya va boshia mahsulotlarni soniga nisbatan doiraviy diagramma ko‘rinishini beruvchi dastur tuzing.
9. Ekranda olimpiada bayrog‘ini tasvirlovchi dastur tuzing.
10. Ekranda xarakatlanuvchi nitalarni tasvirlovchi dastur tuzing.
11. R-radiusli aylana ichida turli ranglardagi nitalarni aks ettiruvchi dastur tuzing.
12. Ekranning turli nuqtalarida 6-burchakli shakllar yaratuvchi dastur tuzing.
13. $y=\sin x + \cos x$ funksiya grafigini tasvirlovchi dastur yarating
14. Jadvalda ko‘rsatilgan iiymatlar bo‘yicha gistogramma yaratuvchi dastur tuzing.

Oylar	1	2	3	4	5	6	7	8	9	10	11	12
T(o‘rtacha gradusda)	-10	-8	-2	5	12	15	22	28	30	27	16	-5

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

FOYDALANILGAN ADABIYOTLAR

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Сер: Классические учебники: COMPUTER SCIENCE. М.: МЦНМО, –960с., 2004.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2010 y.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
4. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyos”, 2011y.
5. T. X. Holmatov, N. I. Toyloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta‘minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
6. М. Ашуров, М. Мирмахмудов, Ш. Сапаев. Замонавий дастурлаш тиллари фанидан лаборатория ишлари. Т. : ТДПУ, 2008 й.
7. M.J.Yunusova, A.V.Rahimov. Dasturlash bo‘yicha praktikum.T.: “Ilm ziyos”, 2006 y.
8. А.В.Файсман. Профессиональное программирование на Турбо – Паскале. Информ Экс - Корпорейшн, 1992 г.
9. B.Boltayev, M.Mahkamov, A.Azamatov. Paskal dasturlash tili. T.; 2007 y.
10. А.Н.Марченко. Программирование в среде *Turbo Pascal 7.0*. К.,Век+, М., «ДЕСС», 1999 y.
11. Karimov R. va boshqalar. Dasturlash.T. “O‘zbekiston”. 2003 y. 206 b.
12. Матрос Д.Ш., Поднебесова Г.Б. Теория алгоритм. Учебник для педагогического образования. М.: Бином. Лаборатория знаний, - 2008. -202с.
13. Макконелл Дж. Основы современных алгоритмов. 2-доп.изд., М.: ТЕХНОСФЕРА, 366с., 2004.
14. Вирт Н. Алгоритмы и структуры данных. С примерами на Паскале. Санкт-Петербург, 352с., 2005.
15. Рейест Р. и др. Алгоритмы: построение и анализ. М., Мир, 1994.
16. Малышко В.В. Алгоритмы и алгоритмические языки. Конспект лекций для студентов Ташкентского филиала МГУ, 68с., 2006.
17. Пильщиков В.Н., Абрамов В.Г., Вылиток А.А., Горячая И.В. Машина Тьюринга и алгоритмы Маркова. Решение задач. (Учебно-методическое пособие) Московский государственный университет им. М.В.Ломоносова, Факультет вычислительной математики и кибернетики, Москва, 47с., 2006.
18. Yunusov A.S. Matematik mantiq va algoritmlar nazariyasi elementlari. Samarqand davlat universiteti nashriyoti, 182 b, 2012y.
19. www.de.uspu.ru/Informatics/metodes/DPP/F/08/1/Index.htm.
20. www.ziyonet.uz.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI
NAVOIY DAVLAT PEDAGOGIKA INSTITUTI**

«INFORMATIKA O‘QITISH METODIKASI» KAFEDRASI

ALGORITMLAR FANIDAN

LABORATORIYA MASHG‘ULOTLARI ISHLANMALARI

Navoiy-2019

201

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Ushbu laboratoriya mashg‘ulotlari ishlanmasi 5110700 - “Informatika o‘qitish metodikasi” ta’lim yo‘nalishi bakalavr talabalari uchun fan sifatida o‘qitilayotgan “*Algoritmlar*” fan dasturi asosida yaratilgan. Undagi barcha mavzular, mazkur fan bo‘yicha tavsiya etilgan adabiyotlar asosida olingan bo‘lib, zamonaviy fan yutuqlarini va pedagogik tajribani hisobga olgan holda ishlab chiqilgan, hamda bo‘lajak mutaxassis egallashi kerak bo‘lgan bilim va ko‘nikmalarni o‘z ichiga oladi.

Tuzuvchilar:

f.-m.f.n. dots. A.A. Ibragimov
k.o‘q. D.N.Xamroyeva
o‘q. S.Q.Absobirov

Ushbu laboratoriya mashg‘ulotlari ishlanmasi “Informatika o‘qitish metodikasi” kafedrasining 2019-yil ____-avgustdagi 1-sonli yig‘ilishida muhokamadan o‘tgan va ma’qullangan.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Talabalar tomonidan laboratoriya mashg‘ulotlarni bajarish uchun belgilangan tartiblar:

1. Laboratoriya xonalarida texnika havsizligi qoidalariga rioya qilish.
2. Laboratoriya ishini bajarish uchun zarur bo‘lgan nazariy ma'lumotlarni ko‘rsatilgan adabiyotlardan to‘plash;
3. Belgilangan variant bo‘yicha shaxsiy topshiriqlarni o‘z vaqtida olish;
4. Berilgan topshiriqlarni shaxsiy kompyuterdan foydalangan holda bajarish;
5. Bajarilgan ishlarni, belgilangan reja asosida rasmiylashtirish va uni o‘qituvchi huzurida himoya qilish (laboratoriya darsi mobaynida);
6. Himoya qilingan, o‘qituvchi tomonidan baholangan laboratoriya ishi hisobotini o‘qituvchiga topshirish;
7. Laboratoriya ishini o‘z vaqtida bajarmagan talaba o‘qituvchi belgilangan vaqtda(dars mashg‘ulotlaridan bo‘sh vaqtda) kafedraga kelib ishni bajaradi va belgilangan tartibda uni himoya qiladi;
8. Agar talaba dars mobaynida va qo‘shimcha belgilangan vaqtda ham laboratoriya ishini bajarmasa shu mavzu uchun reyting ballarini ololmaydi va fandan o‘zlashtirmagan hisoblanadi.

Talaba tomonidan tayyorlanadigan hisobot quyidagilarni o‘z ichiga olishi kerak:

1. Laboratoriya ishi mavzusi.
2. Ishning maqsadi va vazifasi.
3. Berilgan topshiriq matni.
4. Ishning bajarilishi.
5. Xulosa.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

1 - LABORATORIYA ISHI.

Mavzu: Algoritm tushunchasi va ulardan foydalanish.

Ishning maqsadi: Talabalarda algoritm xossalari, turlari va berilish usullari asosida turli algoritmlarni tahlil qilish ko‘nikma va malakalarini shakllantirish. Algoritm blok-sxemalarini tuzish amaliyotini o‘tkazish.

Vazifa:

1. Berilgan topshiriq bo‘yicha so‘zli algoritm tuzish;
2. Tuzilgan algoritmning xossalarini tahlil qilish;
3. Ushbu algoritm turini va berilish usulini tavsiflash.
4. Algoritm blok-sxemasida ishlatiladigan figuralarning vazifalari, hisoblash yo‘nalishlarini belgilash va chiziqli, tarmoqlanuvchi hamda takrorlanuvchi algoritmlar uchun blok-sxemalar yaratish malakasini hosil qilish.

Nazariy qism

Algoritm – bu qoidalarning qat‘iy va chekli sistemasi bo‘lib, ba‘zi ob‘ektlar ustida bajariladigan amallarni aniqlaydi va chekli qadamdan keyin qo‘yilgan maqsadga olib kelishini ta‘minlaydi.

Algoritm – qaralayotgan masalani yechish uchun bajarilishi lozim bo‘lgan amallar ketma-ketligi haqidagi aniq qoidalardir. Demak, biror masalani yechish uchun kerak bo‘ladigan aniq ko‘rsatmalar, buyruqlar, amallar ketma-ketligi uning algoritmini ifodalaydi.

Algoritm deyilganda - biror maqsadga erishishga yoki qandaydir masalani yechishga qaratilgan buyruqlarning aniq, tushunarli, chekli hamda to‘liq tizimi, aniq natijaga olib keladigan amallarning cheklangan ketma-ketligi tushuniladi.

Algoritm inson yoki avtomatik vosita – formal ijrochi tomonidan bajarish uchun mo‘ljallangan bo‘lishi mumkin.

Ijrochining vazifasi – mavjud algoritmni aniq bajarishdan iborat bo‘ladi. Formal ijrochiga esa algoritmning tub mohiyatini to‘liq anglab olish talabi qo‘yilmaydi va uni tushunmasligi ham mumkin. Masalan, kir yuvish – avtomat mashinasi, suvga poroshok solinmasa ham, o‘ziga qo‘yilgan vazifani so‘zsiz bajaraveradi.

Informatikada kompyuter algoritmnining universal ijrochisi bo‘lib hisoblanadi.

Ixtiyoriy **algoritm** 5 ta muhim xossalarga ega:

- ♦ **Tushunarligi** – algoritm ijrochi imkoniyatlariga moslangan holda, ya‘ni ijrochi uchun tushunarli tarzda bo‘lishi kerak.
- ♦ **Algoritmning aniqligi** – har bir qadam bajarilishining bir qiymatliligi.
- ♦ **Diskretligi** – masalani yechish jarayonini bajarilish vaqtida kompyuter yoki insonga qiyinchilik tug‘dirmasligi uchun bir necha sodda bosqichlar (bajarilish qadamlari)ga bo‘lish.
- ♦ **Ommaviylik** – belgilangan masalalar sinfini yechish uchun algoritmning foydaliligi.
- ♦ **Natijaviylik** – oxirgi qadamlarda dastlabki ma‘lumotlarga ega bo‘lgan kerakli natijani olishga imkon beruvchi algoritmning harakatlar yakuni.

Amaliyotda quyidagi **algoritm** turlari mavjud:

Chiziqli – amallar ketma-ket, biror-bir shart tekshirilmasdan bajariluvchi algoritm.

Tarmoqlanuvchi – belgilangan shartlarning o‘zgarishiga bog‘liq holda ko‘rsatmalarning variantlari oldindan mo‘ljallanadigan algoritm.

Takrorlanuvchi (Siklik) – alohida jarayonlar yoki jarayonlar guruhi bir necha marta bajariladigan algoritm.

Algoritmlar bir necha xil usullarda berilishi mumkin: so‘zli; formulali; jadvalli; dasturli, grafikli.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

- 1) **Algoritmning so‘z orqali berilishi.** Bunda ijrochi uchun beriladigan har bir ko‘rsatma so‘zlar orqali buyruq mazmunida beriladi.
- 2) **Algoritmning formulalar yordamida berilishi.** Algoritmning formulalar bilan berilish usulidan matematika, fizika, kimyo va boshqa aniq fanlarni o‘rganishda ko‘proq foydalaniladi.
- 3) **Algoritmning jadval ko‘rinishida berilishi.** Algoritmning jadval ko‘rinishidagi ifodalanishidan ham ko‘p foydalaniladi. Masalan, to‘rt xonalik matematik jadvallar yoki turli lotareya jadvallari. Funktsiyalarning grafiklarini chizishda ham algoritmning qiymatlari jadvali ko‘rinishlaridan foydalanamiz.
- 4) **Algoritmning dastur shaklida ifodalanishi.** Millionlab kompyuterlarning keng tarqalib ketishi algoritmning dastur tarzidagi tasvirning keng ommalashib ketishiga katta turtki bo‘ldi. Sababi shundaki, kompyuterlar doimo dasturlar yordamida boshqariladi.
- 5) **Algoritmning grafik (blok-sxema) shaklida tasvirlanishi.** Algoritmning blok-sxema ko‘rinishidagi tasvirida geometrik figuralar shaklidagi oddiy elementlardan foydalaniladi.

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

Topshiriq matni: Berilgan ikki a va b sonlar uchun Yevklid algoritmini tuzing.

I. Yevklid algoritmini yozamiz:

1. a sonni birinchi son deb, b sonni ikkinchi son deb qaralsin.
2. Birinchi va ikkinchi sonlarni taqqoslang. Agar ular teng bo‘lsa, 5-qadamga, aks holda 3-qadamga o‘tilsin.
3. Agar birinchi son ikkinchi sonda kichik bo‘lsa, ularning o‘rni almashtirilsin.
4. Birinchi sonda ikkinchi son ayirilsin va ayirma birinchi son deb hisoblansin. 2-qadamga o‘tilsin.
5. Birinchi sonni natija sifatida qabul qilinsin. Tamom.

II. Algoritmning xossalari bo‘yicha tahlili.

Algoritmning tushunariligi. Algoritm o‘z tabiatiga ko‘ra ishlash jarayoni sodda qadamlardan iborat. Buni ham yuqoridagi va boshqa misollardan ko‘rish mumkin.

Algoritmning to‘liq aniqlanganligi. Algoritmning kattaliklar sistemasining qiymatlari, o‘zidan oldingi qiymatlari orqali to‘liq aniqlanadi. Yuqoridagi misolda ko‘rganimizdek:

(b, r_1) qiymatlar (a, b) orqali to‘liq aniqlangan va h.k.

(r_{n-2}, r_{n-1}) esa (r_{n-3}, r_{n-2}) orqali ;

(r_{n-1}, r_n) esa (r_{n-2}, r_{n-1}) orqali ;

$(r_n, 0)$ esa (r_{n-1}, r_n) orqali to‘liq aniqlangan.

Algoritmning diskretligi. Har bir algoritm qandaydir miqdorlarning boshlang‘ich qiymatlarida ish boshlab, diskret rejimda ishlaydi. Ma’lum bir vaqt momentida miqdorlarning boshqa qiymatlariga o‘tadi.

Masalan, Yevklid algoritmidan foydalanib, a va b sonlarning eng katta umumiy bo‘luvchisini topaylik,

$$a = b \cdot q_0 + r_1 ;$$

$$b = r_1 \cdot q_1 + r_2 ;$$

$$r_1 = r_2 \cdot q_2 + r_3 ;$$

.....

$$r_{n-3} = r_{n-2} \cdot q_{n-2} + r_{n-1} ;$$

$$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n ;$$

$$r_{n-1} = r_n \cdot q_n .$$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Bundan $(a, b) = r_n$. Ko‘rinib turibdi-ki, a va b sonlarning eng katta umumiy bo‘luvchisini topishda (a, b) miqdorlarning boshlang‘ich qiymati, keyingi qiymati (b, r_1) va h.k. $(r_n, 0)$ miqdorlarning oxirgi qiymati bo‘ladi.

Algoritmning ommaviyligi. Bu haqda yuqorida aytganimizdek, har bir algoritm qandaydir masalalar sinfini yechishga mo‘ljallangandir. Ushbu algoritm ixtiyoriy ikkita butun musbat sonning eng katta umumiy bo‘luvchisini topadi, ya’ni algoritmni natural sonlar to‘plami elementlarining barchasiga qo‘llash mumkin.

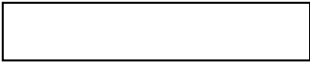
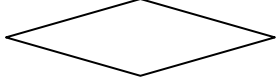
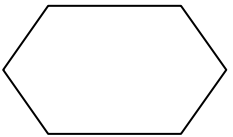

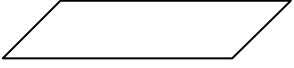


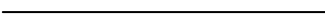

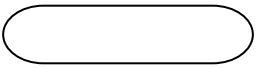
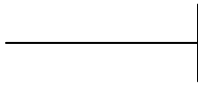
Algoritmning natijaviyligi. Miqdorlar qiymatlarini qurish jarayoni chekli qadamdan so‘ng natija berishi lozim. Qaralayotgan algoritmda, katta sonni 1-son deb olish va katta sondan kichigini ayirib borish orqali doim natijaga, yani EKUB ga ega bo‘lamiz.

III. Algoritmning turi va berilish usulini tavsiflash.

Algoritm bir qarashda chiziqlidek tuyuladi, lekin 2, 3-qadamlarda shart tekshirilish zarurati mavjud bo‘lganligi uchun uni taqmoqlanuvchi ekanligini aniqlaymiz. Tuzilgan algoritm so‘zli berilish usuliga mansub.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat:

Nomi	Figura	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarning bajarilishi natijasida ma'lumotlarning o'zgarishi
Qaror		Biror shartga bog'liq ravishda algoritmning bajarilish yo'nalishini tanlash
Shakl o'zgartirish		Dasturni o'zgartiruvchi buyruq yoki buyruqlar turkumini o'zgartirish amalini bajarish
Avval aniqlangan jarayon		Oldindan ishlab chiqilgan dastur (qism dastur) yoki algoritmdan foydalanish
Kiritish yoki chiqarish		Axborotlarni qayta ishlash mumkin bo'lgan shaklga o'tkazish yoki olingan natijani tasvirlash
Displey		EHMga ulangan displeydan axborotlarni kiritish yoki chikarish
Hujjat		Axborotlarni qog'ozga chiqarish yoki qog'ozdan kiritish
Axborotlar oqimi chizig'i		Bloklar orasidagi bog'lanishlarni tasvirlash
Bog'lagich		Uzilib qolgan axborot oqimlarini ulash belgisi
Boshlash yoki tugatish		Axborotni qayta ishlashni boshlash, vaqtincha yoki butunlay to'xtatish
Izoh		Bloklarga tegishli turli xildagi tushuntirishlar

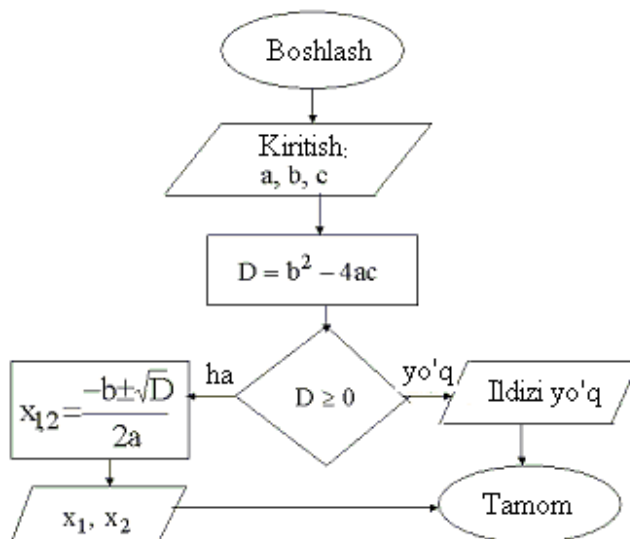
Blok-sxemalar bilan ishlashni yaxshilab o'zlashtirib olish zarur, chunki bu usul algoritmlarni ifodalashning qulay vositalaridan biri bo'lib dastur tuzishni osonlashtiradi, dasturlash qobiliyatini mustahkamlaydi. Algoritmik tillarda blok-sxemaning asosiy strukturalariga maxsus operatorlar mos keladi.

IV. Amaliy qism

Shuni aytish kerakki, blok-sxemalardagi yozuvlar odatdagi yozuvlardan katta farq qilmaydi. Misol sifatida $ax^2+bx+c=0$ kvadrat tenglamani yechish algoritmining blok-sxemasi quyida keltirilgan.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

1-rasm. Kvadrat tenglamani yechish algoritmi



Chiziqli algoritmlar. Har qanday murakkab algoritmni ham uchta asosiy struktura yordamida tasvirlash mumkin. Bular ketma-ketlik, ayri va takrorlash strukturalaridir. Bu strukturalar asosida chiziqli, tarmoqlanuvchi va takrorlanuvchi hisoblash jarayonlarining algoritmlarini tuzish mumkin. Umuman olganda, algoritmlarni shartli ravishda quyidagi turlarga ajratish mumkin:

chiziqli algoritmlar;

tarmoqlanuvchi algoritmlar;

takrorlanuvchi yoki siklik algoritmlar;

ichma-ich joylashgan siklik algoritmlar;

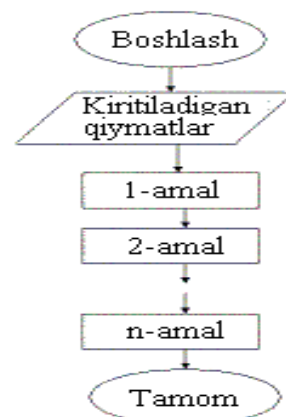
rekurrent algoritmlar;

takrorlanishlar soni oldindan no‘malum algoritmlar;

ketma-ket yaqinlashuvchi algoritmlar.

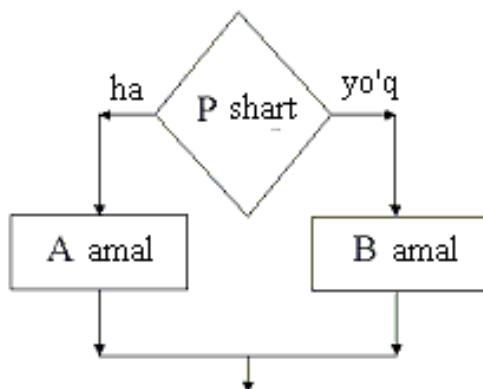
Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga-*chiziqli algoritmlar* deyiladi. Bunday algoritmni ifodalash uchun ketma-ketlik strukturasi ishlatiladi. Strukturada bajariladigan amal mos keluvchi shakl bilan ko‘rsatiladi. Chiziqli algoritmlar blok-sxemasining umumiy strukturasi quyidagi ko‘rinishda ifodalash mumkin:

2-rasm. Chiziqli algoritm blok-sxemasi.



Tarmoqlanuvchi algoritmlar. Agar hisoblash jarayoni biror bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo‘yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlariga tarmoqlanuvchi algoritmlar deyiladi. Tarmoqlanuvchi algoritmlar uchun ayri strukturasi ishlatiladi. Tarmoqlanuvchi strukturasi berilgan shartning bajarilishiga qarab ko‘rsatilgan tarmoqdan faqat bittasining bajarilishini ta‘minlaydi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



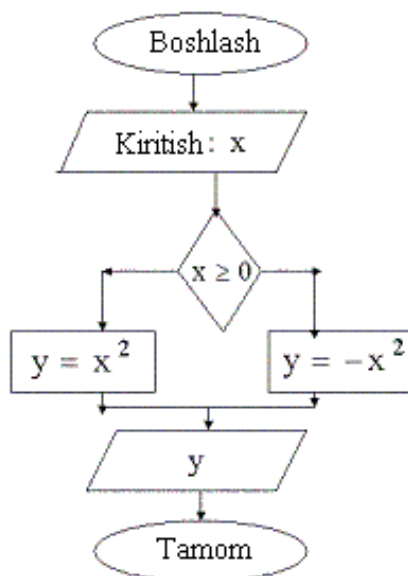
3-rasm. Tarmoqlanishning umumiy ko‘rinishi.

Berilgan shart romb orqali ifodalanadi, P - berilgan shart. Agar shart bajarilsa, "ha" tarmoq bo‘yicha A amal, shart bajarilmasa "yo‘q" tarmoq bo‘yicha B amal bajariladi.

Tarmoqlanuvchi algoritmgga tipik *misol* sifatida quyidagi sodda *misol*ni qaraylik.

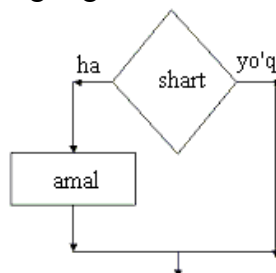
1- Misol:
$$Y = \begin{cases} x^2 & \text{agar } x \geq 0 \\ -x^2 & \text{agar } x < 0 \end{cases}$$

Berilgan x ning qiymatiga bog‘lik holda, agar u musbat bo‘lsa «ha» tarmoq bo‘yicha $y=x^2$ funksiyaning qiymati, aks holda $y=-x^2$ funksiyaning qiymati hisoblanadi.



4-rasm. Interval ko‘rinishidagi funksiya qiymatini hisoblash algoritmi

Ko‘pgina masalalarni yechishda, shart asosida tarmoqlanuvchi algoritmlarning ikkita tarmog‘idan bittasining, ya‘ni yoki «ha» yoki «yo‘q» ning bajarilishi yetarli bo‘ladi. Bu holat tarmoqlanuvchi algoritmning xususiy holi sifatida *aylanish strukturasi* deb atash mumkin. Aylanish strukturasi quyidagi ko‘rinishga ega:



5-rasm. Aylanish strukturasi umumiy ko‘rinishi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Takrorlanuvchi algoritmlar. Agar biror masalani yechish uchun tuzilgan, zarur bo‘lgan amallar ketma-ketligining ma’lum bir qismi biror parametrga bog‘liq ko‘p marta qayta bajarilsa, bunday algoritm *takrorlanuvchi algoritm* yoki *siklik algoritmlar* deyiladi. Takrorlanuvchi algoritmlarga tipik *misol* sifatida odatda qatorlarning yig‘indisi yoki ko‘patmasini hisoblash jarayonlarini qarash mumkin. Quyidagi yig‘indini hisoblash algoritmini tuzaylik.

$$S = 1 + 2 + 3 + \dots + N = \sum_{i=1}^N i$$

Bu yig‘indini hisoblash uchun $i=0$ da $S=0$ deb olamiz va $i=i+1$ da $S=S+i$ ni hisoblaymiz. Bu yerda birinchi va ikkinchi qadamlar uchun yig‘indi hisoblandi va keyingi qadamda i parametr yana bittaga orttiriladi va navbatdagi raqam avvalgi yig‘indi S ning ustiga qo‘shiladi va bu jarayon shu tartibda to $i < N$ sharti bajarilmaguncha davom ettiriladi va natijada izlangan yig‘indiga ega bo‘lamiz. Bu fikrlarni quyidagi algoritm sifatida ifodalash mumkin:

N –berilgan bo‘lsin,

$i=0$ berilsin,

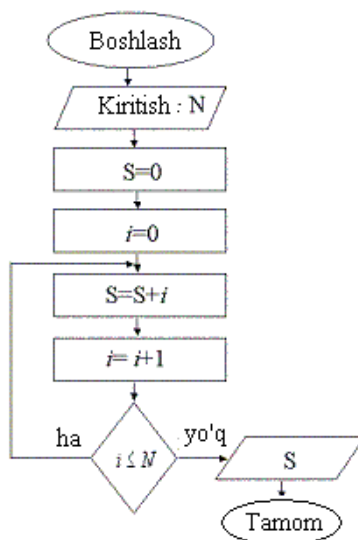
$S=0$ berilsin,

$i=i+1$ hisoblansin,

$S=S+i$ hisoblansin,

$i < N$ tekshirilsin va bu shart bajarilsa, 4-satrga qaytilsin, aks holda keyingi qatorga o‘tilsin,

S ning qiymati chop etilsin.

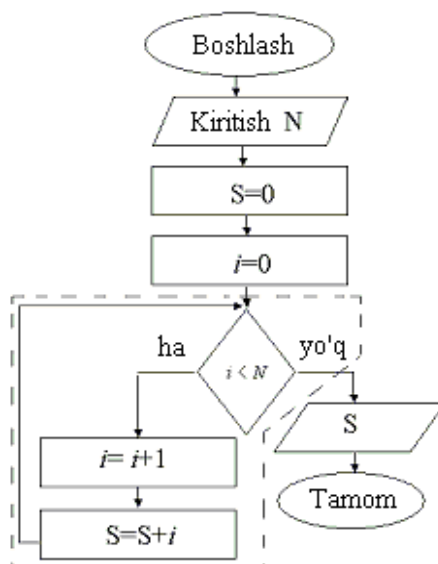


6-rasm. 1 dan n gacha bo‘lgan sonlar yig‘indisini hisoblash algoritmi.

Yuqorida keltirilgan algoritm va blok sxemadan ko‘rinib turibdiki amallar ketma-ketligining ma’lum qismi parametr i ga nisbatan N marta takrorlanayapti. ya’ni shartni oldin tekshiriladigan holatda chizish mumkin edi.

Masalan, yig‘indining algoritmini qaraylik. Bu blok sxemaning takrorlanuvchi qismiga quyidagi, sharti oldin berilgan siklik strukturaning mos kelishini ko‘rish mumkin.

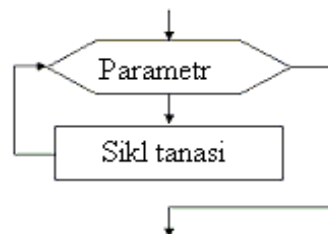
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



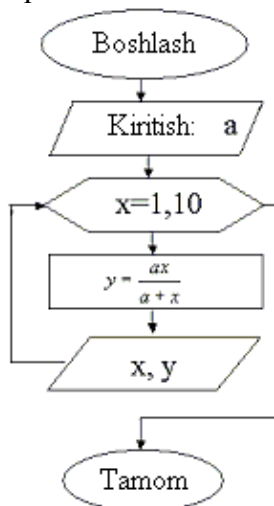
7-rasm. 1 dan n gacha bo‘lgan sonlar yig‘indisini hisoblash algoritmi.

Blok sxemalarining takrorlanuvchi qismlarini, quyidagi parametrlilik takrorlash strukturasi ko‘rinishida ham ifodalash mumkin.

8-rasm. Parametrlilik takrorlash operatorining umumiy ko‘rinishi.



Parametrlilik takrorlash operatoriga *misol* sifatida berilgan $x=1,2,3,\dots,10$ larda $y = \frac{ax}{a+x}$ funksiyasining qiymatlarini hisoblash blok sxemasini qarash mumkin.



9-rasm. Parametrlilik takrorlash operatoriga doir algoritim.

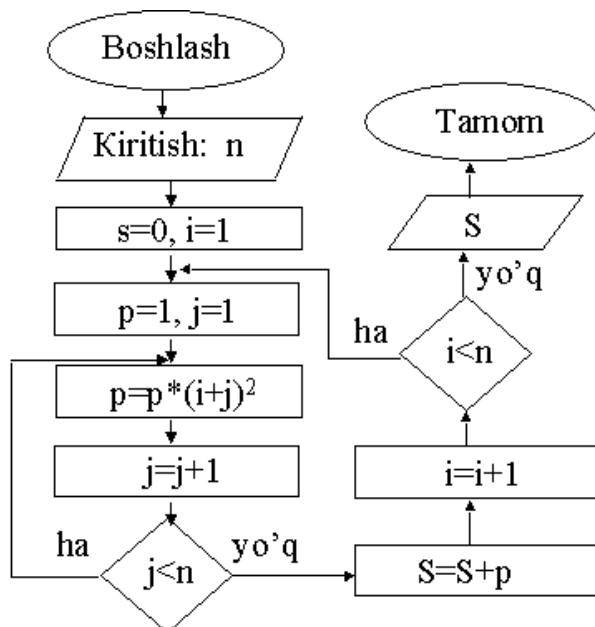
Ichma-ich joylashgan siklik algoritmlar. Ba’zan, takrorlanuvchi algoritmlar bir nechta parametrlarga bog‘liq bo‘ladi. Odatda bunday algoritmlarni ichma-ich joylashgan algoritmlar deb ataladi.

Misol sifatida berilgan $n \times m$ o‘lchovli a_{ij} –matritsa elementlarining yig‘indisini hisoblash masalasini qaraylik.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$S = \sum_{i=1}^n \prod_{j=1}^n (i+j)^2$ Bu yig‘indini hisoblash uchun, i ning har bir qiymatida j bo‘yicha

ko‘paytmani hisoblab, avval yig‘indi ustiga ketma-ket qo‘shib borish kerak bo‘ladi. Bu jarayon quyidagi blok–sxemada aks ettirilgan. Bu yerda i -tashqi sikl - yig‘indi uchun, j -esa ichki sikl-ko‘paytmani hosil qilish uchun foydalanilgan.

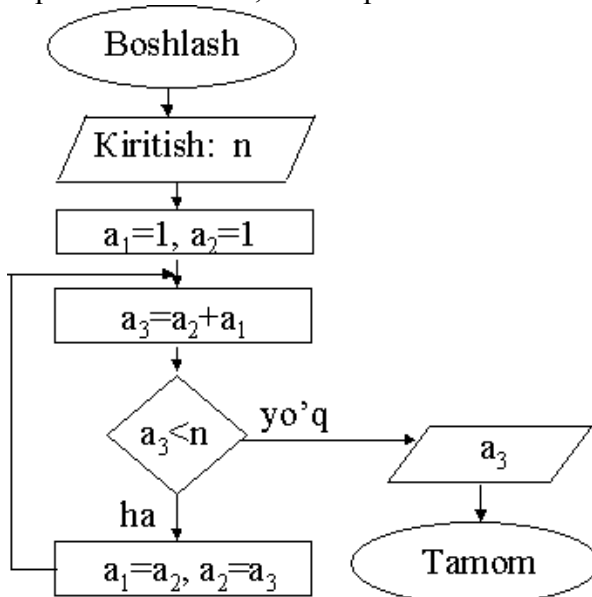


10-rasm. Ichma-ich joylashgan siklik algoritmgaga doir blok-sxema

Rekurrent algoritmlar. Hisoblash jarayonida ba‘zi bir algoritmlarning o‘ziga qayta murojaat qilishga to‘g‘ri keladi. O‘ziga–o‘zi murojaat qiladigan algoritmlarga *rekurrent algoritmlar* yoki *rekursiya* deb ataladi. Bunday algoritmgaga *misol* sifatida Fibonachchi sonlarini keltirish mumkin.

Ma‘lumki, Fibonachchi sonlari quyidagicha aniqlangan:

$a_0=a_1=1, a_i=a_{i-1}+a_{i-2} \quad i=2,3,4,\dots$ Bu rekurrent ifoda algoritmgaga mos keluvchi blok-sxema 11-rasmida keltirilgan. Eslatib o‘tamiz, formuladagi i -indeksga hojat yo‘q, agar Fibonachchi sonining nomerini ham aniqlash zarur bo‘lsa, birorta parametr-kalit kiritish kerak bo‘ladi.



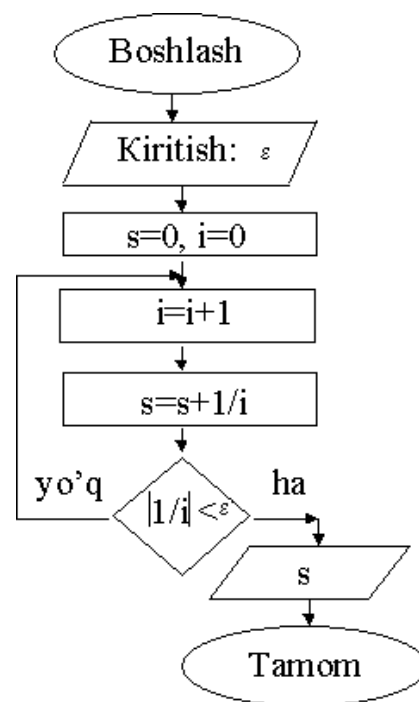
11-rasm. Fibonachchi sonlarining n - hadini hisoblash algoritmi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Amalda shunday masalalar uchraydiki, ularda takrorlanishlar soni oldindan berilmagan, ya’ni noma’lum bo‘ladi. Ammo, bu jarayonni tugatish uchun biror bir shart berilgan bo‘ladi.

Masalan, quyidagi $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots = \sum_{i=1}^{\infty} \frac{1}{i}$ qatorda nechta had bilan chegaralanish berilmagan. Lekin qatorni ε aniqlikda hisoblash zarur bo‘ladi. Buning uchun $\left| \frac{1}{i} \right| < \varepsilon$ shartni olish mumkin.

12-rasm. Takrorlanishlar soni oldindan no‘malum bo‘lgan algoritmlarga doir blok-sxema.



Ketma-ket yaqinlashuvchi yoki iteratsion algoritmlar.

Yuqori tartibli algebraik va transsendent tenglamalarni yechish usullari yoki algoritmlari ketma-ket yaqinlashuvchi – iteratsion algoritmlarga *misol* bo‘la oladi. Ma’lumki, transsendent tenglamalarni yechishning quyidagi asosiy usullari mavjud:

- *Urinmalar usuli (Nyuton usuli),*
- *Ketma-ket yaqinlashishi usuli,*
- *Vatarlar usuli,*
- *Kesmani teng ikkiga bo‘lish usuli.*

Bizga

$$f(x)=0 \tag{1}$$

transsendent tenglama berilgan bo‘lsin. Faraz qilaylik bu tenglama $[a,b]$ oraliqda uzluksiz va $f(a)*f(b)<0$ shartni qanoatlantirsin. Ma’lumki, bu holda berilgan tenglama $[a,b]$ oraliqda kamida bitta ildizga ega bo‘ladi va u quyidagi formula orqali topiladi.

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)}, \quad n = 0, 1, 2, \dots \tag{2}$$

Boshlang‘ich x_0 qiymat $f(x_0)f''(x_0) < 0$ shart asosida tanlab olinsa, (2) iteratsiya albatta yaqinlashadi. Ketma-ketlik

$$|X_{n+1} - X_n| < \varepsilon$$

shart bajarilgunga qadar davom ettiriladi.

Berilgan musbat a haqiqiy sondan kvadrat ildiz chiqarish algoritmi tuzilsin.

Bu masalani yechish uchun kvadrat ildizni x deb belgilab olib,

$$\sqrt{a} = x \tag{3}$$

ifodalash yozib olamiz. U holda (1) tenglamaga asosan

$$f(x) = x^2 - a \tag{4}$$

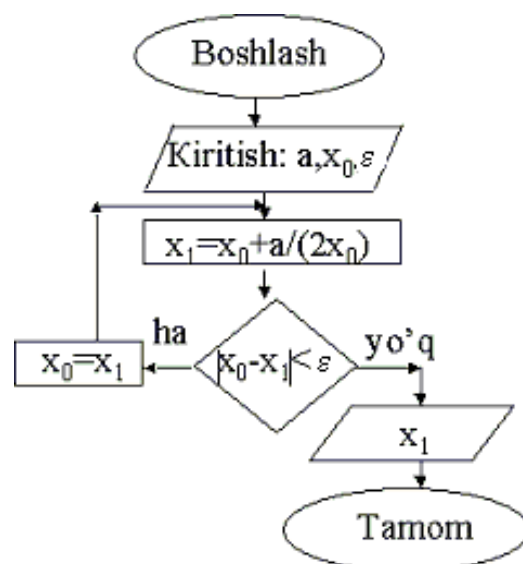
ekanligini topish mumkin. (4) ifodani (2) ga qo‘yib, quyidagi rekurrent formulani topish mumkin:

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{a}{2X_n} \right) \tag{5}$$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Bu formulaga mos blok-sxema 13-rasmda keltirilgan. ε - kvadrat ildizni topishning berilgan aniqligi. Eslatib o‘tamiz, algoritmda indeksli o‘zgaruvchilarga zarurat yo‘q.

13-rasm. Berilgan musbat a haqiqiy sondan kvadrat ildiz chiqarish algoritmi (iteratsion algoritimga doir blok-sxema).



LABORATORIYA ISHI UCHUN TOPSHIRIQLAR:

1. Algoritm blok-sxemasida ishlatiladigan figuralar va ularning ishlatilishini tavsiflang.
2. Chiziqli algoritm uchun blok-sxema tuzing:
 - a) $A = \frac{1 + \sin^2(x - 2y)}{x + y^2 + \cos x} + \operatorname{tg}^2 x$ $B = \cos\left(1 + \frac{Ax - y}{e^x + 10^2} - \sqrt[3]{A}\right)$;
 - b) $A = \ln(y - \sqrt{x} + e^{x+y}) + \sqrt[3]{x - y}$ $B = \left(x + \operatorname{tg} \frac{2\pi}{A}\right) \left(5 \cdot 10^{-6} + \left|\frac{x - 7}{A}\right|\right)$;
 - c) To‘g‘ri burchakli uchburchakning katetlari berilgan. Uchburchak gipotenuzasi va yuzasini hisoblang.
3. Tarmoqlanuvchi algoritm uchun blok-sxema tuzing:
 - a)
$$\begin{cases} m^2n + 1 - c, & \text{agar } n + 1 > 0 \\ (m + n)^2 + cm^2, & \text{agar } n + 1 \leq 0 \end{cases}$$
 - b)
$$\begin{cases} \frac{a^2 + b^2}{c} + \sqrt{a + x}, & \text{agar } x \geq 0 \\ \frac{\sin x + a}{a - b}, & \text{agar } x < 0 \end{cases}$$
4. Takrorlanuvchi algoritm uchun blok-sxema tuzing.

2 –LABORATORIYA ISHI.

Mavzu: Algoritm samaradorligini baholash.

Mashg‘ulotning maqsadi: talabalarga algoritm samaradorligini baholash bo‘yicha yetarli bilim berish va ko‘nikma hosil qilish.

NAZARIY QISM:

Kompyuterda masalani yechish tartiblari

1. Masalaning qo‘yilishi. (problem statement),(input) va (output)
2. Modelni yaratish.
3. Algoritmni ishlab chiqish. (algorithm design)
4. Algoritm to‘g‘riligini tekshirish.
5. Algoritmni amalga oshirish. (implementation)
6. Algoritmni va ularning murakkabligini tahlil qilish.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

7. Dasturni tekshirish.

8. Hujjatlashtirish.

1. **Misol.** Berilgan $A = (a_1, a_2, \dots, a_N)$ massivning eng katta qiymatini va uning nomerini aniqlang.

$$a_i \geq a_j, \quad \forall j \in \{1, 2, \dots, n\}.$$

Массив $a[0:24]$:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
180	155	193	176	159	169	172	195	201	160	167	180	177	174	179	187	166	193	183	175	169	170	157	199	187

$$a_i \geq a_j, \quad \forall j \in \{1, 2, \dots, n\}.$$

Массив $a[0:24]$:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
180	155	193	176	159	169	172	195	201	160	167	180	177	174	179	187	166	193	183	175	169	170	157	199	187



$i = 8$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
function Max(a[1..n])
    maxi = 1
    for i = 2 to n do
        if a[i] > a[maxi] then
            maxi = i
    end for
    return maxi
end function
```

TOPSHIRIQ

1. Berilgan $A(N)$ massivelementlarini $B(N)$ massivga skarif tartib bilan yozing.
2. $A(N)$ massiv berilgan. $B(N)$ massivni quyidagi formulayordamida hosil qiling: $b_i = (a_1 + a_2 + \dots + a_i) / i$.
3. Quyidagini hisoblang:
$$P = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + 3 \cdot 4 \cdot 5 \cdot 6 + \dots + N \cdot (N+1) \cdot \dots \cdot 2N.$$
4. Berilgan $X(N)$ massivning maksimal komponentdan (agar ular bir nechta bo'lsa, maksimal komponentni tartibi bo'yicha birinchisini oling) oldingi barcha manfiy komponentlarni nol bilan almashtiring.
5. Berilgan $X(N)$ massivning beshga karrali bo'lmagan barcha elementlari kvadratlari yig'indisini hisoblang.
6. Berilgan $X(N)$ massivda eng kichik va eng katta elementlarni almashtiring.
7. Berilgan $X(N)$ massivdagi berilgan qiymatdan kichiklarning sonini aniqlang.
8. Berilgan $A(N)$ massivning komponentlarini chapga bir pozitsiya siklik siljishni amalga oshiring, ya'ni $A = (a_2, a_3, \dots, a_N, a_1)$ vektorni hosil qiling.
9. Berilgan $A(N)$ massivning komponentlarini o'ngga ikki pozitsiya siklik siljishni amalga oshiring, ya'ni $A = (a_{N-1}, a_N, a_1, a_2, \dots, a_{N-2})$ vektorni hosil qiling.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3-LABORATORIYA MASHG‘ULOTI

Mavzu: Algoritmlar tahlili. Algoritmlarni ishlab chiqish metodlari.

Mashg‘ulotning maqsadi: Talabalarga algoritmlar tahlili, algoritmlarni ishlab chiqish metodlari bo‘yicha yetarli bilim berish va ko‘nikma hosil qilish.

NAZARIY QISM

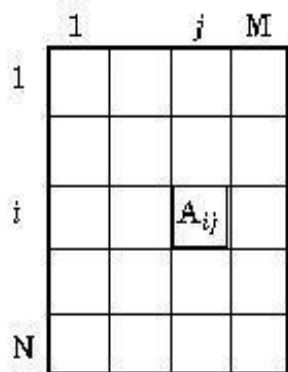
For (uchun) tipidagi ichma-ich joylashgan sikl yordamida bajariladigan algoritmlar va dasturlar

Til	For tipidagi ichma-ich joylashgan sikl sxemasi
Algoritmlar tilda ifodalanishi	sb i uchun A1 dan B1 gacha tashqi sikl tanasi sb j uchun A2 dan B2 gacha ichki sikl tanasi so so
Pascalda ifodalanishi	For i := A1 to B1 do begin For j := A2 to B2 do begin end; end;

Ichma-ich sikllar ko‘pincha matritsalar (ikki o‘lchovli massiv, to‘g‘ri to‘rtburchakli jadval) va vektorlarni (bir o‘lchovli massiv, chiziqli jadval) qayta ishlashda ishlatiladi:

$A(N,M)$ matritsa

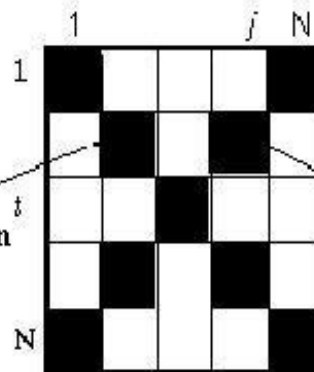
(to‘g‘ri to‘rtburchakli)



A_{ij} elementlar bilan matritsa ning bosh diagonali

$A(N,N)$ matritsa

(kvadrat)



$A_{i, N+1-i}$ elementlar bilan matritsa ning yon diagonali

1 - misol. Berilgan $A(N, M)$ matritsaning ustunlari elementlarning yig‘indisini hisoblang.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Test

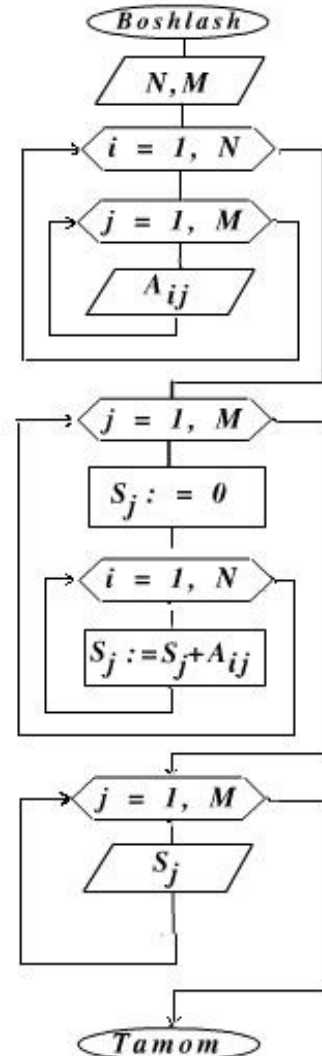
Berilganlar		Natija
N=2 M=2	$A = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}$	S=(6,4)

Algoritmi: Blok sxemasi:

alg Ustun_Yig (but N, M,
haq jad A[1:N, 1:M], **haqjad** S[1:M])
arg N,M,A
boshlbut i, j
sb j uchun 1 dan M gacha
 S[j]:=0
sb i uchun 1 dan N gacha
 S[j]:=S[j] + A[i, j]
so
so
tamom

Algoritmning bajarilishi

j	i	S[i]
1	1	S1=0 S1=0+2=2
	2	S1=2+4=6
2	1	S2=0 S2=0+1=1
	2	S2=1+3=4



2 - misol. Berilgan butun sonli A(N, M) matritsada maksimal qiymati necha marta uchrashishini aniqlang.

Test

Berilgan		Natija
N=2 M=3	$A = \begin{pmatrix} 1 & 2 & 5 \\ 5 & 1 & 5 \end{pmatrix}$	K=3

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Algoritmi:

alg Maks_soni(**but** N,M,K **but jad** A[1:N,1:M])

argN,M,A

natK

boshl but i, j, Amax

Amax := A[1, 1]

sb iuchun 1 dan N gacha

sb juchun 1 dan M gacha

agar A[i, j] > Amax

u holda Amax := A[i, j]

hal bo’ldi

so

so

K := 0

sbi uchun 1 dan N gacha

sb juchun 1 dan M gacha

agar A[i, j] = Amax

u holda K := K+1

hal bo’ldi

so

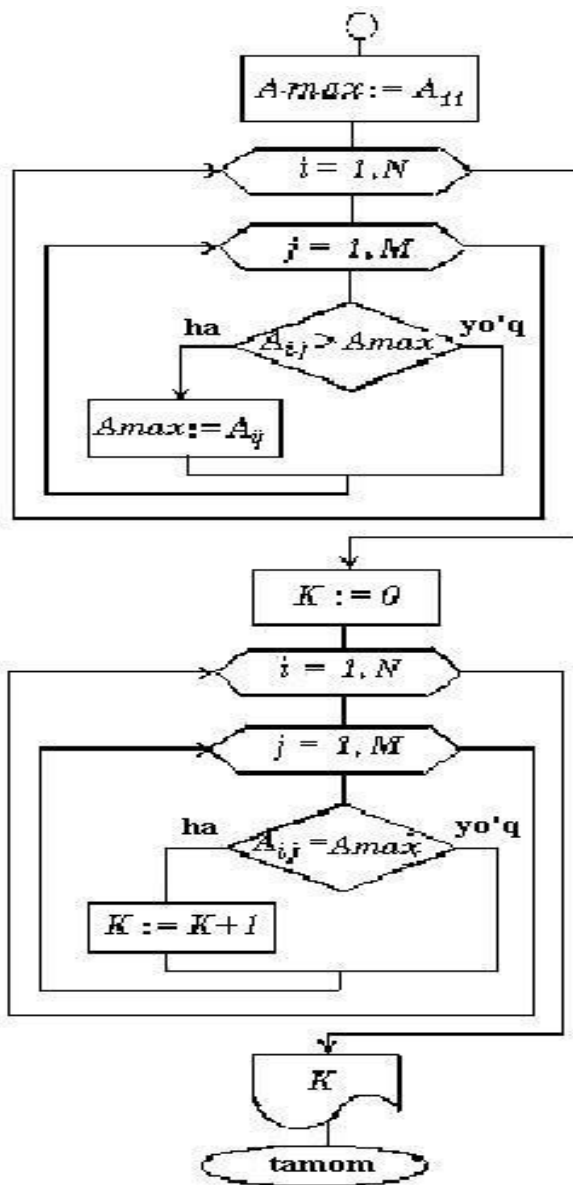
so

tamom

<u>Algoritmning bajarilishi (Amax ni topish)</u>				<u>(Amax necha marta uchrashini aniqlash)</u>			
i	j	A[i,j]>Amax	Amax	i	j	A[i,j]=Amax	K
	1		1	1	1	-	0
1	2	+	2		2	-	
	3	+	5		3	+	1
2	1	-		2	1	+	2
	2	-			2	-	
	3	-			3	+	3

BLOK-SXEMA

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



3 - misol. Berilgan $A(N, M)$ matritsaning P va Q satrlari o'rnini almashtiring ($1 \leq P \leq N, 1 \leq Q \leq N$).

Test	
Berilgan	Natija
$N=3 \ M=3 \ P=1 \ Q=3$ $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \\ 3 & 1 & 3 \end{pmatrix}$	$A = \begin{pmatrix} 3 & 1 & 3 \\ 2 & 2 & 2 \\ 1 & 2 & 1 \end{pmatrix}$

Algoritmi:

alg Almashtirish(**but** N, M, P, Q , **haq jad** $A[1:N, 1:M]$)

nat N, M, A, P, Q

natija A

boshl but j , **haq** $Temp$

sb juchun 1 dan M **gacha**

$Temp := A[P, j]; A[P, j] := A[Q, j]; A[Q, j] := Temp$

so

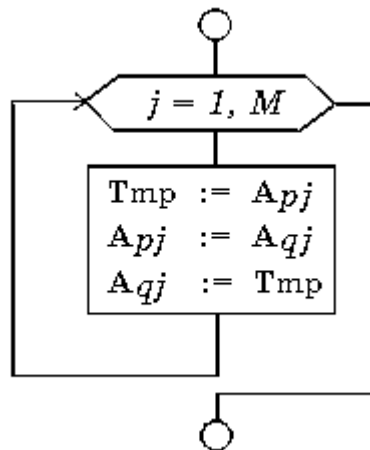
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

tamom

Algoritmning bajarilishi

J	Tmp	A[1,j]	A[3,j]
1	1	3	1
2	2	1	2
3	1	3	1

Blok-sxemasi fragmenti:



4 - misol. a_1, a_2, \dots, a_N ketma-ketlikni o'sish tartibida joylashtiring.

Test

Berilgan		Natija
N=4	A=(5, 2, 7, 1)	A=(1, 2, 5, 7)

Algoritmi:

alg O'sish (but N, haq jad A[1:N])

arg N,A

natija A

boshl but i, j, xak Tmp

sb i uchun 1 dan N-1 gacha

sb j uchun i+1 dan N gacha

agar $A[i] > A[j]$ u xolda $Tmp:=A[i]$;

$A[i]:=A[j]$; $A[j]:=Tmp$

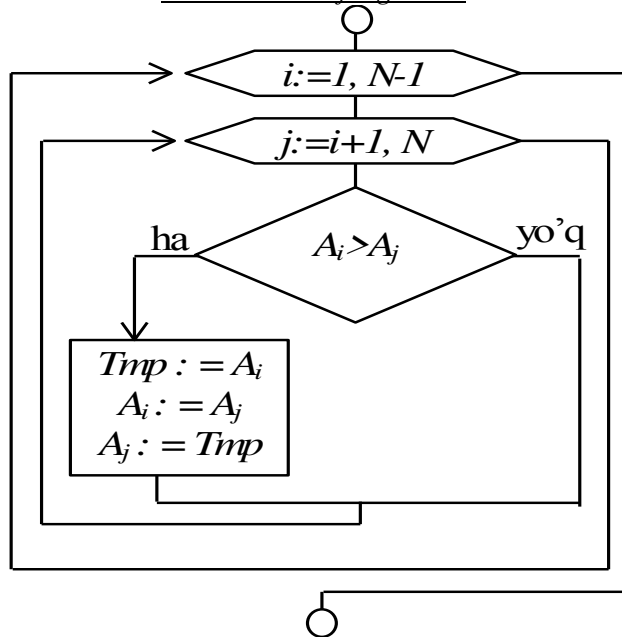
hal bo'ldi

so

so

tamom

Blok-sxemasi fragment:



Algoritmning bajarilishi

I	J	$A[i] > A[j]$	A massiv
1	2	+	2, 5, 7, 1
	3	-	
	4	+	1, 5, 7, 2
2	3	-	
	4	+	1, 2, 7, 5
3	4	+	1, 2, 5, 7

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

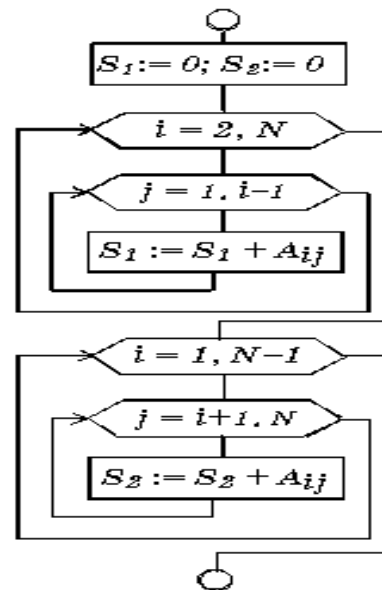
5-misol $A(N, N)$ massivdagi bosh digonaldan yuqorida joylashgan elementlar va pastda joylashgan elementlar yig'indilarini hisoblang.

Test	
Berilgan	Natija
$N=3$ $\begin{pmatrix} 1 & 2 & 4 \\ 3 & 1 & 3 \\ 2 & 1 & 2 \end{pmatrix}$	$S_1=6$ $S_2=9$

Algoritmi:

alg Yig' (but N ,
haq jad $A[1:N, 1:N]$, **haq** S_1, S_2)
arg N, A **natija** S_1, S_2
boshl but i, j
 $S_1:=0; S_2:=0$
sb i uchun 2 dan N gacha
sb j uchun 1 dan $i-1$ gacha
 $S_1:=S_1 + A[i, j]$
so
so
sb i uchun 1 dan $N-1$ gacha
sb j uchun $i+1$ dan N gacha
 $S_2:=S_2 + A[i, j]$
so
so
tamom

Blok-sxemasi fragmenti:



Algoritmnining bajarilishi

i	j	S1	S2
		0	0
2	1	0+3=3	
3	1	3+2=5	
	2	5+1=6	
1	2		0+2=2
	3		2+4=6
2	3		6+3=9

TOPSHIRIQ

1. Berilgan $A(N, M)$ matritsadagi eng katta elementni va u joylashgan satr hamda ustun nomerini toping.
2. Berilgan $A(N, M)$ matritsadagi har bir satr elementlarining yig'indisini, musbatlari o'rta arifmetigi va sonini hisoblang
3. Berilgan $A(N, M)$ butun sonli matritsaning elementlari yig'indisi juft son bo'ladimi, yo'kmi aniqlang.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

4. Berilgan $A(N, M)$ matritsada barcha elementlarining o‘rta arifmetigidan katta bo‘lgan elementlar sonini aniqlang.
5. Berilgan $A(N, M)$ butun sonli matritsaning toq qiymatli elementlarining yig‘indisi va ko‘paytmasini hisoblang.
6. $A(N, M)$ matritsa berilgan. $X(M)$ vektorni hisoblang, bu erda X_j qiymati A matritsaning j -chi usundagi musbat elementlar yig‘indisi.
7. $A(N, M)$ matritsa berilgan. $X(M)$ vektorni hosil qiling, u matritsaning P -satriga teng va $Y(N)$ vektorni hosil qiling, u matritsaning Q –ustiga teng.
8. Berilgan $A(N, M)$ matritsada eng katta va eng kichik elementlari o‘rnini almashtiring.
9. Berilgan n natural songa ko‘ra (n, n) tartibli ushbu shakl-dagi matritsalar hosil qiling.

A) $\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$	b) $\begin{pmatrix} n & n-1 & n-2 & \dots & 1 \\ 0 & n & n-1 & \dots & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & n \end{pmatrix}$	v) $\begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 2 & \dots & n-1 & n \end{pmatrix}$
---	---	---

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

4 –LABORATORIYA ISHI.

Mavzu: PASKAL TILI DASTURLASH TILINING ALIFBOSI, BUYRUQLAR TIZIMI VA OPERATORLARI.

Ishning maqsadi: Talabalarga Pascalda mantiqiy amallar mavzusi bo‘yicha yetarli bilim berish va ko‘nikma hosil qilish.

Qisqacha nazariy ma’lumot

Paskal - dasturning umumiy ko‘rinishini quyidagi ko‘rinishda yozib olaylikda, so‘ng har bir bo‘limni to‘laroq tahlil qilib chiqamiz:

Program <dastur ismi>;

label

<metkalar ro‘yxati>;

const

<o‘zgarmlar va ularning qiymatlari>;

type

<ma’lumotlarning yangi, nostandart turlarini aniqlash>;

var

<o‘zgaruvchilarni, protseduralar va funksiyalarni e’lon qilish>;

begin

<operatorlar bo‘limi>

end.

Arifmetik ifodalar va operatsiyalar

Arifmetik ifodalar, arifmetik operatsiyalar, butun sonli bo‘lish (DIV), (MOD) moduli bo‘yicha bo‘lish, arifmetik I (AND), chapga siljitish (ShL), unga siljitish (ShR), mantiqiy qo‘shish (OR), yoki (XOR), (+) saqlash xar operatsiyasi, (-) saqlash ular operatsiyasi, NOT operatsiyasi.

Arifmetik ifodalar butun va xaqiqiy qiymatlarni qaytaradi. Arifmetik ifodalarning eng sodda ko‘rinishlari quyidagicha:

belgisiz butun va xaqiqiy konstantalar;

butun va xaqiqiy o‘zgaruvchilar;

butun va xaqiqiy toifadagi massiv elementlari;

butun va xaqiqiy qiymatlarni qabul qiladigan funktsiyalar.

Massiv elementi va o‘zgaruvchisining qiymati arifmetik ifodada paydo bo‘lgunga qadar aniqlangan bo‘lishi kerak.

Arifmetik operatsiyalar ifodalarda butun va xaqiqiy toifali operatsiyalarda qiymatlari ustida arifmetik amallarni bajaradi. PascalABC tilining arifmetik operatsiyalari 1-jad.da keltirilgan.

Arifmetik operatsiyalar

Operatsiya	Amallar	Operandalar toifasi	Natija toifasi
+	Qo‘shish	Butun	Butun xaqiqiy Butun
-	Ayirish Ko‘paytirish	Xaqiqiy butun	xaqiqiy
*	Bo‘lish	Xaqiqiy butun	Butun xaqiqiy
/	Butun sonli bo‘lish	Xaqiqiy butun Butun	Butun
Div			Butun
Mod	sonli bo‘lish	Butun	
AND	Qoldiq	Butun	Butun
Shl	Arifmetik Va chapga	Butun	Butun
Shr	Surish O‘ngga Surish	Butun	Butun
Or	Arifmetik Yoki	Butun	Butun
Xor	Dizyunksiya Belgini	Butun	Butun
	Saqlash Belgini Inkor	Xaqiqiy Butun	Butun

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Unar + - Not	Etish Arifmetik Inkor	Xaqiqiy Butun Butun	Xaqiqiy Butun Xaqiqiy Butun Butun
--------------------	-----------------------	------------------------	---

Mantiqiy amallar

Operatsiya	Amallar	Ifoda	A	B	Natija
NOT	Mantiqiy inkor	NOT A	True		False
			False		True
AND	Mantiqiy VA	A AND B	True	True	True
			True	False	False
			False	False	False
			False	True	False
OR	Mantiqiy YOKI	A or B	True	True	True
			True	False	True
			False	True	True
			False	False	False
XOR	YOKI bekor qilish	AxorB	True	True	False
			True	False	True
			False	True	True
			False	False	False

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Paskal dasturlash tilida uchraydigan standart funksiyalar jadvali.

Funksiyani paskaldagi ifodasi	Funksiyaning matematik ifodasi	Argumentning turi	Funksiyaning turi
abs(x)	x	REAL, INTEGER	REAL, INTEGER
sqr(x)	x^2	REAL, INTEGER	REAL, INTEGER
sqrt(x)	\sqrt{x}	REAL, INTEGER	REAL, INTEGER
exp(x)	e^x	REAL, INTEGER	REAL REAL
ln(x)	$\ln x$	REAL, INTEGER	REAL, REAL
sin(x)	$\sin x$	REAL, INTEGER	REAL, REAL
cos(x)	$\cos x$	REAL, INTEGER	REAL, REAL
arctan(x)	$\arctg x$	REAL, INTEGER	REAL, REAL
round(x)	x ni yaxlitlash	REAL	INTEGER
trunc(x)	x ni butun qismini olish	REAL	INTEGER
pred(x)	x dan oldingi qiymatni olish.	INTEGER, CHAR BOOLEAN	INTEGER, CHAR BOOLEAN
succ(x)	x dan keyingi qiymatni olish.	INTEGER, CHAR BOOLEAN	INTEGER, CHAR BOOLEAN
a div b	A ni B ga bo‘lib butun qismini olish	INTEGER, INTEGER	INTEGER
a mod b	A ni B ga bo‘lib qoldig‘ini olish	INTEGER, INTE GER	INTEGER
chr(x)	x ni tartib nomeriga ko‘ra simvolini aniqlash	INTEGER	CHAR
ord(x)	x simvolini tartib nomerini aniqlash	CHAR	INTEGER
odd(x)	x ning toq yoki juftligini aniqlash	INTEGER x-toq, x-juft	BOOLEAN TRUE, FALSE

TOPSHIRIQ

Ishning tartibi

- 1-jadvalda keltirilgan matematik ko‘rinishdagi masalalarni paskal dasturlash tilida yozing.

№	Arifmetik ifoda
1.	$a = \frac{2\cos(x - \frac{\pi}{6})}{1/2 + \sin^2 y} \quad b = 1 + \frac{z^2}{3 + z^3/5}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

2.	$j = \left x^{y/x} - \sqrt[3]{y/x} \right \quad \psi = (y-x) \frac{y-z/(y-x)}{1+(y-x)^2}$
3.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}, \quad \varphi = x(\sin x^3 + \cos^2 y)$
4.	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }, \quad S = b \sin(at^2 \cos 2t) - 1$
5.	$\omega = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x$ $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2+b^2}}$
6.	$S = x^3 t g^2(x+b)^2 + \frac{a}{\sqrt{x+b}}, \quad Q = \frac{bx^2-a}{Ax}$
7.	$R = x^2(x+1)/b - \sin^2(x+a), \quad S = \sqrt{xb/a} + \cos^2(x+b)^3$
8.	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}, \quad Z = \frac{x^2}{A} + \cos(x+b)^2$
9.	$f = \sqrt[3]{mtgt} + c \sin t , \quad z = m \cos(bt \sin t) + c$
10.	$y = bt g^2 x - \frac{A}{\sin^2(x/a)}, \quad S = b \sin(at^2 \cos 2t) - 1$
11.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, \quad b = 1 + \frac{z^2}{3 + z^3/5}$
12.	$j = \left x^{y/x} - \sqrt[3]{y/x} \right , \quad \psi = (y-x) \frac{y-z/(y-x)}{1+(y-x)^2}$
13.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}, \quad \varphi = x(\sin x^3 + \cos^2 y)$
14.	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }, \quad S = b \sin(at^2 \cos 2t) - 1$
15.	$\omega = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x, \quad y = \cos^2 x^3 - \frac{x}{\sqrt{a^2+b^2}}$
16.	$S = x^3 t g^2(x+b)^2 + \frac{a}{\sqrt{x+b}}, \quad Q = \frac{bx^2-a}{Ax}$
17.	$R = x^2(x+1)/b - \sin^2(x+a), \quad S = \sqrt{xb/a} + \cos^2(x+b)^3$
18.	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}, \quad Z = \frac{x^2}{A} + \cos(x+b)^2$
19.	$f = \sqrt[3]{mtgt} + c \sin t , \quad z = m \cos(bt \sin t) + c$
20.	$y = bt g^2 x - \frac{A}{\sin^2(x/a)}, \quad S = b \sin(at^2 \cos 2t) - 1$
21.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, \quad b = 1 + \frac{z^2}{3 + z^3/5}$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

22.	$j = \left x^{y/x} - \sqrt[3]{\frac{y}{x}} \right $, $\psi = (y-x) \frac{y-z/(y-x)}{1+(y-x)^2}$
23.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$, $\varphi = x(\sin x^3 + \cos^2 y)$
24.	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }$, $S = b \sin(at^2 \cos 2t) - 1$
25.	$\omega = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2+b^2}}$
26.	$S = x^3 t g^2(x+b)^2 + \frac{a}{\sqrt{x+b}}$, $Q = \frac{bx^2 - a}{Ax}$
27.	$R = x^2(x+1)/b - \sin^2(x+a)$, $S = \sqrt{xb/a} + \cos^2(x+b)^3$
28.	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x+b)^2$
29.	$f = \sqrt[3]{mtgt + c \sin t }$, $z = m \cos(bt \sin t) + c$
30	$y = bt g^2 x - \frac{A}{\sin^2(x/a)}$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2+b^2}}$

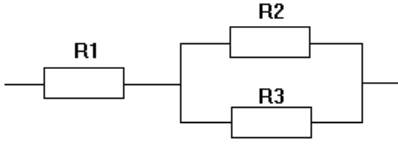
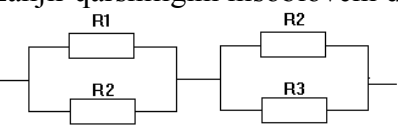
Hisobot mundarijasi

Ushbu laboratoriya mashg'ulotining hisobotida yuqorida keltirilgan ikkita mashq uchun dastlabki ma'lumotlar va natijalar o'z aksini topishi lozim.

1-jadval

Variant tartib raqami	Shart	Dastlabki ma'lumotlar
1	Koordinatalar tekisligida ikkita nuqta orasidagi masofani aniqlovchi dastur tuzing	x1, u1, X2, U2
2	To'rtta son berilgan. Shu sonlarning o'rtta arifmetigi va o'rtta geometr modulini hisoblovchi dastur tuzing.	a, b, c, d
3	Quyidagi formula bilan berilgan uchburchakka tashqi chizilgan aylanani radiusini hisoblovchi dastur tuzing: $R = \frac{a * b * c}{4 * \sqrt{p * (p - a) * (p - b) * (p - c)}}$ bunda r – uchburchakning yarim perimetri.	a, b, s
4	Uchburchakning a,b,c tomonlari berilgan. Shu uchburchakka ichki chizilgan aylanani radiusini aniqlovchi dastur tuzing: $r = \frac{\sqrt{(p - a) * (p - b) * (p - c)}}{p}$ bunda r – uchburchakning yarim perimetri.	a, b, s

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

5	Uchburchakning uchta burchagi va balandligi orqali uning yuzini hisoblovchi dastur tuzing: $S = \frac{h^2 * \sin(A)}{2 * \sin(B) * \sin(C)}$	A, B, C, h
6	Uchburchakning berilgan uchta burchaklari va tomoni orqali uning yuzini hisoblovchi dastur tuzing: $S = \frac{a^2 * \sin(B) * \sin(C)}{2 * \sin(A)}$	A, B, C, a
7	Zanjir qarshiligini hisoblovchi dastur tuzing: 	R1, R2, R3
8	Zanjir qarshiligini hisoblovchi dastur tuzing: 	R1, R2, R3
9	ABC uch burchakning ikkita tomoni va ular orasidagi burchagi berilgan. Shu uchburchakning qolgan uchinchi tomoni va ikkita burchagini aniqlovchi dastur tuzing $C^2 = A^2 + B^2 - 2 * A * B * \cos \alpha$	A, B, α
10	Savdo do'konida kostyum uchun gazlama sotilmoqda. Uning kv.metri so'mda baholangan. X mert uzunlikka va Y mert engaga ega bo'lgan gazlama narxini hisoblang.	X, U, b
11	Rombning tomoni va o'tkir burchagi berilgan uning yuzini hisoblang: $S = a^2 \sin \alpha$, bunda a – rombnig tomoni, α -esa o'tkir burchagi	a, α
12	Arifmetik progressiyaning birinchi hadi va ayirmasi berilgan. Uning 30-chi hadini va dastlabki 40 ta hadining yig'indisini toping.	a1, d
13	Kesmaning koordinatalari berilgan. Shu kesmaning o'rtasini koordinatalarni toping.	X1, U1, X2, U2
14	1 dan m gacha natural sonlar berilgan. Shu sonlarni arifmetik progressiya yordamida yig'indisini hisoblang.	

5 - LABORATORIYA ISHI.

Mavzu. Chiziqli va tarmoqlanuvchi jarayonlarni dasturlash

Ishning maqsadi: Pascal dasturlash muhitida chiziqli, tarmoqlanish operatorlari bilan tanishish, dastur tahlilini o‘tkazish, shartsiz va shartli operatorlarning yozilishi va ularning qo‘llanishini o‘rganish, ushbu amallarni bajarish bo‘yicha ko‘nikma hosil qilish.

Vositalar: Pascal dasturlash tili o‘rnatilgan kompyuterlar.

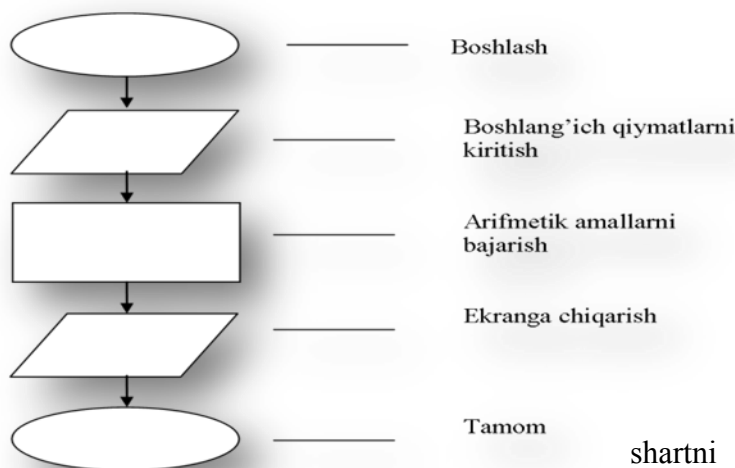
Mashg‘ulotning borishi.

1.Nazariy material bilan tanishish.

Chiziqli dastur tuzish uchun quyidagi operatorlar zarur bo‘ladi. := - o‘zlashtirish operatori, read(a) va readln(a) - kiritish operatorlari write(a) va writeln(a) - chiqarish operatori.

Chiziqli dastur tuzish uchun quyidagilarni eslab olish kerak:

- Chiziqli dasturda buyruqlar ketma-ket bajariladi.
- Chiziqli dastur algoritmi quyidagicha ko‘rinish olishi mumkin:



Ma‘lum bir shartni bajarilishi yoki bajarilmasligiga qarab, tarmoqlanuvchi jarayon holatlari aniqlanadi. Tarmoqlanuvchi jarayonlarni hisoblash uchun shartli operatoridan foydalaniladi. Shartli operatori ikki xil ko‘rinishda bo‘ladi.

- to‘liq shartli operator;
- chala shartli operator.

To‘liq shartli operator quyidagi formada yoziladi:

if <mantiqiy ifoda> then <operator> else <operator>

bu erda *if* (agar), *then* (u xolda), *else* (aks xolda) xizmatchi so‘zlar. Shunday qilib, to‘liq shartli operatorni quyidagicha yozish mumkin:

if S then S1 else S2; bu erda S - mantiqiy ifoda;

S1 – S mantiqiy ifoda rost qiymat qabul qilganda ishlovchi operator;

S2 - S mantiqiy ifoda yolg‘on qiymat qabul qilganda ishlovchi operator.

Shartli operatorning bajarilishi unda yozilgan S1 yoki S2 operatorlaridan birini bajarilishiga olib keladi, ya‘ni agar S mantiqiy ifoda bajarilishidan so‘ng *true* (rost) qiymati hosil bo‘lsa S1 operatori, aks holda esa S2 operatori bajariladi.

Chala (to‘liqmas) shartli operatorning yozilishini quyidagicha ifodalanadi:

if S then S1;

bu erda S - mantiqiy ifoda, S1 - operator.

Agar S ifoda qiymati *true* (rost) bo‘lsa S1 operatori bajariladi, aks holda esa boshqarish shartli operatoridan keyin yozilgan operatorga uzatiladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

2. Tayyor tuzilgan dasturlarning tahlili.

1-Misol. Paralleloipedning hajmini hisoblash dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko‘rsatilgan. (Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shriftlarda ko‘rsatilgan).

Paralleloipedning hajmini hisoblash.

Eni (sm): **7.5**

Bo‘yi (sm): **2.5**

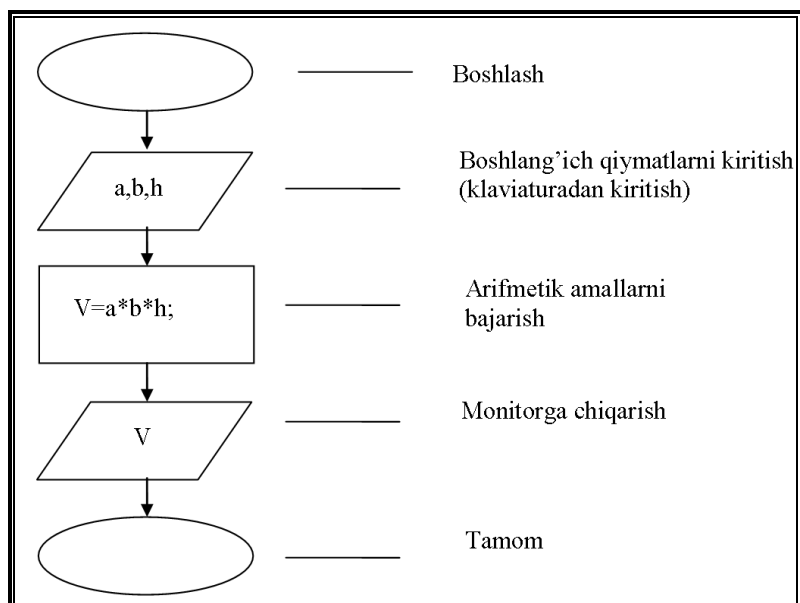
Balandligi (sm): **3**

Paralleloiped hajmi: 56.25 kub.sm

Ushbu masalani yechish uchun quyidagi bosqichlardan o‘tish kerak.

- Matematik formula.
- Algoritm tuzish.
- Dastur tuzish.

$V=a*b*h$ formuladan foydalaniladi.



<pre>Program M_1; Var V,a,b,h:real; Begin Writeln('Paralleloipedning hajmini hisoblash. '); Write('Eni (sm): '); Readln(a); {kiritish operatori} Write('Bo‘yi (sm): ');</pre>	<pre>Readln(b); Write('Balandligi (sm): '); Readln(h); V:=a*b*h; Writeln('paralleloipedning hajmi: ',V:3:2,' kub.sm'); Readln; end.</pre>
---	---

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Tayyor tuzilgan dasturlarning tahlili.

1-Misol.

$$y = \begin{cases} |x|, & \text{agar } x < 0 \text{ bo'lsa} \\ e^x, & \text{agar } x = 0 \\ x, & \text{agar } x > 0 \text{ bo'lsa} \end{cases}$$

Ushbu funksiya qiymatini hisoblash algoritmi va dasturini tuzing. Masalani yechishning hisoblash jarayoniga mos algoritmini keltiramiz. Bu masalani hisoblash jarayoniga mos dasturni ikki usulda keltiramiz.

1- Usul.	2- Usul.
<pre>var x,y:real; begin writeln('xni qiymatini kirit:'); readln(x); if x=0 then y:=exp(x) else if x>0 then y:=x else y:=abs(x); writeln('NATIJA y=',y:1:1); end.</pre>	<pre>var x,y:real; label a,b,c; begin writeln('xni qiymatini kirit:'); readln(x); if x=0 then goto a; if x<0 then goto b; y:=x; goto c; a: y:=exp(x); goto c; b: y:=abs(x); goto c; c:writeln('NATIJA y=',y:1:1); end.</pre>

5-LABORATORIYA ISHI UCHUN TOPSHIRIQLAR:

1-Misol. Paralleloipedning hajmini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko‘rsatilgan.

(Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shrift bilan ko‘rsatilgan).

Paralleloiped hajmini hisoblash.

Qiymatlarni kiriting:

Bo‘yi (sm) -> **9**

Eni (sm) -> **7.5**

Balandligi (sm)->**5**

Hajmi:337.50 kub.sm.

2-Misol. Kubni hajmini hisoblash dasturi tuzilsin. (Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shrift bilan ko‘rsatilgan).

Kubni hajmini hisoblash dasturi.

Kubni tomoni uzunligini (sm) kiriting va <Enter> ni bosing.-> **9.5**

Kub hajmi: 857.38 kub.sm

3-Misol. Bir nechta daftar va qalamni sotib olingandagi harajatni hisoblash dasturi tuzilsin.

Quyida dasturni ishlash jarayoni ko‘rsatilgan.

Harajatni hisoblash:

Daftar narxi (so‘m)-> **100**

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Daftarlar soni -> **5**

Qalam narxi (so‘m)->**50**

Qalamlar soni -> **3**

Jami harajat: 650 so‘m

Dasturni tugatish uchun <Enter> ni bosing.

4-Misol. Bir necha kilogrammdan iborat olmani narxini hisoblash dasturi tuzilsin.

Quyida dasturni ishlash jarayoni ko‘rsatilgan.

Harajatni hisoblash dasturi.

Boshlang‘ich qiymatlarni kiriting:

Bir kg olma narxi(so‘m) -> **425**

Necha kg -> **2.5**

Harajat: 1062.5

5-Misol. Agar uchburchakning ikki tomoni va ular orasidagi burchak ma‘lum bo‘lsa, uchburchak yuzini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko‘rsatilgan.

Uchburchak yuzini topish:

Bit satrda probel orqali uchburchakni ikki tomonini (sm) kiriting->**25 17**

Ular orasidagi burchak-> **30**

Uchburchak yuzini: 106.25 sm.kv.

(Vazifalarda F(x)-funksiya talaba tomonidan mustaqil aniqlanadi, masalan, F(x)=sinx)

$$1. \ y = \begin{cases} 0, & \text{agar } -2 < x < 2 \text{ bo'lsa} \\ 4, & \text{aks holda} \end{cases}$$

$$2. \ y = \begin{cases} x^2 + 4x + 5, & \text{agar } x < 2 \text{ bo'lsa} \\ \frac{41}{x^2 + 4x + 5}, & \text{aks holda.} \end{cases}$$

$$3. \ y = \begin{cases} 1, & \text{agar } |x| \geq 1 \text{ bo'lsa} \\ |x|, & \text{agar } -1 < x < 1 \end{cases}$$

$$4. \ y = \begin{cases} 0, & \text{agar } x < 0 \text{ bo'lsa} \\ x, & \text{agar } 0 < x < 1 \text{ bo'lsa} \\ x^4, & \text{aks holda.} \end{cases}$$

$$5. \ y = \begin{cases} 0, & \text{agar } x < 0 \text{ bo'lsa} \\ x^2 - x, & \text{agar } 0 < x < 1 \text{ bo'lsa} \\ x^4 - \sin(x^2 - 1), & \text{aks holda.} \end{cases}$$

$$6. \ b = \begin{cases} \ln(f(x)) + (f(x)^2 + y)^3, & x/y > 0 \\ \ln|f(x)/y| + (f(x) + y)^3, & x/y < 0 \\ (f(x)^2 + y)^3, & x = 0 \\ 0, & y = 0. \end{cases}$$

$$7. \ c = \begin{cases} f(x)^2 + y^2 + \sin(y), & x - y = 0 \\ (f(x) - y)^2 + \cos(y), & x - y > 0 \\ (y - f(x))^2 + \operatorname{tg}(y), & x - y < 0. \end{cases}$$

8. Foydalanuvchi tomonidan kiritilgan ikkita sonni solishtiruvchi dasturni tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

9. Uchta haqiqiy son berilgan. Manfiy sonlarni kvadrat darajaga, musbatlarini esa kub darajaga ko‘taring.
10. Ixtiyoriy uchta a , b va s sonlarni kamayish tartibida joylashtiring
11. Foydalanuvchi tomonidan kiritilgan butun sonni 3 ga bo‘linishini aniqlovchi dasturni tuzing
12. Foydalanuvchi tomonidan kiritilgan butun sonni juft toqligini aniqlash dasturi tuzilsin
13. $A(x_1, u_1)$ va (x_2, u_2) nuqtalari berilgan. Qaysi biri koordinata boshiga yaqin joylashganini aniqlang.
14. Agar 3 ta har xil x , y , z butun sonlar yig‘indisi 1 dan kichik bo‘lsa, u holda bu uchta sondan eng kichigini qolgan ikkitasining yarim yig‘indilari bilan almashtiring aks holda x va y lardan kichigini qolgan ikkitasi yarim yig‘indilari kvadrati bilan almashtiring.
15. $M(x,y)$ nuqtaning markazi va raduisi berilgan aylanaga tegishlilikini aniqlovchi dastur yarating.

6- LABORATORIYA ISHI

MAVZU: TAKRORLANUVCHI JARAYONLARGA DOIR MASALALAR VA ULARNING HISOBLASH DASTURLARINI YARATISH.

Ishning maqsadi: Talabalarga sikl operatorlari yordamida dasturlar tuzish bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

Vositalar: Pascal dasturlash tili o‘rnatilgan kompyuterlar.

Mashg‘ulotning borishi.

2. Nazariy material bilan tanishish.

Ma‘lum bir shart asosida algoritmda bir necha marta takrorlanish yuz beradigan jarayonlar ham ko‘plab uchraydi. Masalan, yil fasllarining har yili bir xilda takrorlanib kelishi, har haftada bo‘ladigan darslarning kunlar bo‘yicha takrorlanishi va hokazo. Demak, takrorlanuvchi algoritmlar deb shunday algoritmlarga aytiladiki, unda bir yoki bir necha amallar ketma-ketligi bir necha marta takrorlanadi, bu ketma-ketlik tarmoqlardan iborat bo‘lishi ham mumkin. Bundan chiziqli va tarmoqlanuvchi algoritmlar takrorlanuvchi algoritmlarning xususiy holi ekanligi kelib chiqadi.

Murakkab jarayonlarni dasturlashda ko‘pincha buyruqlar ketma-ketligini ma‘lum shartlar asosida qayta-qayta bajarishda to‘g‘ri keladi. Ma‘lum bir o‘zgaruvchining turli qiymatlarida ma‘lum buyruqlar tizimining biron-bir qonuniyatga asosan qayta-qayta bajarilishi takrorlanuvchi hisoblash jarayoni — *sikl* deb ataladi.

Takrorlanuvchi hisoblash jarayonining takror-takror hisoblanadigan qismini takrorlanishning *tanasi (jismi)* deb ataladi.

Takrorlanish ichida qiymatlari o‘zgarib boradigan o‘zgaruvchini takrorlanish o‘zgaruvchisi yoki takrorlanishni boshqaruvchi o‘zgaruvchisi (*sikl parametri*) deb yuritiladi.

Takrorlanuvchi jarayonning algoritmi umumiy holda qo‘yidagilarni o‘z ichiga olishi kerak:

1. Takrorlanishni tayyorlash – takrorlanishni boshlashdan oldin takrorlanishda qatnashadigan o‘zgaruvchilarning boshlang‘ich qiymatlari yoki takrorlanish o‘zgaruvchisining boshlang‘ich qiymati o‘rnatiladi, takrorlanish o‘zgaruvchisining o‘zgarish qadami belgilanadi.
2. Takrorlanish tanasi — takrorlanish o‘zgaruvchilarining turli qiymatlari uchun takror bajariladigan amallar ketma-ketligi ko‘rsatiladi.
3. Takrorlanish o‘zgaruvchisiga yangi qiymat berish — har bir takrorlanishdan avval o‘zgaruvchiga o‘zgarish qadamiga mos ravishda yangi qiymat beriladi.
4. Takrorlanishni boshqarish — takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga o‘tish ko‘rsatiladi.

Paskal algoritmik tilida uch xil ko‘rinishda takrorlanuvchi hisoblash jarayonini tashkil qilish mumkin va bu jarayonlarni dasturlash uchun maxsus operatorlar belgilangan:

- sharti avval tekshiriladigan (sharti oldin kelgan «toki») takrorlanish jarayoni, bu maxsus WHILE operatori orqali amalga oshiriladi;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

- sharti keyin tekshiriladigan («...gacha») takrorlanish jarayoni, bu jarayonni maxsus REPEAT operatori orqali amalga oshiriladi;
- parametrli takrorlanish jarayoni, bu jarayon maxsus FOR operatori yordamida amalga oshiriladi.

Yuqorida keltirilgan jarayonlar uchun laboratoriya ishlarini alohida-alohida ko`rib chiqamiz.

While B do S; { takrorlanuvchi operatori. }

Bu yerda mantiqiy **B** - mantiqiy ifoda, **S** - ixtiyoriy ifoda yoki operatorlar majmuasi. Bu operator “... bo‘lguncha bajar ...” ma’nosini bildiradi, ya’ni **while** operatoridan keyingi mantiqiy ifoda **B** yolg‘on qiymat qabul qilgunicha **do** operatoridan keyingi qaralayotgan operatorlar tanasi hisoblanmish **S** qaytarilishi davom etaveradi.

REPEAT S UNTIL B; Bu yerda avval **S** operator tanasi bajariladi, uning tugashini **B** mantiqiy ifoda aniqlab keladi. Agar **B** mantiqiy ifoda yo‘lg‘on qiymat qabul qilsa, takrorlanish jarayoni to‘xtaydi.

Parametri takrorlanuvchi operatorlar.

Takrorlanishlar soni avvaldan ma’lum bo‘lganda parametrli takrorlanish operatorlaridan foydalaniladi. Umumiy ko‘rinishi:

FOR <takrorlanish parametri>:=<boshlang‘ich qiymat> to <oxirgi qiymat> DO

Bu yerda:

FOR, DO – Pascal tilidagi maxsus so‘zlar;

<takrorlanish parametri> - integer turidagi o‘zgaruvchi;

<boshlang‘ich qiymat> - takrorlanish parametrining boshlang‘ich qiymati;

<oxirgi qiymat> - takrorlanish parametrining oxirgi qiymati;

For i:=1 to 10 do

write(i, ' ');

dastur qismi *1 2 3 4 5 6 7 8 9 10* natijasini beradi. i takrorlanish operatori 1 dan 10 gacha 1 qadam bilan o‘sib boradi.

For i:=10 downto 1 do

write(i, ' ');

dastur qismi *10 9 8 7 6 5 4 3 2 1* natijasini beradi. i takrorlanish operatori 10 dan 1 gacha 1 qadam bilan kamayib boradi.

Agar takrorlanish jarayonida amaldan soni bittadan ortiq bo‘lsa, begin ... end; operatorlari ichiga joylashtiriladi.

k:=0;

For i:=1 to 5 do

begin

k:=k+i;

write(k, ' ');

end;

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

1-misol. $1+1/2+1/3+\dots+1/n$ ifodani qiymatini hisoblash dasturini tuzing.

Program M_4;

var i, n: Byte;

S: real;

begin

readln(n);

S:=0; i:=1;

repeat

S:= S+1/i;

i:=i+1;

until i>n;

writeln (S);

end.

2-misol

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$$P = \prod_{k=2}^{10} (k + 2) \text{ yoki } P=3*4*5*6*7*8*9*10*11*12 \text{ ifoda qiymatini hisoblovchi}$$

dastur tuzing.

var

k: byte;

p: longint;

begin

p:=1;

for k:=2 to 10 do

begin

p:=p*(k+1);

write (p, ' ');

end; end.

3-misol. Berilgan 523^{***} soninig oxirgi uchta raqami shunday tanlash lozimki, hosil bo'lgan olti xonali son 7 ga, 8ga va 9ga bo'linsin.

DASTUR

program mlad;

var s,a,b,c,n: integer;

begin

for n:=0 to 999 do

begin

s:=523000+n;

a:= s mod 7;

b:= s mod 8;

c:= s mod 9;

if (a=0) and (b=0) and (c=0) then writeln ('n=',n:5)

end

end.

Ctrl+F9

Alt+F5

4-misol. $s=1-2+3-4+5- \dots$ qator yig'ingisini berilgan hadlar uchun hisoblang.

DASTUR

program sum;

var i,n:integer; {qo`shiluvcnilar soni va hisobchisi}

s:real; {qator yig`indisi}

begin

writeln('Nechta qo`shiluvchi ?');

readln(n);

if n<=0 then

writeln(n ga qayt qiymat berilsin')

else begin

s:=0;

for i:=1 to n do

if odd(i) then {tartib raqamning juftligini tekshirish}

s:=s+i

else

s:=s-i;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
writeln('s=',s:10:3);  
end;  
end.
```

3-misol. Ikkita natural sonning EKUB (eng katta umumiy bo‘luvchi) uchun dasturi.

```
program Evklid;  
var a,b,x,y,z: integer;  
begin  
  writeln ('Soni kiriting');  
  readln (a,b);  
  x:=a; y:=b;  
  while (x < >y) do  
    if x > y  
      then x:=x-y  
      else y:=y-x;  
  z:=x;  
  writeln ('EKUB=',z)  
end.
```

5-misol. $s = \frac{1}{2*3} + \frac{2}{3*4} + \frac{3}{4*5} + \dots$ qator yig‘indisini xadlarini e aniqlikgacha hisoblang.

DASTUR

```
program Summa;  
var n: integer;  
s,z,e: real;  
begin  
  writeln ('aniqlikgacha kiriting');  
  readln (e);  
  n:=1; z:=1/6; s:=0;  
  While abs(z) >=e do  
    begin  
      s:=s+z;  
      n:=n+1;  
      z:=n/((n+1)*(n+2))  
    end;  
  writeln ('s=', s:10:2)  
end.
```

TOPSHIRIQ

ISHNING TARTIBI

- 1-misol (2-jadval) uchun algoritm, blok sxema va dastur tuzish. Dasturni ishga tushirish va natijalarni tahlil qilish.
2. 3-jadvalda berilgan qatorlarni yigindisini hisoblash algoritmi va dasturini tuzish. Natijalar olish.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

HISOBOT MUNDARIJASI

Ushbu laboratoriya mashg‘uloti ikkita topshiriq bo‘yicha o‘zida shartlarni, dastur matnlarini, dastlabki ma‘lumotlarni va har bir dasturdan olingan natijalarni aks ettirishi lozim.

2-JADVAL

Variant tartib raqami	Shart
1	Raqamlari yigindisining kubiga teng bo‘lgan barcha uch xonali sonlarni toping.
2	2,3,4,5,6,7,8,9 sonlariga ko‘paytirilganda raqmlarining yig‘indisi o‘zgamaydigan ikki xonali sonlarni toping.
3	Raqamlari yig‘indisi berilgan butun songa teng uch xonali barcha sonlarni toping.
4	Barcha shunday uch xonali sonlarni toping-ki, ularni kvadratga ko‘targanda uchata raqami bir xil bo‘lsin, nol bundan mustasno.
5	Do‘konda 16 kgli, 17 kgli va 21 kgli pol buyoqlari yashiklarda mavjud. Ularni ochib ko‘rmasdan 185 kg pol buyoq tanlahg. Barcha variantlarni ko‘rib chiqing.
6	$42 \cdot 4^*$ yulduzchalar o‘rniga shunday raqamlarni tanlang, hosil bo‘lgan besh xonali son 72ga bo‘lsin.
7	O‘zi va raqamlari yig‘indisi 7ga karrali uch xonali sonlarni toping.
8	Shunday to‘rt xonali sonlarni toping-ki, 133ga bo‘lganda qoldiq 125ga va 134ga bo‘lganda qoldiq 111ga teng bo‘lsin.
9	Raqamlari yig‘indisi berilgan butun songa teng to‘rt xonali barcha sonlarni toping.
10	O‘zi va raqamlari yig‘indisi 11ga karrali to‘rt xonali sonlarni toping.
11	$42 \cdot 4^*$ yulduzchalar o‘rniga shunday raqamlarni tanlang, hosil bo‘lgan olti xonali son 11ga bo‘lsin.
12	Berilgan natural sonni bo‘luvchilari yig‘indisini toping.
13	Berilgan N sonidan oshmaydigan barcha pifagor natural sonlarni toping. (Agar berilgan uch xonali sonning dastlabki ikkita raqami yig‘indisining kvadrati uchinchi raqamiga teng bo‘lsa, bunday natural son pifagor soni deb ataladi)
14	Birinchi ikkita raqamlari yig‘indisi oxirgi ikkita raqamlari yig‘indisiga teng bo‘lgan barcha to‘rt xonali sonlarni toping.

6-JADVAL

Variant tartib raqami	Shart
1	$s = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$
2	$s = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots$
3	$s = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

4	$s = 1 - \frac{1}{1} + \frac{1}{2} - \frac{1}{3} + \dots$
5	$s = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots$
6	$s = \frac{1}{1*3} + \frac{1}{2*4} + \frac{1}{3*5} + \dots$
7	$s = 1 - \frac{1}{1} + \frac{1}{3} - \frac{1}{5} + \dots$
8	$s = \frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} + \dots$
9	$s = \frac{1}{1*3} + \frac{1}{3*5} + \frac{1}{5*7} + \dots$
10	$s = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots$
11	$s = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$
12	$s = \frac{1}{2*3} + \frac{2}{3*4} + \frac{3}{4*5} + \dots$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

7- LABORATORIYA ISHI

MAVZU: PASKALDA MASSIVLAR BILAN ISHLASH.

Maqsad: Pascalda massivlar va to‘plamlar bilan ishlash, ulardan foydalanish ko‘nikmalarini shakllantirish.

Vositalar: Pascal dasturlash tili o‘rnatilgan kompyuterlar.

NAZARIY QISM.

Massiv o‘zida bir xil turdagi qiymatlarni saqlovchi kattalik xisoblanadi. Massiv bilan ishlash uchun quyidagilarni amalga oshirish kerak:

Massiv o‘zgaruvchilar yoki ozgarmaslar bo‘limida e‘lon qilinishi mumkin:

Massivni kiritish, chiqarish, elementlari bilan ishlash uchun qaytarilish operatorlaridan foydalanish kerak bo‘ladi. (while, repeat. for).

Agar dasturda massiv ishtirok etsa uni quyidagicha e‘lon qilinadi:

var

**massiv_nomi:array[boshlang‘ich indeks..oxirgi indeks] of
[qiymatlar turi];**

Bu yerda massiv_nomi uchun pascal tili qonuniyati bo‘yicha ixtiyoriy identifikator tanlanadi, Array-massivni e‘lon qiluvchi so‘z, Boshlang‘ich va oxirgi indekslar – butun sonlar bo‘lishi shart. Misol uchun yanvar oyidagi temperatura qiymatlarini o‘zida saqlovchi massivni e‘lon qilish uchun:

var

Yanvar:array[1..31] of real;

Bu holda biz bo‘sh massiv e‘lon qildik. Masalani hal qilish davomida bu massini to‘ldirish, qiymatlari ustida amallar bajarish kerak bo‘ladi, buning uchun massivning har bir elementiga murojaat qilish kerak bo‘ladi. Quyida massivni to‘ldirish ko‘rsatilgan:

For i:=1 to 31 do

Begin

Write(i,‘-elementni kiriting:’);

Readln(a[i]);

End;

Massivlar umumiy holda quyidagicha tavsiflanadi:

ARRAY[<indeks toifasi>] OF ARRAY [<indeks toifasi>] OF <elementlar toifasi>;

yoki

ARRAY[<indeks toifasi, indeks toifasi>] OF <elementlar toifasi>;

Masalan, Bir o‘lchamli massiv:

Type mas1=array[1..100] of real;

Var a1:mas;

yoki

Var a1:array[1..100] of real;

ko‘rinishda, ko‘p o‘lchamli massiv:

Type mas2=array[1..10,1..10] of real;

Var a2:mas;

yoki

Var a2:array[1..10,1..10] of real;

ko‘rinishda tavsiflanadi.

Dastur ichida a1[1]:=5; a2[2,3]:=10; shaklda foydalanish mumkin.

Nazariya bo‘yicha qisqacha ma‘lumot

1. Yozuvlar haqida umumiy ma‘lumotlar

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

E’tiboringizga yana bir yangi, boshqa algoritmik tillarda mavjud bo‘lgan, Paskal tilining hosilaviy turlaridan birini - kombinatsiyali turni havola qilamiz. Kombinatsiyali turning qiymati ham xuddi massivlarniki kabi (massivlar haqida boshlang‘ich ma’lumotlarga egasiz deb o‘ylaymiz) bir nechta haddan tashkil topadi, lekin massivdan farqli o‘laroq, uning har bir hadi turlicha turli bo‘lishi mumkin. Bu turning hadlariga, ularning joylashgan tartib raqamlari bilan emas, balki ismlari orqali murojaat qilinadi.

1-Misol. x va u mavhum sonlari ustida qo‘shish, ayirish va ko‘paytirish amallarini bajarish dasturini tuzing.

Masalani yechish algoritmi:

Agar $x = Re\ x + i\ Im\ x$, $y = Re\ y + i\ Im\ y$ bo‘lsa, ular ustida sanab o‘tilgan amallarni bajarish algoritmi quyidagicha bo‘ladi:

$$\begin{aligned}u &= x + y, & Re\ u &= Re\ x + Re\ y, & Im\ u &= Im\ x + Im\ y; \\v &= x - y, & Re\ v &= Re\ x - Re\ y, & Im\ v &= Im\ x - Im\ y; \\w &= x * y, & Re\ w &= Re\ x * Re\ y - Im\ x * Im\ y, \\ & & Im\ w &= Re\ y * Im\ x + Re\ x * Im\ y\end{aligned}$$

Endi mazkur algoritmnı dasturda ifoda etamiz:

Program L1;

type

comp = record

re, im: real

end;

var

x, y, u, v, w: comp;

begin { x va u mavhum sonlarning haqiqiy ($Re\ x$, $Re\ y$) va mavhum ($Im\ x$, $Im\ y$) qismlarini kiritish}

readln (*x.re, x.im, y.re, y.im*);

{ $u=x+y$ }

u.re := x.re + y.re; u.im := x.im + y.im;

{ $v=x-y$ }

v.re := x.re - y.re; v.im := x.im - y.im;

{ $w=x*y$ }

*w.re := x.re * y.re - x.im * y.im;*

*w.im := x.re * y.im + x.im * y.re;*

writeln (' $x + y =$ ', *u.re*, ' + ', *u.im*, '*i');

writeln (' $x - y =$ ', *v.re*, ' + ', *v.im*, '*i');

writeln (' $x * y =$ ', *w.re*, ' + ', *w.im*, '*i');

end.

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

1-misol. Matritsalarini qo‘shish.

Program L2;

const n = 3; m = 4;

*{ n - matritsa satrlari soni,
m - ustunlar soni }*

var i, j: integer;

A, B, C: array [1..n, 1..m] of real;

begin { A, V matritsa hadlarini kiritish}

for i := 1 to n do

for j := 1 to m do

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
    readln (A[i,j], B[i,j]);
  for i := 1 to n do
  for j := 1 to m do
    begin
      C[i,j] := A[i,j] + B[i,j];
      writeln (C[i,j])
    end
  end.
```

2-misol. Matritsani vektorga ko‘paytirish.

```
Program L3;
const n = 3; m = 4;
type matr = array [1..n, 1..m] of real;
      vect = array [1..m] of real;
var   i, j: byte;
      A: matr;
      B, C: vect;
begin
  writeln ('A matritsa hadlarini kiriting');
  for i:=1 to n do
    for j:=1 to m do
      readln (A[i,j]);
  writeln ('B vektor hadlarini kiriting');
  for i:=1 to n do readln (B[i]);
  for i:=1 to n do
    begin
      C[i]:=0;
      for j:=1 to m do
        C[i]:= C[i] + A[i,j] * B[j];
      writeln (C[i]);
    end
  end.
```

3-misol. Matritsa hadlarining eng kattasini topish va uning joylashgan joyini aniqlash.

```
Program L4;
const n=3; m=4;
var   A: array [1..n, 1..m] of real;
      R: real;
      i, j: byte; K, L: byte;
begin {A matritsa hadlarini kiritish}
  for i:=1 to n do
    for j:=1 to m do
      readln (A[i,j]);
  R:= A[1,1]; L:= 1; K:=1;
  for i:=1 to n do
    for j:=1 to m do
      begin
        if R < A[i,j] then
          begin
            R:=A[i,j];
            L:= i; K:= j;
          end;
      end;
  writeln ('max A=', R);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
writeln ('satr=' ,L,'ustun =' ,K);  
end.
```

4-misol.

Foydalanuvchi tomonidan kiritilgan son massivda necha marta qatnashishini aniqash dasturi tuzilsin. (Massivda biror sonni qidirish).

```
Program m_1;  
Const N=7;  
var  
a : array[1..n] of integer;  
s, q, i : integer;  
begin  
writeln(n,' ta butun sonni kiriting: ');  
for i:=1 to n do  
begin  
write(I,'-son:');  
readln(a[i]);  
end;  
write('Qidirilayotgn sonni kiriting: ');  
readln(q);  
s:=0;  
for i:=1 to n do  
if a[i]=q then s:=s+1;  
if s<>0 then writeln(q,' soni ', s,' marta qatnashdi' )  
else writeln('Bunday element massivda mavjud emas');  
readln;  
end.
```

7-LABORATORIYA ISHI UCHUN TOPSHIRIQLAR:

1. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa ustunlarida turuvchi elementlar ichidan eng kattalari yig'indisi topilsin.

2. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa satrida turuvchi elementlar ichidan eng kichiklari ko'paytmasini toping.

3. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa ustunlarida turuvchi elementlar ichidan eng kichiklarining o'rtacha arifmetigi topilsin.

4. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa satrlarida turuvchi elementlar ichidan eng kattalarining o'rtacha geometrigi topilsin.

5. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa elementlarining o'rtacha qiymati topilsin. Massivda o'rtacha qiymatdan kichik va katta elementlari son topilsin.

6. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa ustunlarida 4-ga karrali element bor yo'qligini aniqlansin. Agar bor bo'lsa ustunga mos $C(n)$ massiv elementiga 1 qiymat aks holda 0 qiymat berilsin, yani $C(n)$ massiv elementlari qiymati topilsin.

7. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa ustunlari 2-ga karralilarining yig'indisini mos $B(n)$ massiv elementiga joylashtirilsin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

8. Haqiqiy elementlardan tashkil topgan kvadrat matrisa $A = \{a_{ij}\}$ berilgan. Nol va birlardan tashkil topgan b_1, b_2, \dots, b_n ketma-ketlik tuzilsin, unda $b_i = 1$ agar matrisa satrlarida hech bo‘lmaganda 1 bilan 10 orasida yotuvchi bitta element mavjud bo‘lsa.

9. Haqiqiy elementlardan tashkil topgan kvadrat matrisa berilgan. Nol va birlardan tashkil topgan b_1, b_2, \dots, b_n ketma-ketlik tuzilsin, unda $b_i = 1$ agar matrisa ustunlarida hech bo‘lmaganda bitta 2-ga karrali element mavjud bo‘lsa.

10. Ikkita $A(n)$ va $B(m)$ massivlar berilgan. Bu massivlar elementlari ichida nechtdan manfiy va musbat elementlar borligi topilsin.

11. $A = \{a_{ij}\}$ matrisa asosiy diagonidagi elementlar o‘rta arifmetigi topilsin. Bu yerda $a_{ij} = (5^i - 3^i) / n; ij = 1, 2, \dots, n$.

12. $A = \{a_{ij}\}$ matrisa ustun elementlari orasida 4-ga karralilari soni topilsin. Bu yerda $a_{ij} = 4^i$ agar $i = j$ bo‘lsa, $a_{ij} = 2^{i+j}$ agar $i \neq j$ bo‘lsa; $ij = 1, 2, \dots, n$.

13. $A = \{a_{ij}\}$ matrisa berilgan. Matrisaning juft nomerlari ustun elementlari orasida 2 va 4-ga karralilari soni topilsin.

14. $A = \{a_{ij}\}$ matrisa satr elementlari orasida 2-ga karralilari soni topilsin. Bu yerda $a_{ij} = 2^{i+j}$ agar $i+j > 3$ bo‘lsa, $a_{ij} = 5$ agar $i+j \leq 3$ bo‘lsa; $i = 1, 2, \dots, m; j = 1, 2, \dots, n$.

15. $A = \{a_{ij}\}$ matrisa berilgan. Matrisaning toq nomerli satr elementlari orasida 3-ga karralilarining o‘rta arifmetigi topilsin.

16. X son va $A = \{a_{ij}\}$ matrisa berilgan. Matrisa elementlari ichida X sonidan kattalari soni topilsin va shu elementlar kvadratga oshirilsin.

17. X son $A = \{a_{ij}\}$ matrisa berilgan. Matrisa elementlari ichida X sonidan kattalari soni topilsin va shu elementlar ichida X^2 dan katta element mavjudligi aniqlansin.

18. $A = \{a_{ij}\}$ matrisa berilgan bo‘lsa quyidagilarni hisoblang:

$$x_i = \max\{a_{ij}\}; y_j = \min\{a_{ij}\};$$

19. $A = \{a_{ij}\}$ matrisa berilgan bo‘lsa quyidagilarni hisoblang:

$$x_i = \max\{a_{ij}\}; y = \min\{x_i\}; z = \max\{x_i\}.$$

20. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa bosh diagonali va 1-ustun elementlari o‘rta arifmetigi topilsin va ular orasidan kattasi aniqlansin.

21. $A = \{a_{ij}\}$ matrisa berilgan. Matrisa bosh diagonali va n -chi satr elementlar yig‘indisi topilsin va ular orasidan kattasi aniqlansin.

22. $A = \{a_{ij}\}$ va $B = \{b_{ij}\}$ matrisalari berilgan. Matrisa elementlari o‘rta arifmetigi hisoblansin va ular kattasi aniqlansin.

23. $A = \{a_{ij}\}$ va $B = \{b_{ij}\}$ matrisalari berilgan. Bu matrisa elementlari eng kichigini topib, ulardan kattasi aniqlansin.

24. Elementlari butun $A = \{a_{ij}\}$ matrisa berilgan. Matrisa elementlarining 2-ga karralilari va ularning sonlari aniqlansin.

25. Elementlari butun $X(n)$ vektori berilgan. Vektorning juft qiymatli elementlarini oldin, toq qiymatli elementlarini keyin $Y(n)$ vektoriga joylashtirilsin.

26. Elementlari butun $X(n)$ vektori berilgan. Vektorning manfiy qiymatli elementlarini oldin, musbat qiymatli elementlarini keyin $Y(n)$ vektoriga joylashtirilsin.

27. $X(n)$ vektori berilgan. Vektorning elementlari $Y(n)$ vektoriga teskari joylashtirilsin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

28. $X(n)$ vektori va z soni berilgan. Vektorning elementlarini $Y(n)$ vektoriga oldin z sonidan kichiklari va keyin kattalari joylashtirilsin.

29. Butun sonli $X(n)$ vektori berilgan. Vektorning eng katta elementining eng kichik umumiy bo‘luvchisi topilsin.

30. Butun sonli $X(n)$ vektori berilgan. Vektorning eng katta elementidan bitta oldingi eng katta elementini topib, uning eng katta umumiy bo‘luvchisi topilsin.

Berilgan $A(N)$ massiv elementlarini $B(N)$ massivga teskari tartib bilan yozing.

2.1. $A(N)$ massiv berilgan. $B(N)$ massivni quyidagi formula yordamida hosil qiling:
 $b_i = (a_1 + a_2 + \dots + a_i) / i$.

2.2. Quyidagini hisoblang:

$$P = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + 3 \cdot 4 \cdot 5 \cdot 6 + \dots + N \cdot (N+1) \cdot \dots \cdot 2N.$$

2.3. Berilgan $X(N)$ massivning maksimal komponentdan (agar ular bir nechta bo‘lsa, maksimal komponentni tartibi bo‘yicha birinchisini oling) oldingi barcha manfiy komponentlarni nol bilan almashtiring.

2.4. Berilgan $X(N)$ massivning beshga karrali bo‘lmagan barcha elementlari kvadratlari yig‘indisini hisoblang.

2.5. Berilgan $X(N)$ massivda eng kichik va eng katta elementlar o‘rnini almashtiring.

2.6. Berilgan $X(N)$ massivdagi berilgan qiymatdan kichiklarning sonini aniqlang.

2.7. Berilgan $A(N)$ massivning komponentlarini chapga bir pozisiya siklik siljishni amalga oshiring, ya’ni $A = (a_2, a_3, \dots, a_N, a_1)$ vektorni hosil qiling.

2.8. Berilgan $A(N)$ massivning komponentlarini o‘ngga ikki pozisiya siklik siljishni amalga oshiring, ya’ni $A = (a_{N-1}, a_N, a_1, a_2, \dots, a_{N-2})$ vektorni hosil qiling.

Berilgan $A(N, M)$ matritsada eng katta elementni va u joylashgan satr hamda ustun nomerini toping.

3.1. Berilgan $A(N, M)$ matritsada har bir satr elementlarining yig‘indisini, musbatlari o‘rta arifmetigi va sonini hisoblang.

3.2. Berilgan $A(N, M)$ butun sonli matritsaning elementlari yig‘indisi juft son bo‘ladimi, yo‘kmi aniqlang.

3.3. Berilgan $A(N, M)$ matritsada barcha elementlarining o‘rta arifmetigidan katta bo‘lgan elementlar sonini aniqlang.

3.4. Berilgan $A(N, M)$ butun sonli matritsaning toq qiymatli elementlarining yig‘indisi va ko‘paytmasini hisoblang.

3.5. $A(N, M)$ matritsa berilgan. $X(M)$ vektorni hisoblang, bu erda X_j qiymati A matritsaning j -chi usundagi musbat elementlar yig‘indisi.

3.6. $A(N, M)$ matritsa berilgan. $X(M)$ vektorni hosil qiling, u matritsaning P -satriga teng va $Y(N)$ vektorni hosil qiling, u matritsaning Q -ustuniga teng.

3.7. Berilgan $A(N, M)$ matritsada eng katta va eng kichik elementlari o‘rnini almashtiring.

3.8. Berilgan n natural songa ko‘ra (n, n) tartibli ushbu shakl-dagi matritsalarini hosil qiling.

LABORATORIYA ISHI №8

MAVZU. PASKALDA SATRIY KATTALIKLAR BILAN ISHLASH.

Maqsad: Paskalda satriy kattaliklar bilan tanishish, ulardan foydalanish ko‘nikmalarini shakllantirish.

Vositalar: Pascal dasturlash tili o‘rnatilgan kompyuterlar.

Mashg‘ulotning borishi.

3. Nazariy material bilan tanishish.

Satrlar ASCII kodining belgilarining apostroflar (` `) ichiga olingan ketma-ketligidan iborat. Masalan `mart`, `123` va hokazo. Satr uzunligi deb unda qatnashayotgan belgilar soniga aytiladi. Bunda chegaralovchi apostrof belgilar hisobga olinmaydi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Masalan, ‘ ’ satr uzunligi 1ga teng, ‘ ’ uzunligi 0ga teng. Satriy o‘zgaruvchilar quyidagicha e‘lon qilinadi:

```
var
    O`zgaruvchi_Nomi:string;
yoki
var
    O`zgaruvchi_Nomi:string[m];
```

Bu yerda M-ushbu satrga ajratilgan bayt hisobi. Satrning har bir belgisi xotirada 1 bayt joy egallaydi. Agar satr uzunligi ko‘rsatilmagan bo‘lsa, ushbu satr 255 belgidan iborat bo‘lishi mumkin.

“Pascal” tilida matnli ma‘lumotlar bilan ishlash uchun bi satr funksiya va proseduralar ishlab chiqilgan:

```
Copy(S:string;L:integer):string;
```

Bu yerda

S satrning start belgisidan boshlab L uzunlikdagi satrni belgilaydi(aniqlaydi).

4. Tayyor tuzilgan dasturlarning tahlili.

1- Topshiriq:

1-Misol.

```
Program m-1;
var s, s1:string;
begin
    S:= ‘algoritm’;
    S1:=copy(s,5,4); {s1=ritm}
    Writeln(s1);
    Readln;
End.
```

Natijada ekranda ritm so‘zi chop etladi.

```
Delete (var s:string; start, len:integer);
```

S satrning start simvolidan boshlab len uzunlikdagi satrini o‘chiradi.

```
S:= ‘algoritm’;
```

```
Delete(s,1,4);
```

Natija Ritm

```
Delete(s,3,1); ritm satrini aniqlaydi.
```

```
Insert(subs:string; var s:string;start:integer);
```

S satrga start o‘rindan boshlab subs satrini qo‘shadi.

```
S:= ‘alm’;
```

```
Subs:= ‘gorit’;
```

```
Insert(subs,s,3);
```

```
Pos(subs,s:string);byte;
```

Subs satri s satriga tegishli ekanligini aniqlaydi.

2-Misol.

```
S:= ‘algoritm’;
```

```
Subs:= ‘ritm’;
```

```
X:=pos(subs,s);
```

```
If x<>0 then writeln(subs, ‘ satri ’,s, ‘ satrining ’,x, ‘ o‘rnidan boshlanadi’’);
```

4-Misol.

Berilgan so‘zni Polindrom yoki Polindrom emasligini tekshiruvchi dasturni tuzing.

```
Program m_1;
```

```
uses crt;
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

```
var s,t:string;
    k,i:integer;
begin
clrscr;
write('Tekshirilayotgan so`zni kiriting:');
readln(s);
k:=length(s);
t:='';
for i:=1 to k do
t:=copy(s,i,1)+t;
if s=t then writeln('Palindrom`) else writeln('Palindrom emas`);
readln;
end.
```

Ushbu masalada berilgan so`zni polindromligini topish uchun shu so`zni har bir harfini aniqlash uchun qaytarilish jarayoni ichida copy(s,i,1) funksiyasidan foydalaniladi va uni teskarida yig`ib borish uchun copy(s,i,1)+t amali bajariladi.

Buni biz katak so`zi misolida ko`rib chiqamiz:

Qaytarilish jarayoni bajarilganda copy(s,i,1)+t buyrug`i

k

ak

tak

atak

katak

so`zni hosil qiladi va if s=t then writeln('Palindrom`) buyrug`i bajarilgandan so`ng ekranda Palindrom javobi paydo bo`ladi.

Mustaqil bajarish uchun laboratoriya ishlari:

1. Berilgan satriy kattalikdagi mavjud barcha kichik shriftlarni kattasiga almashtirish dasturini tuzing.
2. Berilgan satriy kattalikda mavjud barcha kichik shriftlarni kattalriga almashtirish dasturini tuzing.
3. Berilgan satriy kattalikning uzunligini aniqlovchi va uning har bir simvoli kodini aniqlovchi dastur tuzing.
4. Ixtiyoriy 2 ta va 3 ta satriy kattaliklarni qo`shib chop etuvchi dastur tuzing.
5. Biror satriy kattalikdan uning ma`lum bir qismni nusxa olish dasturini tuzing.
6. Biror satriy kattalikdan uning ma`lum bir qismini o`chirib yangi satriy kattalikni qo`shish dastrini tuzing.
7. Ixtiyoriy satriy kattalikning har bir belgisini kodlar jadvalida undan keying ikki belgilar bilan almashtiruvchi dastur tuzing.
8. Bior son berilgan. Barcha 2, 3, 4 xonali sonlar ichidan raqamlarning kvadratlari yig`indisi su songa teng bo`lganlarini chop etuvchi dastur tuzing.
9. Berilgan natural sonda juft o`rinda keluvchi raqamlar yig`indisini hisoblash dasturini tuzing.
10. Ixtiyoriy satri kattalik tarkibidagi 'a' harfini ishtirok etish sonini aniqlovchi dastur tuzing.
11. Ixtiyoriy kiritilgan s satriy kattalikni teskarilovchi dastur tuzing.
12. Berilgan satriy kattalik tarkibidagi 'A' belgisini 'B' belgiga almashtiruvchi dastur tuzing.
13. Ixtiyoriy berilgan jumla tarkibidagi so`zlar sonini aniqlovchi dastur tuzing.
14. Ixtiyoriy ko`p xonali son tarkibiga 6 raqaming ishtirok etish sonini aniqlovchi dastur tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

15. Etti xonali son takibida eng kichik raqamning ishtirok etis sonini aniqlovchi dastur tuzing.
16. Ixtiyoriy jumla kiritish natijasida undagi har bir belgisi sonini aniqlovchi dastur tuzing.
17. Satriy kattalik bir necha soʻzdan iborat. Undagi har bir soʻzning bosh harfini berilgan simvolga oʻzgartirish dasturini tuzing.
18. Kiritilgan har bir soʻz probellar bilan ajratilgan. Satrdagi birinchi k ta soʻzni oʻchirish dasturini tuzing.
19. Matndagi soʻzlarning birinchi harfi bilan oxirgi harfini oʻrinlarini almashtirish dasturini tuzing.
20. Matndagi lotin alfavitida yozilgan soʻzlarning harflarini ularga mos keluvchi kirill alfaviti harflari bilan almashtirish dasturini tuzing.
21. Kirill alfavitida yozilgan matnni har bir soʻzidagi harflarni unga mos keluvchi lotin alfaviti harflariga oʻzgartiruvchi dasturini tuzing.
22. Berilgan satriy kattalikdagi matnnining barcha 'dan' qoʻshimchalarini қўшимчаларини 'da' qoʻshimchasiga oʻzgartirish dasturini tuzing.
23. Berilgan satriy kattalikdagi matnnining barcha soʻzlarining uzunligini hisoblovchi dasturini tuzing.
24. Berilgan satriy kattalikdagi soʻzlarning oʻrnini teskari tartibda joylashtiring.

9 – LABORATORIYA MASHG‘ULOTI

MAVZU: QISM DASTURLAR-FUNKSIYALAR VA PROTSEDURALARGA DOIR DASTURLAR TUZISH.

Ishning maqsadi: MAQSAD: Maʼruzada olingan bilimlarni mustaxkamlash, dasturlash koʻnikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qism dasturlar bilan ishlash koʻnikmasini shakllantirish.

TOPSHIRIQLAR: Berilgan variantlarga koʻra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY QISM

Dasturlash jarayonida murakkab dasturlarning bir necha joyida bir xil vazifani bajaruvchi operatorlar guruhini qoʻllashga toʻgʻri keladi va dasturda bir-biriga aynan oʻxshash bir necha qism dasturlar vujudga keladi. Komputer otirasini va dasturtuzuvvchi vaqtini tejash maqsadida ushbu qismdagi oʻxshash peratorlardan bir martta asosiy dastur dasturdan ajratib yoziladi va unga asosiy dastur bajarilishi jarayonida murojaat qilinadi. Dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatilishi mumkin boʻlan buday operatorlar guruhida qism dastur deb ataladi va u asosiy dastur bilan birr butunni tashkil qiladi.

Qism dasturlarni ishlatish dasturning hajmini kichraytiradi va uning koʻrinishini, oʻqilishini hamda xatolar sonini kamaytiradi. Protsedura quyidagicha eʼlon qilinadi:

Procedure<nom>(<sohta parametrlar roʻyhati>); Bu yerda PROCEDURE-xizmatchi soʻz; <nom>-protseduraning nomi; <soxta paramtrlar roʻyxati>-

turlari koʻrsatilgan erilganlarni va natijalarni ifodalaydigan nomlar turlari sanab oʻtiladigan satr.

Protsedura - bu dasturning nomlangan qismi hisoblanib, uni nomi bilan chaqiriladi va ular ustida ishlash yoki maʼlum operatsiyalar bajarish mumkin. Protsedura strukturasi dastur strukturasi oʻxshash boʻladi. Protsedura ifodada operant holatda uchrashi mumkin emas.

Pascal tilining funksiya va protseduralari ikkita guruhga boʻlinadi: foydalanuvchi tomonidan aniqlangan va ilgari mavjud boʻlgan protsedura va funksiyalar. Birinchisi foydalanuvchi tomonidan nomlanadi va chaqiriladi. Ikkinchisi tilning tarkibiga kirib, uni

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

faqat quyilgan nomi bo‘yicha chaqiriladi.

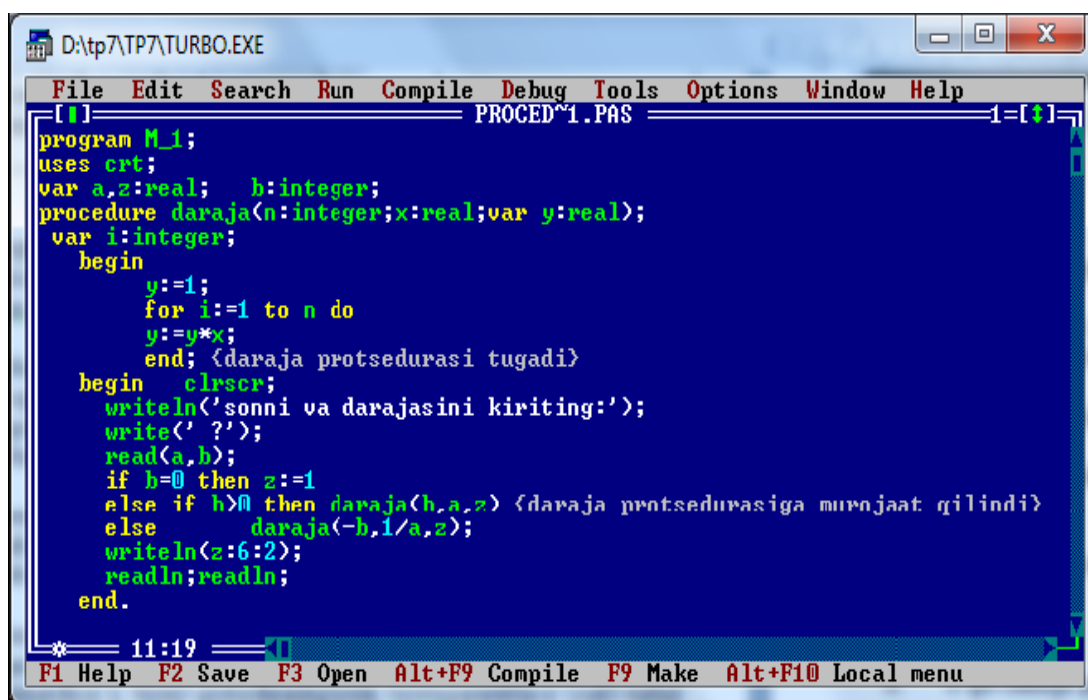
Hamma standart vositalar maxsus biblioteka modullarida saqlanadi va ular sistema nomlariga ega bo‘ladi. Protsedura protsedura nomi va protsedura tanasidan iborat bo‘ladi. Protsedura protsedura sarlavhasi (nomi) va tanasidan iborat. *Protseduraning sarlavhasi* PROCEDURE rezervlashgan so‘zdan boshlanib, protseduraga nom berilib, qavs ichida har bir parametrlarning tiplari ko‘rsatilgan formal parametrlar ro‘yxati keltiriladi. Protseduraning nomi - programma chegarasidagi identifikatori. *Protsedura tanasi* strukturasi bo‘yicha dasturga o‘xshagan lokal blokdan tashkil topadi. Belgi, konstanta, tiplar va boshqalar shu protsedura chegarasida haqiqiydir. Protsedura tanasida turli global konstanta va o‘zgaruvchilar qo‘llanilishi mumkin.

```
PROCEDURE <nomi> (Formal parametrlar);  
CONST ...;  
TYPE ...;  
VAR ...;  
BEGIN  
  <operatorlar>  
END;
```

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

1-MISOL.

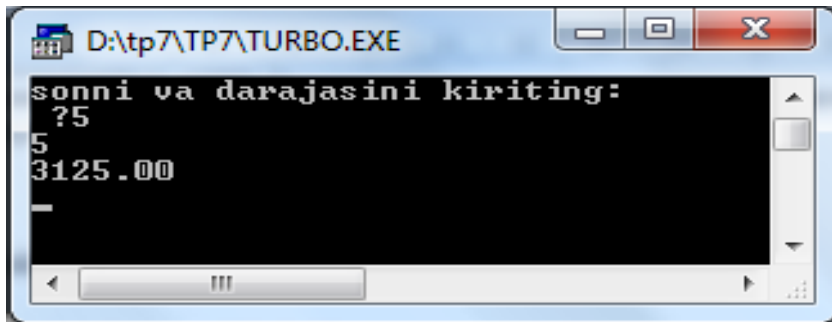
$y = x^n$ ni hisoblash protsedura orqali hal qilinsin.



```
D:\tp7\TP7\TURBO.EXE  
File Edit Search Run Compile Debug Tools Options Window Help  
PROCED~1.PAS  
program M_1;  
uses crt;  
var a,z:real; b:integer;  
procedure daraja(n:integer;x:real;var y:real);  
  var i:integer;  
  begin  
    y:=1;  
    for i:=1 to n do  
      y:=y*x;  
    end; {daraja protsedurasi tugadi}  
  begin clrscr;  
    writeln('sonni va darajasini kiriting:');  
    write(' ? ');  
    read(a,b);  
    if b=0 then z:=1  
    else if b>0 then daraja(h,a,z) {daraja protsedurasiga murojaat qilindi}  
    else daraja(-b,1/a,z);  
    writeln(z:6:2);  
    readln;readln;  
  end.  
* 11:19 *  
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

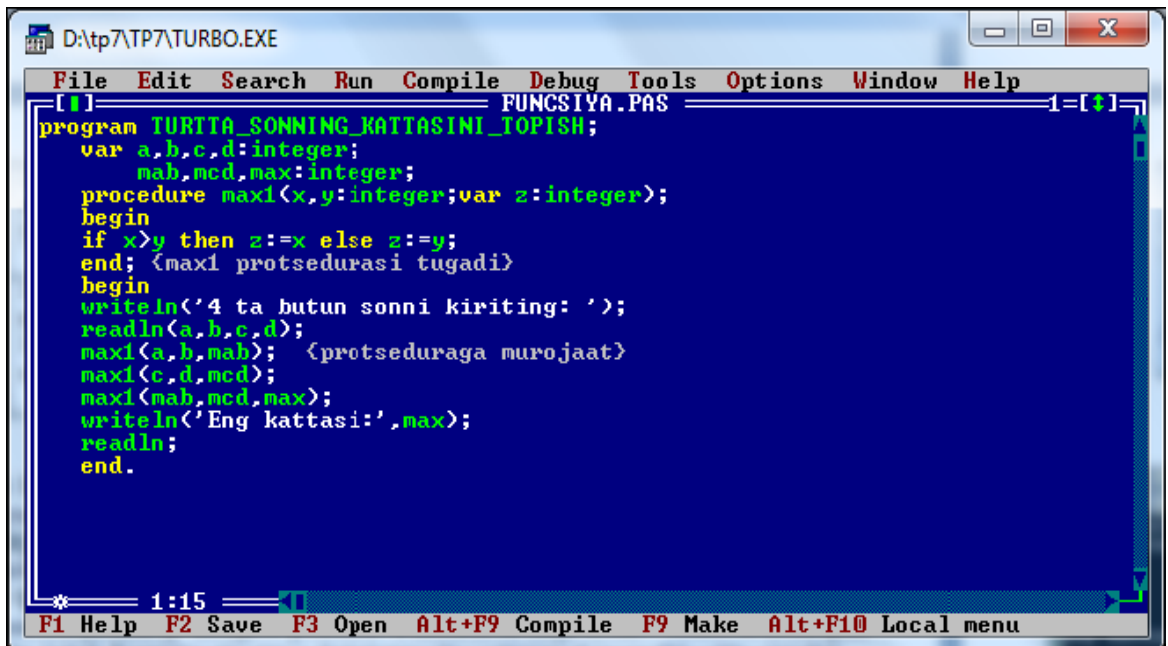
NATIJA:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



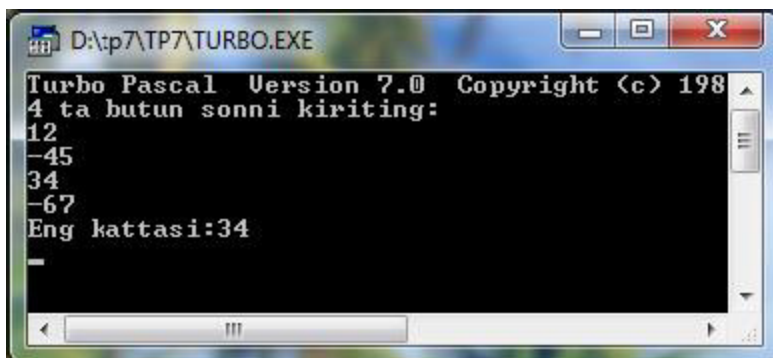
```
D:\tp7\TP7\TURBO.EXE
sonni va darajasini kiriting:
5
3125.00
```

2-MISOL. 4 ta sonning eng kattasini topish protsedura orqali hal qilinsin.



```
D:\tp7\TP7\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
FUNCSIYA.PAS
program TURITA_SONNING_KATTASINI_TOPISH;
var a,b,c,d:integer;
    mab,mcd,max:integer;
procedure max1(x,y:integer;var z:integer);
begin
if x>y then z:=x else z:=y;
end; {max1 protsedurasi tugadi}
begin
writeln('4 ta butun sonni kiriting: ');
readln(a,b,c,d);
max1(a,b,mab); {protseduraga murojaat}
max1(c,d,mcd);
max1(mab,mcd,max);
writeln('Eng kattasi:',max);
readln;
end.
```

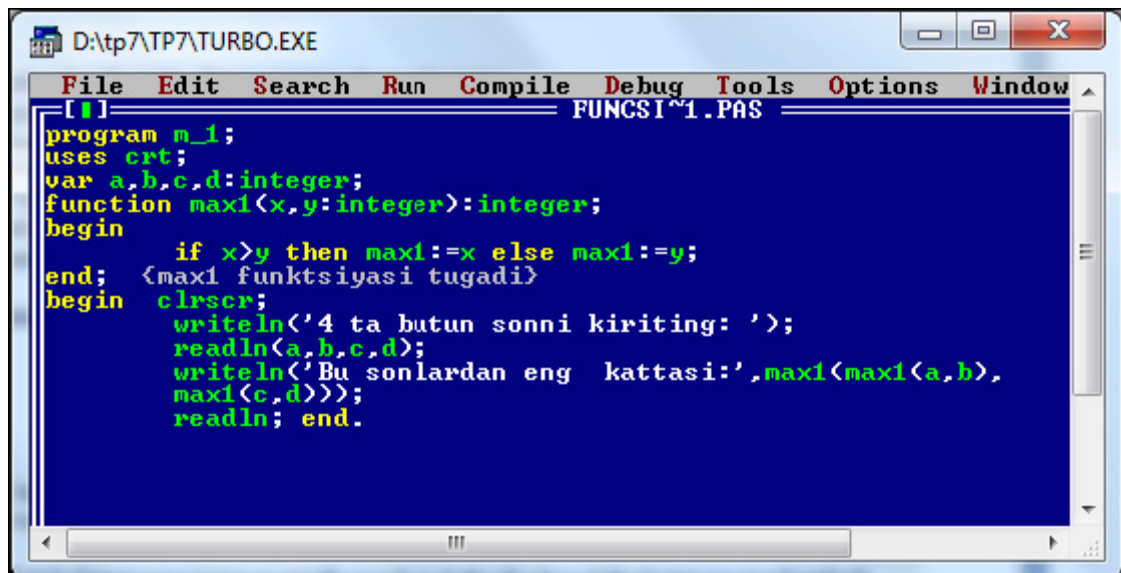
NATIJA:



```
D:\tp7\TP7\TURBO.EXE
Turbo Pascal Version 7.0 Copyright (c) 198
4 ta butun sonni kiriting:
12
-45
34
-67
Eng kattasi:34
```

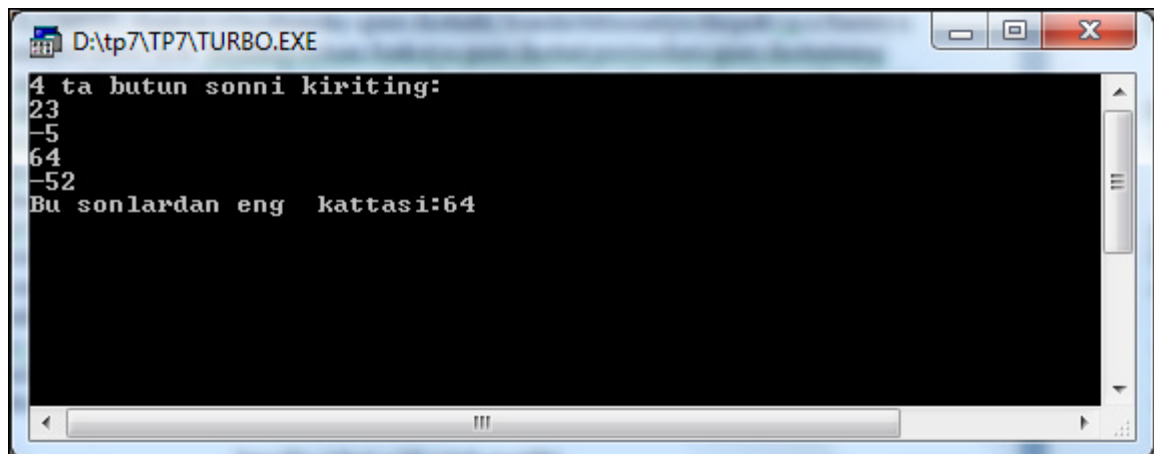
3-MISOL. 4 ta sonning eng kattasini topish funksiya orqali hal qilinsin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



```
File Edit Search Run Compile Debug Tools Options Window
[ ]
D:\tp7\TP7\TURBO.EXE
FUNCSI~1.PAS
program m_1;
uses crt;
var a,b,c,d:integer;
function max1(x,y:integer):integer;
begin
    if x>y then max1:=x else max1:=y;
end; {max1 funktsiyasi tugadi}
begin
    clrscr;
    writeln('4 ta butun sonni kiriting: ');
    readln(a,b,c,d);
    writeln('Bu sonlardan eng kattasi:',max1(max1(a,b),
    max1(c,d)));
    readln; end.
```

NATIJA:



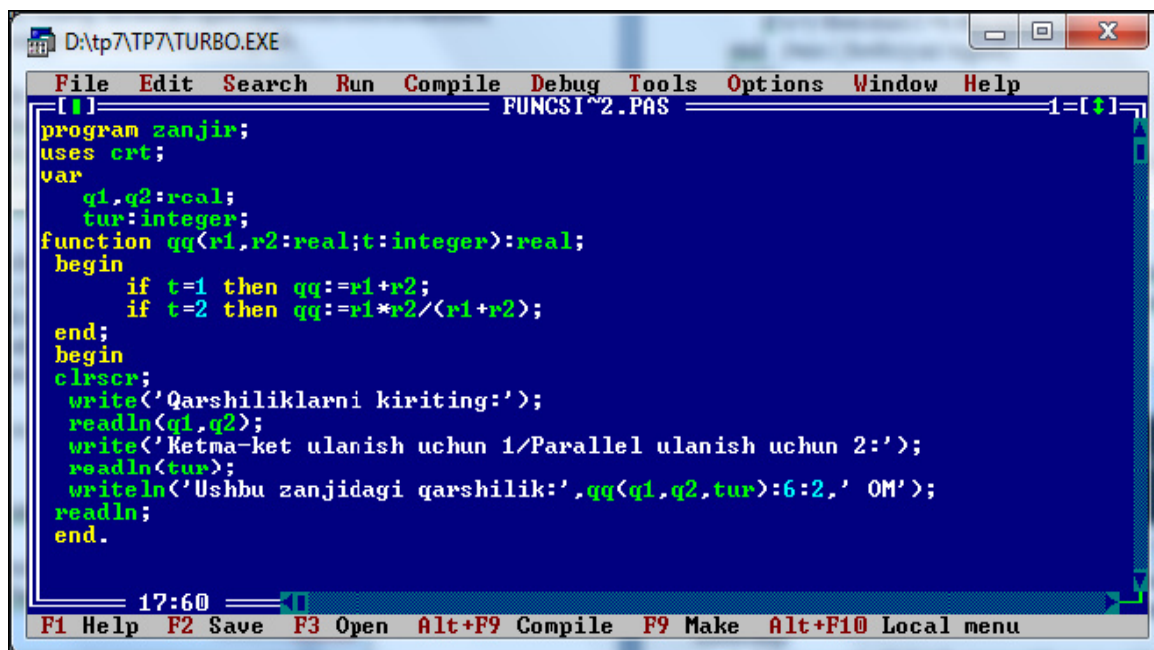
```
D:\tp7\TP7\TURBO.EXE
4 ta butun sonni kiriting:
23
-5
64
-52
Bu sonlardan eng kattasi:64
```

4-MISOL. Ikkita registrdan tashkil topgan zanjirdagi qarshilikni hisoblash dasturi tuzilsin.

Funksiya parametrlari bo‘lib ulanish turi (ketma-ket yoki parallel) turi qabul qilinsin.

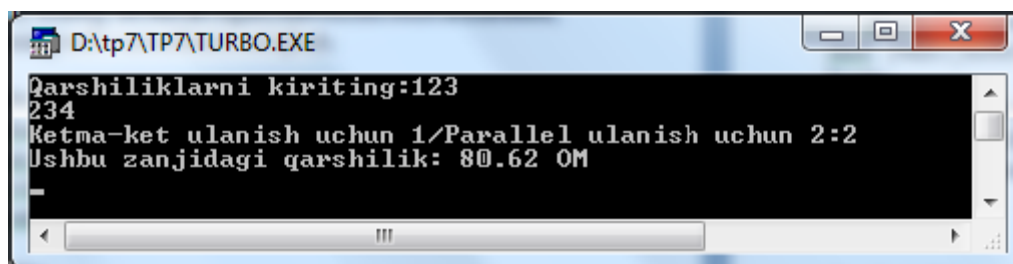
DASTURI:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



```
D:\tp7\TP7\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
[ ] FUNCSI~2.PAS 1=[ ]
program zanjir;
uses crt;
var
  q1,q2:real;
  tur:integer;
function qq(r1,r2:real;t:integer):real;
begin
  if t=1 then qq:=r1+r2;
  if t=2 then qq:=r1*r2/(r1+r2);
end;
begin
  clrscr;
  write('Qarshiliklarni kiriting:');
  readln(q1,q2);
  write('Ketma-ket ulanish uchun 1/Parallel ulanish uchun 2:');
  readln(tur);
  writeln('Ushbu zanjidagi qarshilik:',qq(q1,q2,tur):6:2,' OM');
  readln;
end.
17:60
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

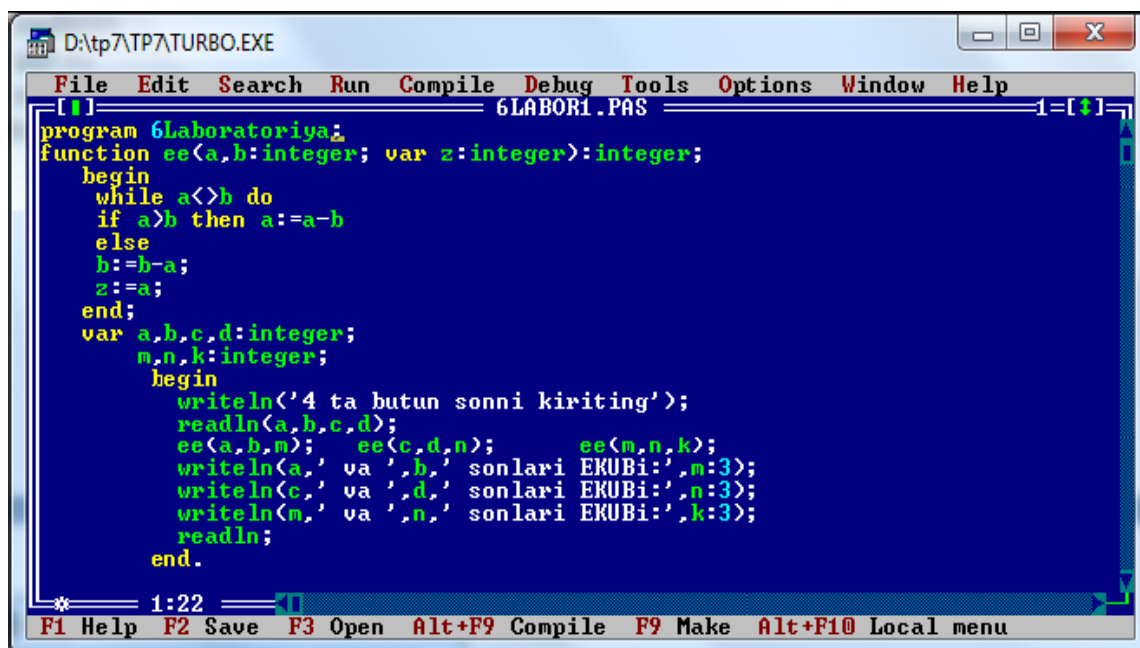
NATIJA:



```
D:\tp7\TP7\TURBO.EXE
Qarshiliklarni kiriting:123
234
Ketma-ket ulanish uchun 1/Parallel ulanish uchun 2:2
Ushbu zanjidagi qarshilik: 80.62 OM
```

5-MISOL. Berilgan 4 ta natural sonning eng kichik umumiy bo‘luvchi EKUB ini toping.

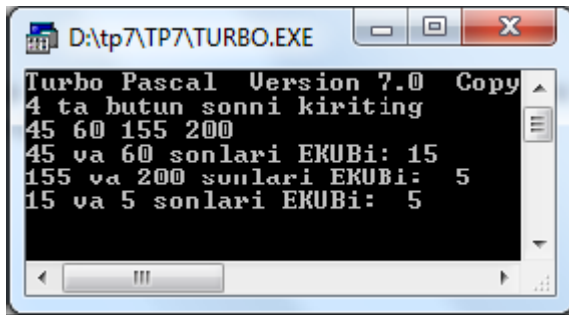
DASTURI:



```
D:\tp7\TP7\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
[ ] 6LABOR1.PAS 1=[ ]
program 6Laboratoriya;
function ee(a,b:integer; var z:integer):integer;
begin
  while a<>b do
    if a>b then a:=a-b
    else
      b:=b-a;
      z:=a;
  end;
var a,b,c,d:integer;
    m,n,k:integer;
begin
  writeln('4 ta butun sonni kiriting');
  readln(a,b,c,d);
  ee(a,b,m); ee(c,d,n); ee(m,n,k);
  writeln(a,' va ',b,' sonlari EKUBi:',m:3);
  writeln(c,' va ',d,' sonlari EKUBi:',n:3);
  writeln(m,' va ',n,' sonlari EKUBi:',k:3);
  readln;
end.
1:22
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

NATIJA:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

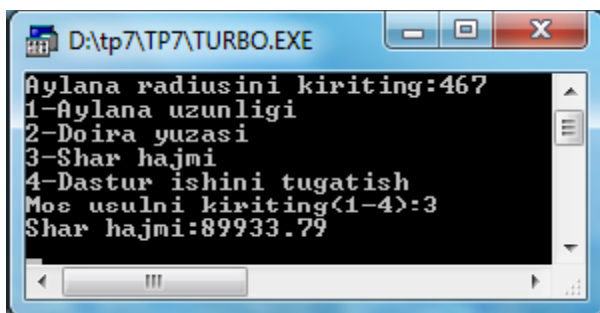


```
Turbo Pascal Version 7.0 Copy
4 ta butun sonni kiriting
45 60 155 200
45 va 60 sonlari EKUBi: 15
155 va 200 sonlari EKUBi: 5
15 va 5 sonlari EKUBi: 5
```

6-MISOL. Quyidagi imkoniyatga ega dastur tuzing. 1-aylana uzunligini, 2-doira yuzasini, 3-shar hajmini, 4-dasturdan chiqishni ta'minlovchi dastur tuzing.

DASTURI:

NATIJA: .



```
Aylana radiusini kiriting:467
1-Aylana uzunligi
2-Doira yuzasi
3-Shar hajmi
4-Dastur ishini tugatish
Mos usulni kiriting(1-4):3
Shar hajmi:89933.79
```

1-MISOL. $A(N)$ massiv elementlarini o‘shish tartibida joylashtiring.

DASTUR

```
program tartiblash;
type x=array [1..20] of real;
```

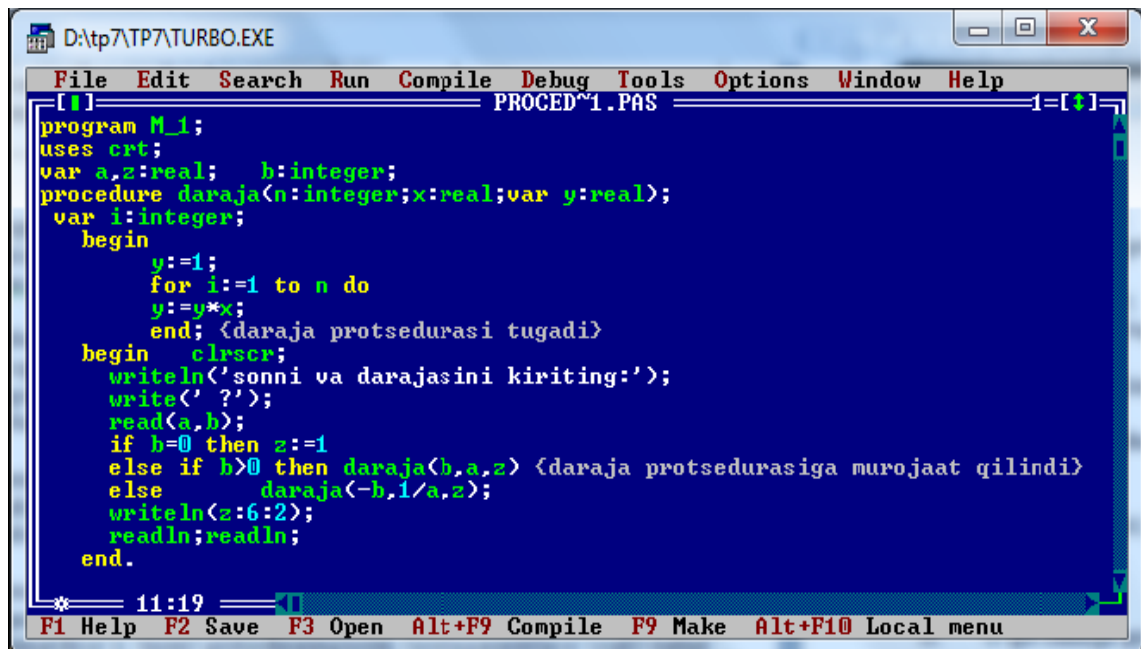
“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

```
var a:x;
    i,j,n,k: integer;
    R: real;
procedure max (q: integer; var nmax: integer);
    var m: real;
    l: integer;
begin
    m:=A[1]; nmax:=1;
    for l:=1 to q do
        if A[l] > m then
            begin
                m:=A[l]; nmax:=l
            end
        end; {prosedurani tasvirlash tugadi}
begin
    writeln ('n=');
    readln (n);
    writeln ('massiv elementlarni kiritish');
    for i:=1 to n do
        readln (A[i]);
    j:=n;
    while j <> 1 do
        begin
            max (j,k);
            R:=A[k]; A[k]:=A[j]; A[j]:=R;
            j:=j-1;
        end;
    writeln ('massiv elementlarni tartiblash');
    for i:=1 to n do
        writeln (A[i]);
end.
```

2-misol.

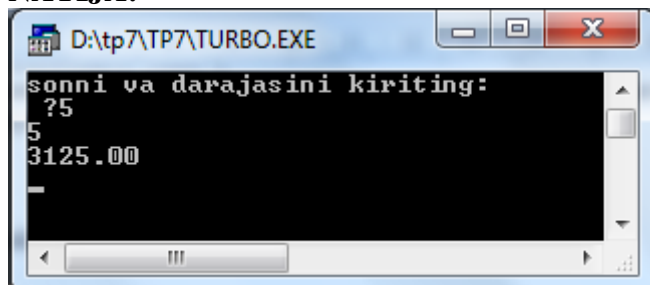
$y = x^n$ ni hisoblash protsedura orqali hal qilinsin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



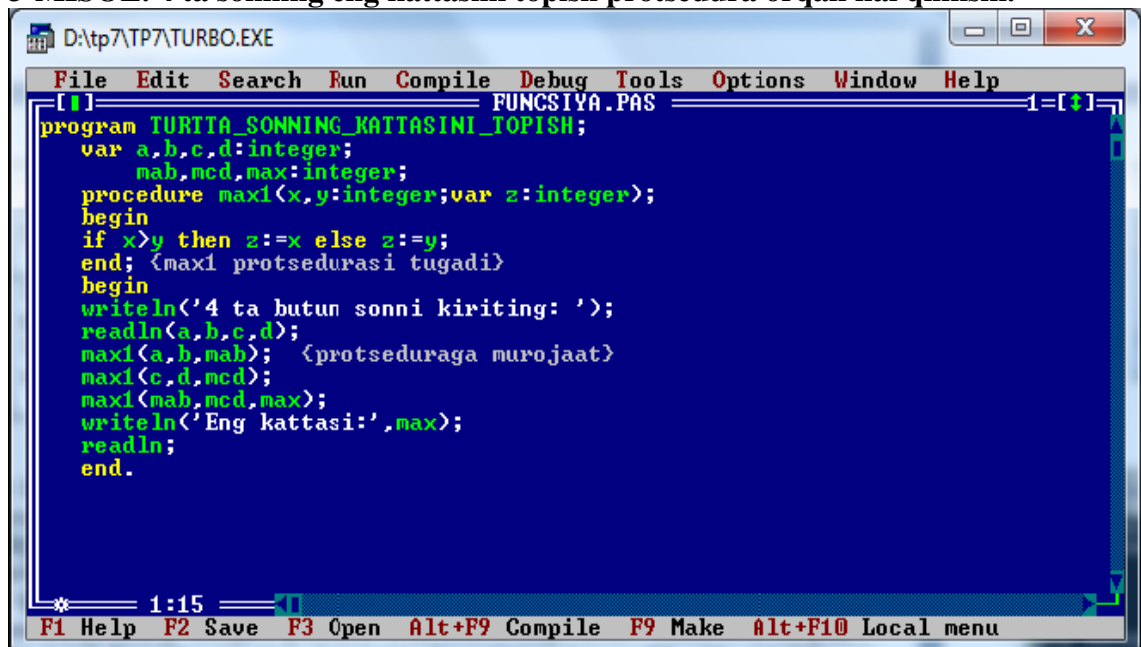
```
File Edit Search Run Compile Debug Tools Options Window Help
[ ] PROCED~1.PAS 1=[+]
program M_1;
uses crt;
var a,z:real; b:integer;
procedure daraja(n:integer;x:real;var y:real);
var i:integer;
begin
  y:=1;
  for i:=1 to n do
    y:=y*x;
  end; {daraja protsedurasi tugadi}
begin clrscr;
  writeln('sonni va darajasini kiriting:');
  write(' ?');
  read(a,b);
  if b=0 then z:=1
  else if b>0 then daraja(b,a,z) {daraja protsedurasiga murojaat qilindi}
  else daraja(-b,1/a,z);
  writeln(z:6:2);
  readln;readln;
end.
11:19
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

NATIJA:



```
D:\tp7\TP7\TURBO.EXE
sonni va darajasini kiriting:
5
3125.00
```

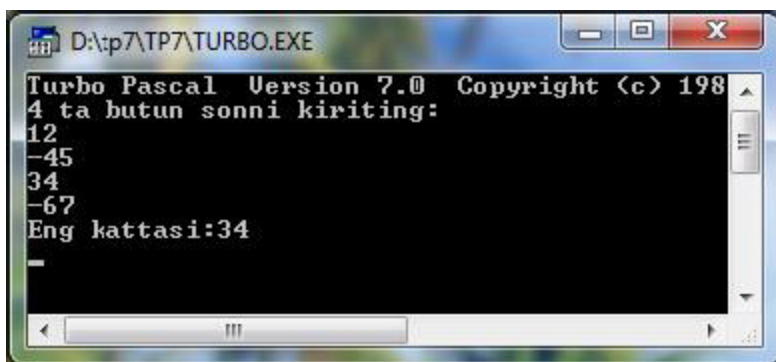
3-MISOL. 4 ta sonning eng kattasini topish protsedura orqali hal qilinsin.



```
D:\tp7\TP7\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
[ ] FUNCSIYA.PAS 1=[+]
program TURITA_SONNING_KATTASINI_TOPISH;
var a,b,c,d:integer;
    mab,mcd,max:integer;
procedure max1(x,y:integer;var z:integer);
begin
  if x>y then z:=x else z:=y;
end; {max1 protsedurasi tugadi}
begin
  writeln('4 ta butun sonni kiriting: ');
  readln(a,b,c,d);
  max1(a,b,mab); {protseduraga murojaat}
  max1(c,d,mcd);
  max1(mab,mcd,max);
  writeln('Eng kattasi:',max);
  readln;
end.
1:15
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

NATIJA:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



```
D:\tp7\TP7\TURBO.EXE
Turbo Pascal Version 7.0 Copyright (c) 198
4 ta butun sonni kiriting:
12
-45
34
-67
Eng kattasi:34
```

LABORATORIYA ISHINI BAJARISH UCHUN TOPSHIRIQLAR

1. Radian qiymatli burchakni gradus qiymatga o‘tkazuvchi protsedura tuzing va undan dasturda
2. foydalaning
3. 10-elementli massiv berilgan. Uning musbat elementlarini 4-ga ko‘paytiruvchi dastur tuzing
4. $1!+2!+\dots+15!$ –ni xisoblash dasturini tuzing.
5. To‘g‘ri chiziq koordinatalariga ko‘ra to‘g‘ri chiziq uzunligini topish dasturini tuzing.
6. Berilgan uchburchak uchlari koordinatalariga ko‘ra, teng tomonli uchburchakligini aniqlash dasturi tuzilsin.
7. Berilgan ikkita sonning EKUB va EKUK larini topish dasturini tuzing.
8. $y=kx+b$ tenglama bilan berilgan to‘g‘ri chiziqda M nuqtalari yotadimi?
9. Berilgan soat, minut va sekundni sekundda aniqlovchi dastur tuzing.
10. Kvadrat tenglamani ildizini topish dasturini tuzing.
11. 4 xonali son berilgan. Div va mod amallaridan foydalanib, xar bir xonadagi sonlar yig‘indisini topish dasturini tuzing.
12. 4 xonali sonni palindromga tekshiruvchi dastur tuzing.
13. Berilgan sonni radiandan o‘lchovdan gradus o‘lchovga o‘tkasuvchi dastur tuzing.
14. Sonlar ustida quyidagi amallarni bajaruvchi dasturni tuzing. 1- qo‘shish, 2- ayirish, 3- ko‘paytirish, 4- bo‘lish.
15. Haqiqiy a, b va c sonlari berilgan. $S=\max(a-b, a, a+b)+\max(a, b+c, a-c)$ ifodani qiymatini topish dasturini tuzing.
16. Agar $a_1=1!, a_2=2!, \dots, a_n=n!$ bo‘lsa, $A(n)$ massivni hosil qilish dasturini tuzing.
17. $X(5,5)$ va $Y(7,7)$ kvadrat massiv berilgan. Bosh dioganalarda joylashgan elementlar yig‘indisini topish dasturini tuzing.
18. Quyidagi fikrni tekshiruvchi dastur tuzing: “Berilgan 3 xonali sonning xonalarida joylashgan sonlar geometrik progressiyani tashkil etadi”.
19. 3 ta $A(15)$, $B(10)$, $C(20)$ massivlari berilgan. Har bir massiv yi‘gindisini hisoblab, kamayish tartibida chop etuvchi dastur tuzing.

PROTSEDURAGA DOIR TOPSHIRIQLAR

1. * dan tashkil topgan satrni chop etuvchi satrni dasturini tuzing. * lar soni protsedura parametric bo‘lsin.
2. Parallelopipedni hajmini, asos yuzasini hisoblash protsedurasini tuzing. Protседura parametrlari: bo‘yi, eni, balandligi.
3. Agar M_1, M_2, M_3 $X(30)$, $Y(15)$, $Z(20)$ massivlarining maksimum elementlari bo‘lsa, $K=(M_1+M_2+M_3)/3$ ni hisoblash dasturini tuzing
4. To‘rtburchakning burchaklari koordinatalari berilgan. To‘rtburchak perimetri va yuzasini topish dasturini tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

5. A(15), V(10), S(20) massivlari berilgan. Massiv elementlarini o‘shish tartibida chop etuvchi dastur tuzing.
6. Agar m_1 va m_2 sonlar X(20) va X(35) massivlarning eng kichik elementlari bo‘lsa, $Z=(m_1+m_2)/2$ ni hisoblash dasturini tuzing.
7. X(5,5) va Y(7,7) massivlarining bosh dioganallarida joylashdan elementlar yig‘indisini topish dasturini tuzing.
8. A(m,n) massivdagi eng ko‘p ketme-ket keluvchi bir xil qiymatli sonlar ketma-ketligini aniqlovchi dastur tuzing.
9. Faqat toq satrlaridan iborat bo‘lgan massiv elementlarining:
 1. yig‘indisini topish.
 2. O‘rtacha arifmetigini topish.
 3. Eng kattasini topish.
 4. Eng kichigini topish dasturini tuzing.
10. Uchburcakni berilgan 3 ta tomoniga ko‘ra perimetrini va yuzasini topish dasturi tuzilsin. Dastur foydalanuvchi tomonidan kiritilgan qiymatlarni tekshirishi kerak va ‘Kiritish xatosi ma’lumotini chop etsin.

10 – LABORATORIYA MASHG‘ULOTI

MAVZU: PASCALDA FAYLLAR BILAN ISHLASH

Mashg‘ulotning maqsadi: Talabalarga mavzu bo‘yicha bilim, ko‘nikma va malaka hosil qilish.

Nazariy qism:

Fayl deb, kompyuter tashqi xotirasining nomlangan soxasiga aytiladi va uning uchta xarakterli xususiyatlari mavjud. Birinchidan, uning nomi mavjud bulib, bu nomdan dasturda foydalaniladi. Ikkinchidan uning komponentlari bir tipga mansub va bu fayl tipidan boshqa barcha tiplar bo‘lishi mumkin. Uchinchidan yangi yaratiluvchi faylning uzunligi haqida uning e‘lon qilish vaqtida “fikr yuritilmaydi” va bu faqat tashqi xotira elementining xajmiga bog‘liq.

Fayl tipi quyidagi uch yo‘llarning biri bilan yaratiladi:

`<nom>=file of<tip>; <nom>=Text; <nom>=File;`

Bu yerda `<nom>`- fayl tipining nomi (to‘g‘ri nomlangan identifikator), `file`, `of` xizmatchi so‘zlar, `<tip>` bu fayl tipidan boshqa barcha tiplar.

Misol: Type

```
Product=record Name:String; Code:Word;End;
Text80=file of String [80];
```

Var

```
F1: File of Char; F2: Text; F3: File; F4: Text80;
F5: File of Product;
```

Fayllarni e‘lon qilish usullariga ko‘ra, ularni uch turga ajratish mumkin:

tiplashirilgan fayllar(File of ... bilan beriladi, yuqoridagi misolda, F1, F4, F5);

matnli fayllar(TextFle tipi bilan aniqlanadi, yuqoridagi misolda, F2);

tiplashirilmagan fayllar(File tipi bilan beriladi, yuqoridagi misolda, F3);

Faylning turi uning saqlanish usulini aniqlaydi va umuman Turbo Paskal tilida oldindan yaratilgan faylni nazorat qilish vositalari mavjud emas va bu vazifani dasturchi o‘z zimmasiga olishi lozim. Fayllar bilan ishlash faqat faylni ochish protsedurasi bajarilgandan so‘ng bajarilishi mumkin. Bu odindan e‘lon qilingan fayl o‘zgaruvchisini yaratilgan yoki

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

yaratilishi lozim bo‘lgan fayl nomi bilan bog‘lash protsedurasi bo‘lib, undan so‘ng fayldan o‘qish yoki unga yozish yo‘nalishi beriladi. Har qanday fayllar (yoki mantiqiy qurilmalar) faylni (mantiqiy qurilmani) ochish maxsus protsedurasi yordamida dasturga u bilan ishlash imkoniyatini beradi. Fayl o‘zgaruvchisi avvaldan yaratilgan fayl nomi bilan quyidagi standart protsedura yordamida bog‘lanadi:

Assign (<fayl o‘zgaruvchisi>,<fayl nomi>) protsedurasi fayl o‘zgaruvchisini fayl nomi bilan bog‘laydi.

AssignFile (<fayl o‘zgaruvchisi >,< fayl nomi >);- bu protseduraning umumiy ko‘rinishi bulib, bu yerda fayl o‘zgaruvchisi - dasturda e‘lon qilingan fayl tipidagi o‘zgaruvchi, fayl nomi esa, fayl nomini yoki ungacha bo‘lgan yo‘lni ifodalovchi matn.

Fayl initsializatsiyasi deb, bu faylga ma‘lumotlarni jo‘natish yoki undan olish yo‘nalishiga aytiladi.

Faylni o‘qish uchun fayl Reset protsedurasi yordamida initsializatsiya qilinadi va bu protseduraning ko‘rinishi quyidagicha:

Reset (<fayl uzgaruvchisi >);

Izoh: fayl o‘zgaruvchisi –avval Assign protsedurasi yordamida mavjud fayl bilan bog‘langan bo‘lishi lozim.

Bu protsedura bajarilishi natijasida fayl o‘qish uchun tayyorlanadi va natijada maxsus ko‘rsatgich bu faylni boshiga, ya‘ni 0-tartib nomerli komponentni ko‘rsatib turadi.

Delphi dasturlash tilida Reset protsedurasi yordamida ochilgan tiplashtirilgan fayllarga read protsedurasi bilan murojaat qilish mumkin. Reset protsedurasi yordamida ochilgan matnli fayllar uchun Write yoki Writeln protseduralaridan foydalanib bo‘lmaydi.

Rewrite (< fayl o‘zgaruvchisi >) protsedurasi fayl o‘zgaruvchisi bilan bog‘langan faylga yozish uchun beriladi va bunda mavjud fayldagi barcha ma‘lumotlar avval “o‘chiriladi” va yangi ma‘lumotlar buyruqlarga ko‘ra faylga yoziladi.

Append (<fayl uzgaruvchisi >) protsedurasi mavjud faylni kengaytirish, ya‘ni unga qushimcha ma‘lumotlarni yozish imkoniyatini beradi.(faqat Text tipidagi fayllar uchun qo‘llaniladi)

Close (<fayl uzgaruvchisi >) protsedurasi faylni yopish uchun qo‘llaniladi, ammo fayl o‘zgaruvchisi bilan bog‘lanish o‘z kuchini saqlaydi.

Quyida keltirilgan ikki protseduradan foydalanish uchun Reset, Rewrite, yoki Append protseduralari yordamida ochilgan fayllar avval yopilgan bo‘lishlari shart.

Rename (<fayl uzgaruvchisi >,<yangi nom>) protsedurasi faylni qayta nomlash uchun ishlatiladi.

Erase (<fayl uzgaruvchisi >) fayli o‘chiradi.

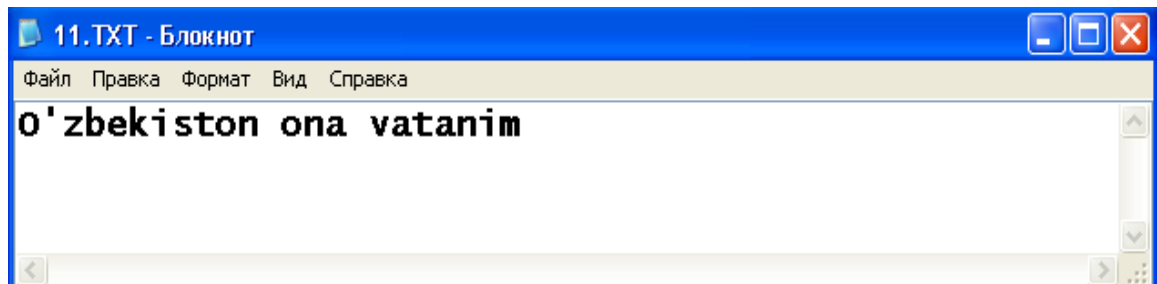
Topshiriq:

1-misol. S-mantiqiy diskda “11.txt” faylni yaratish va unda 'O‘zbekiston ona vatanim' matnini joylashtiruvchi dastur yarating.

```
var f:text;
begin
assign ( f,'c:/11.txt');
rewrite(f);
writeln(f,'O'zbekiston ona vatanim');
close(f);
end.
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

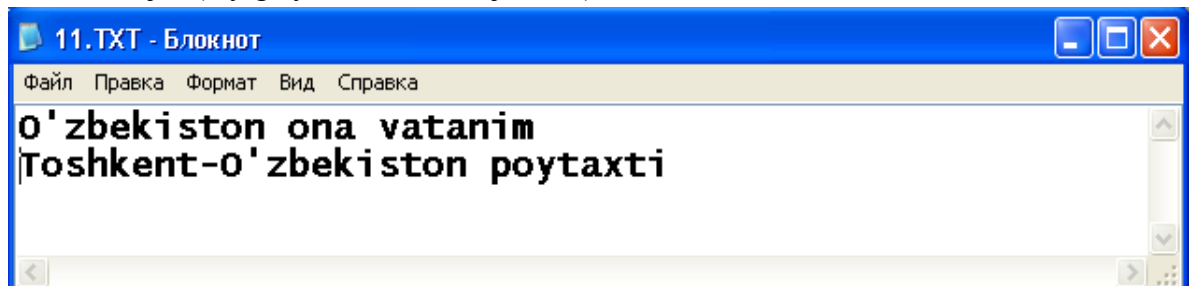
Dastur natijasi(faylga yozish amali bajariladi)::



2-misol. S-mantiqiy diskda joylashgan “11.txt” faylidagi 'O‘zbekiston ona vatanim' matnini “ Toshkent- O‘zbekiston poytaxti” matni bilan to‘ldirish.

```
var f:text;  
begin  
assign ( f,'c:/11.txt');  
rewrite(f);  
writeln(f,'O'zbekiston ona vatanim');  
close(f);  
end.
```

Dastur natijasi(faylga yozish amali bajariladi):

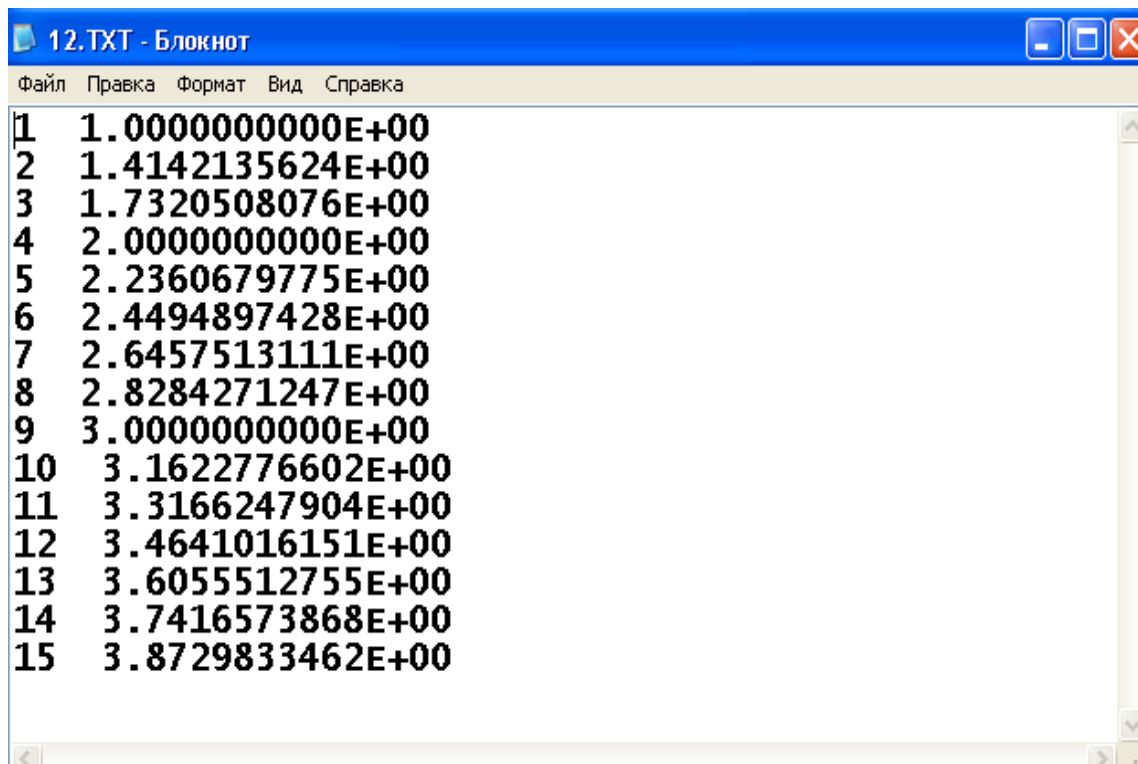


3-misol.

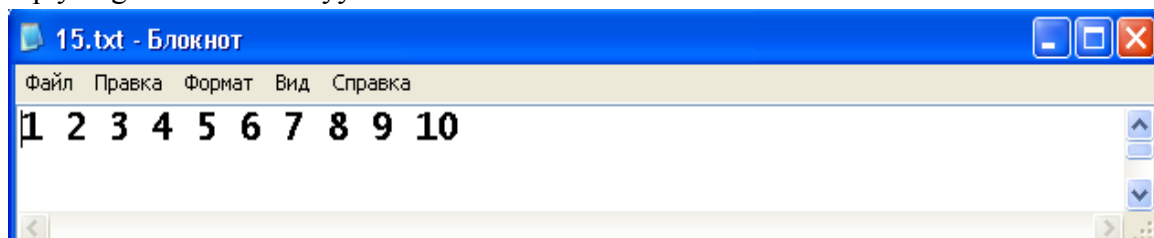
```
var f:text;  
x,i:integer;  
begin  
assign ( f,'c:/12.txt');  
rewrite(f);  
for i:=1 to 15 do  
begin  
write(f,i);  
write(f,' ');  
write(f,sqrt(i));  
writeln(f);  
end;  
close(f);  
end.
```

Dastur natijasi(faylga yozish amali bajariladi):

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA



Fayldan o‘qish amali bajarilishi kuzatish uchun avval S mantiqiy diskda (yoki boshqa manbada) o‘qish uchun faylni tayyorlaymiz. Buning uchun «Bloknot» dasturida «15.pas» faylini quyidagi ko‘rinishda tayyorlab olamiz:



4-misol. S mantiqiy diskdagi «15.pas» faylidan 10 ta soni o‘qib ularning kvadratlarini monitorda aks ettiruvchi dastur yarating.

```
uses crt;  
var f:text;  
x,i:integer;  
begin  
assign ( f,'c:/15.txt');  
reset(f); clrscr;  
for i:=1 to 10 do  
begin  
  read (f,x);  
  writeln(i,'-ning kvadtati ',sqr(x),' ga teng');  
  end;  
  close(f);  
end.
```

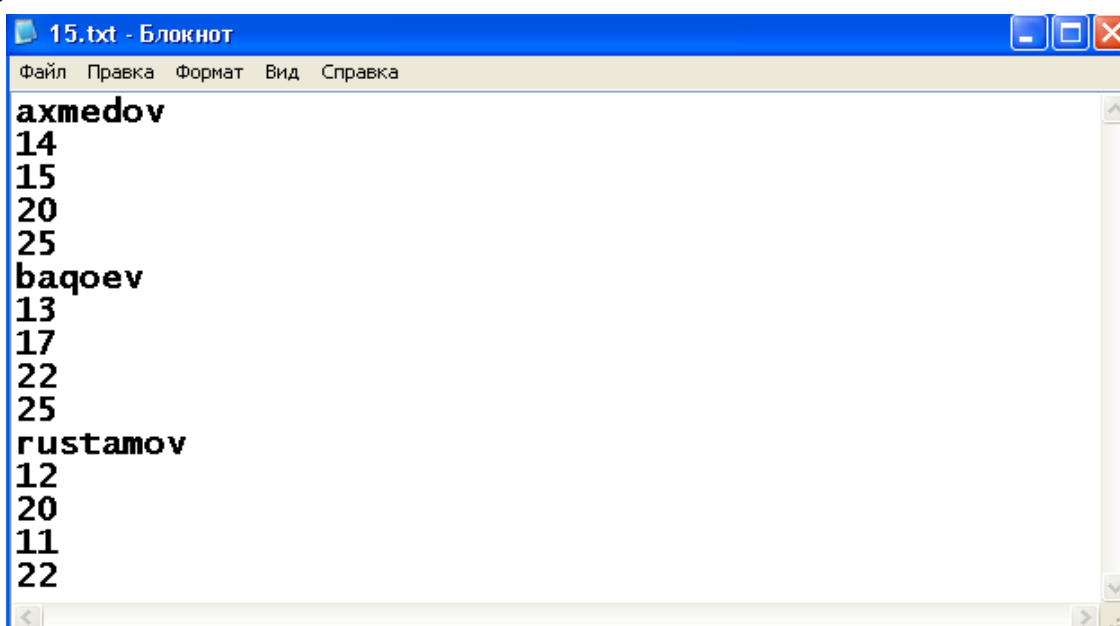
Dastur natijasi:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA



```
C:\> TPX.EXE
1 -ning kbadrati 1 ga tenmg
2 -ning kbadrati 4 ga tenmg
3 -ning kbadrati 9 ga tenmg
4 -ning kbadrati 16 ga tenmg
5 -ning kbadrati 25 ga tenmg
6 -ning kbadrati 36 ga tenmg
7 -ning kbadrati 49 ga tenmg
8 -ning kbadrati 64 ga tenmg
9 -ning kbadrati 81 ga tenmg
10 -ning kbadrati 100 ga tenmg
```

Fayldan o‘qishda faylning oxirini aniqlovchi EOF(f) mantiqiy funksiyasidan foydalanishga misol keltiramiz. Buning uchun siz quyidagi kabi sinfdoshlaringizni 4 fandan olgan test natijalarini aks ettiruvchi «15.txt» hujjatni yaratib kerakli joyda saqlashingiz zarur.



```
15.txt - Блокнот
Файл Правка Формат Вид Справка
axmedov
14
15
20
25
baqoev
13
17
22
25
rustamov
12
20
11
22
```

5-misol. O‘quvchilarning 4 fandan olgan natijalarini fayldan o‘qib, har bir o‘kuvchining familiyasi va ballari yig‘indisini ekranda chop etuvchi dastur yarating.

Dastur ko‘rinishi:

```
uses crt;
var f:text;
fam:string;
a,b,c,d:integer;
begin
assign ( f,'c:/15.txt');
reset(f); clrscr;
while not eof(f) do

begin
    readln (f,fam, a,b,c,d);
    writeln(fam,a+b+c+d);
    end;
close(f);
```

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

end.

(dasturni mustaqil bajarib, xulosa chiqaring)

topshiriq

1. mustaqil bajarib ko‘ring.

```
var f:text;
```

```
x,i:integer;
```

```
begin
```

```
mkdir ('c:/v1'); {yangi katalog ochish uchun}
```

```
assign ( f,'c:/v1/12.txt');
```

```
rewrite(f);
```

```
writeln(f,sqrt(2011));
```

```
close(f);
```

```
end.
```

2.

```
var f:text;
```

```
x,i:integer;
```

```
begin
```

```
assign ( f,'c:/v1/12.txt');
```

```
rewrite(f);
```

```
writeln(f,sqrt(2011));
```

```
close(f);
```

```
rename(f,'c:/v1/y1.txt');
```

```
end.
```

Savol va topshiriqlar.

1. Fayl turdagi to‘zgaruvchi deganda nimani tushinasiz?
2. Paskalda matnli faylni ifodalovchi xizmatchi so‘zni ayting.
3. Assign operatori vazifasini aytib bering.
4. Rewrite operatori vazifasini aytib bering.
5. Rewrite operatori bilan ochilayotgan fayl tashqi xotirada avvaldan mavjud bo‘lsa qanday xodisa yuz beradi?
6. Close operator nima uchun qo‘llaniladi?
7. Append protsedurasini vazifasini aytib bering.
8. Fayldagi ma’lumotlarni ochish uchun qaysi operator yordamida ochiladi.
9. EOF funksiyasini vazifasini aytib bering.

11,12- LABORATORIYA ISHI

MAVZU: GRAFIKA MODULI VA ULAR BILAN ISHLASH. PASKAL TILIDA SODDA SHAKLLAR CHIZISH.

Ishning maqsadi: Talabalarga GraphABC moduli va grafik operatorlar bilan ishlash bo‘yicha yetarli ko‘nikma va malaka hosil qilish.

NAZARIY QISM.

Graph.tpu faylidan iboratdir. Grafik tartibotida ishlash uchun bu fayl kompilyator uchun ishchi fayl bo‘lishi kerak. Dastur boshida Graph.tpu fayli joylashgan katalogga yo‘l ko‘rsatilishi kerak.

a) Grafik holatini o‘rnatish va undan chiqish tartiblari.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Komputer ekranining oddiy ish tartiboti matnli hisoblanadi. Matn tartibotidan grafik tartibotiga o‘tish uchun Graph modulining InitGraph proseduralari ishlatiladi. Uning umumiy ko‘rinishi quyidagicha:

InitGraph (D, M, F); - ekranni grafik holatga o‘tkazish. Bu yerda D- drayver nomeri, M- tartibot nomeri, F- esa kerakli drayver mavjud bo‘lgan faylga yo‘l. Agar F o‘zgaruvchi bo‘sh (F='') satrdan iborat bo‘lsa, drayver joriy katalogdan izlanadi. D va M lar o‘zgaruvchi parametrlardir. InitGraph proseduralari ishga tushirayotgan paytda D ning qiymati 0 ga teng bo‘lsa kerakli drayver va shu drayver uchun optimal grafik tartibot avtomatik tarzda aniqlanadi.

Graph modulida 0 ga teng bo‘lgan Detect o‘zgarmasi ishlatiladi.

Grafik holatidan chiqish yoki monitorni boshlang‘ich ish holatiga o‘tkazish uchun CloseGraph proseduralari ishlatiladi.

CloseGraph- grafik holatidan chiqish.

b) Graph modulining funksiya va proseduralari

Quyida Graph moduliga tegishli prosekura va funksiyalarni qarab chiqamiz.

Koordinatalarni o‘rnatish proseduralari va funksiyalari

Ko‘plab grafik proseduralari va funksiyalarda ekranda joriy o‘rinni ko‘rsatuvchi ko‘rsatkichdan foydalaniladi. Ko‘rsatkichning matn kursoridan farqi shundaki, u ekranda ko‘rinmaydi. Ana shu ko‘rsatkichning holati, shu bilan birga umuman grafik holatda har qanday koordinata ekranning yuqori chap burchagi (0,0) koordinataga nisbatan beriladi. Shunday qilib, ekranning gorizontol koordinatasi chapdan o‘ngga qarab, vertikal koordinatasi esa yuqoridan pastga qarab o‘tib boradi.

GetMaxX va GetMaxY funksiyalari. Joriy ish tartibotida mos ravishda maksimal gorizontol va vertikal koordinatalarini aniqlaydi.

GetX va GetY funksiyalari. Integer turidagi qiymatlar bo‘yicha ekranning gorizontol va vertikal koordinatalarini o‘rnatadi. Agar oyna o‘rnatilmagan bo‘lsa, koordinatalar ekranning chap yuqori burchagiga nisbatan olinadi.

SetViewPort proseduralari - grafik ekranda to‘g‘ri burchakli oyna hosil qiladi. Uning umumiy ko‘rinishi:

Procedure SetViewPort (X1, Y1, X2, Y2: integer; ClipOn: boolean);

Bu yerda X1...Y2 lar oyna burchagining koordinatalari. (X1, Y1)- yuqori chap va (X2, Y2) o‘ng quyi burchak koordinatalari, ClipOn – Boolean tipidagi ifoda.

MoveTo proseduralari - ko‘rsatkichni yangi joriy holatga o‘rnatadi. Uning umumiy ko‘rinishi:

Procedure MoveTo(X,Y:integer); Bu yerda X,Y ko‘rsatkich-ning gorizontol va vertikal yo‘nalishlar bo‘yicha yangi koordinatalari. Koordinata ekranning yuqori chap burchagiga asosan olinadi.

ClearDevice proseduralari – grafik oynani tozalaydi.

Uning umumiy ko‘rinishi:

Procedure ClearDevice; Bu proseduralari bajarilganda ekran tozalanadi, ko‘rsatkich ekranning yuqori chap burchagiga joylashadi va ekran fon rangi bilan bo‘yaladi.

Ko‘pburchaklar hosil qilish funksiya va proseduralari Rectangle proseduralari – ko‘rsatilgan burchak koordinatalari bo‘yicha to‘g‘ri to‘rtburchak hosil qilish. Uning umumiy ko‘rinishi:

Procedure Rectangle (X1, Y1, X2, Y2: integer); bu yerda X2 – X1... to‘g‘ri to‘rtburchak burchaklarining koordinatalari. (X1, Y1) – yuqori chap to‘rtburchak va (X2, Y1) quyi o‘ng burchak koordinatalari. To‘g‘ri burchak joriy rang va joriy chiziq qalinligi bo‘yicha hosil qilinadi.

DrawPoly proseduralari – egilish nuqtalarining koordinatalari bo‘yicha ixtiyoriy siniq chiziqlarni chizadi. Uning umumiy ko‘rinishi:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Procedure DrawPoly(N:Word;Var Points); Bu yerda N – egilish nuqtala-rining soni; Points – nuqta koordinatalarini tashkil etuvchilar. Chizish jarayonida joriy rang va joriy chiziq qalinligi tanlanadi.

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

1-masala. Dastur tarkibida GetDriverName funksiyasidan foydalanib, yuklangan grafikali drayver nomini aniqlang.

```
uses Graph;
var
Driver, Mode,Error:integer;  dr:string;
begin
Driver:= Detect;
InitGraph(Driver,Mode, 'c:\tp\bgi');
dr:=GetdriverName;
Writeln(dr, ' ');
Writeln("Enter" tugmasini bosing...:50);
Readln;
CloseGraph ;
end.
```

2-masala. GetMaxMode funksiyasidan foydalanib, adapterning grafikali holatlarining maksimal miqdorini aniqlovchi dastur yarating.

```
uses Graph;
var
Driver, Mode,Error:integer;  d:integer;
begin
Driver:= Detect;
InitGraph(Driver,Mode, 'c:\tp\bgi');
d:=GetMaxMode;
restorecrtMode;
Writeln(d);
Writeln("Enter" tugmasini bosing...:50);
Readln;
CloseGraph ;
end.
```

3-masala. GetModName funksiyasi yordamida ekranning yechim holatlarini va adapterning ish holati nomini parametr bo‘yicha aniqlang.

Dastur ko‘rinishi:

```
uses Graph;
var
Driver, Mode,Error,i:integer;  ds:string;
begin
Driver:= Detect;
InitGraph(Driver,Mode, 'c:\tp\bgi');
writeln (getdrivername);
for i:=0 to 3 do
begin
ds:=GetModeName(i);
Writeln(ds);
end;
Writeln("Enter" tugmasini bosing...:50);
```


“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Readln;
CloseGraph;
end.

4-masala. Keltirilgan dastur tahlilini keltirning.

```
Program Cube;
Uses CRT,Graph;
Type a_t=array [1..8] of integer;
Procedure Cubik(x,y:a_t;c:integer);
Begin
SetColor(c);
line(x[1],y[1],x[2],y[2]);
line(x[2],y[2],x[3],y[3]);
line(x[3],y[3],x[4],y[4]);
line(x[4],y[4],x[1],y[1]);
line(x[5],y[5],x[6],y[6]);
line(x[6],y[6],x[7],y[7]);
line(x[7],y[7],x[8],y[8]);
line(x[8],y[8],x[5],y[5]);
line(x[1],y[1],x[5],y[5]);
line(x[2],y[2],x[6],y[6]);
line(x[3],y[3],x[7],y[7]);
line(x[4],y[4],x[8],y[8]);
End;
Var x1,x:a_t;
y1,y:a_t;
d,m:integer;
x0,y0:integer;
t:real;
i:integer;
Begin
d:=detect; m:=detect; InitGraph(d,m,"");
x[1]:=300; y[1]:=220; x[5]:=320;
y[5]:=200;
x[2]:=300; y[2]:=260; x[6]:=320;
y[6]:=240;
x[3]:=340; y[3]:=260; x[7]:=360;
y[7]:=240;
x[4]:=340; y[4]:=220; x[8]:=360;
y[8]:=200;
x0:=310; y0:=200;
t:=0;
repeat
for i:=1 to 8 do begin
x1[i]:=x0+round((x[i]-x0)*cos(t)-(y[i]-
y0)*sin(t));
y1[i]:=y0-round((x[i]-x0)*sin(t)+(y[i]-
y0)*cos(t));
end;
Cubik(x1,y1,white);
delay(1000);
Cubik(x1,y1,black);
t:=t+0.05;
until KeyPressed;
CloseGraph;
End.
```

LABORATORIYA ISHI UCHUN TOPSHIRIQLAR:

Modullar va grafik imkoniyatlardan foydalanib dastur tuzish:

- a) Satri ma'lumotlar bilan ishlash.
- b) Grafik ma'lumotlar bilan ishlash.
- c) Grafiklar bilan ishlash.
 1. Ekranda aylana va uning ichida joylashgan uchburchak tasvirini hosil qilish dasturini tuzing.
 2. Ekranda uy rasmini hosil qilish dasturini tuzing.
 3. $y = ax^2 + bx + c$ funksiya grafigini chizing.
 4. $y = ax^2 + bx$ funksiya grafigini chizing.
 5. $y = ax^2$ funksiya grafigini chizing.
 6. Ekranda televizor tasvirini hosil qilish dasturini tuzing.
 7. Ekranda to'rtburchak va uning ichida joylashgan ochiq konvert tasvirini hosil qilish dasturini tuzing.
 8. Ichma-ich joylashagan ellips, aylana, kvadrat va uchburchakni hosil qilish dasturini tuzing.
 9. Ekranda stol tasvirini hosil qilish dasturini tuzing.
 10. Ekranda tog' va orqasida ko'rinib turgan quyosh rasmini hosil qi ling.
 11. Ikkita parallel va perpendikulyar tekisliklar chizing.
 12. Svetofor rasmini hosil qilish dasturini tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

13. Piramida va uning kesmalarini chizing.
14. Ekkranda shaxmat doskasi hosil qilish dasturini tuzing.
15. Ichma-ich joylashagan ellips, aylana, kvadrat va uchburchakni hosil qilish dasturini tuzing.
16. $y = \frac{ax^2 + bx}{x - 3}$ funksiya grafigini chizing.
17. $y = \frac{ax^2}{x + 3}$ funksiya grafigini chizing.
18. $y = 2 \lg(x - 2)$ funksiya grafigini chizing.
19. $y = \lg(x + 2)$ funksiya grafigini chizing.
20. $y = 2 \operatorname{tg}(x - 2)$ funksiya grafigini chizing.
21. $y = \sin(x - 2)$ funksiya grafigini chizing.
22. Silindr tasvirini hosil qilish dasturini tuzing.
23. Uch o‘lchovli koordinatalar sistemasini chizing, yo‘nalishlarini ko‘rsating va o‘qlarini x, y, z bilan belgilang.
24. Stol va uning ustida gul vazasini chizish dasturini tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

FOYDALANILGAN ADABIYOTLAR

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Сер: Классические учебники: COMPUTER SCIENCE. М.: МЦНМО, –960с., 2004.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2010 y.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
4. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyoy”, 2011y.
5. T. X. Holmatov, N. I. Toyloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta‘minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
6. M. Ashurov, M. Mirmaxmudov, Sh. Saipaev. Zamonauiy dasturlash tillari fanidan laboratoriya ishlari. T. : TDPY, 2008 й.
7. Матрос Д.Ш., Поднебесова Г.Б. Теория алгоритм. Учебник для педагогического образования. М.: Бинум. Лаборатория знаний, - 2008. -202с.
8. Макконелл Дж. Основы современных алгоритмов. 2-доп.изд., М.: ТЕХНОСФЕРА, 366с., 2004.
9. Вирт Н. Алгоритмы и структуры данных. С примерами на Паскале. Санкт-Петербург, 352с., 2005.
10. Рейест Р. и др. Алгоритмы: построение и анализ. М., Мир, 1994.
11. Носов В.А. Основы теории алгоритмов, анализа их сложности. Курс лекций. М., 139с., 1992.
12. Малышко В.В. Алгоритмы и алгоритмические языки. Конспект лекций для студентов Ташкентского филиала МГУ, 68с., 2006.
13. Пильщиков В.Н., Абрамов В.Г., Вылиток А.А., Горячая И.В. Машина Тьюринга и алгоритмы Маркова. Решение задач. (Учебно-методическое пособие) Московский государственный университет им. М.В.Ломоносова, Факультет вычислительной математики и кибернетики, Москва, 47с., 2006.
14. Yunusov A.S. Matematik mantiq va algoritmlar nazariyasi elementlari. Samarqand davlat universiteti nashriyoti, 182 b, 2012y.
15. www.de.uspu.ru/Informatics/metodes/DPP/F/08/1/Index.htm.
16. www.ziyonet.uz.

III. MUSTAQIL TA'LIM MAVZULARI

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Talaba mustaqil ta’limni tayyorlashda fanning xususiyatlarini hisobga olgan holda quyidagi shakllardan foydalanish tavsiya etiladi:

- darslik va o‘quv qo‘llanmalar bo‘yicha fan boblari va mavzularini o‘rganish;
- tarqatma materiallar bo‘yicha ma‘ruzalar qismini o‘zlashtirish;
- avtomatlashtirilgan o‘rgatuvchi va nazorat qiluvchi dasturlar bilan ishlash;
- maxsus adabiyotlar bo‘yicha fanlar bo‘limlari yoki mavzulari ustida ishlash;
- talabaning o‘quv va ilmiy-tadqiqot ishlarini bajarish bilan bog‘liq bo‘lgan fanlar bo‘limlari va mavzularni chuqur o‘rganish;
- zamonaviy dasturlash muhitidan foydalanishga mustaqil o‘rganishlari;

Tavsiya etilayotgan mustaqil ishlarning mavzulari:

T/r	Mustaqil ta’lim mavzulari	Berilgan topshiriqlar	Bajarish muddati	Hajmi (soatda)
I SEMESTR				
1.	Algoritm tushunchasi va ulardan foydalanish.	Algoritm tushunchasi va ulardan foydalanishni o‘rganish.	1- hafta	2
2.	Algoritm va uning xossalari. Algoritm turlari.	Algoritmning xossalari va turlarini o‘rganish.	2- hafta	2
3.	Algoritm nazariyasiga asos solgan olimlar.	Algoritm nazariyasiga asos solgan olimlar haqidagi ma’lumotlarni o‘rganish.	3- hafta	2
4.	Algoritmilar samaradorligini baholash.	Murakkab turdagi algoritmilar va ularning qo‘llanilishiga doir masalalar yechish.	4- hafta	2
5.	Algoritmilar tahlili.	Algoritmilar tahlilini o‘rganish.	4- hafta	2
6.	Algoritmilar ishlab chiqish metodlari.	Algoritmilar ishlab chiqish metodlarini o‘rganish.	5- hafta	2
7.	Chiziqli va tarmoqlanuvchi algoritmilar.	Chiziqli, tarmoqlanuvchi jarayonlarni algoritmilashni o‘rganish.	6- hafta	2
8.	Takrorlanuvchi algoritmilar.	Takrorlanuvchi jarayonlarni algoritmilashni o‘rganish.	6- hafta	2
9.	Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.	Paskal dasturlash tilining kelib chiqishi va qo‘llanishini o‘rganish.	7- hafta	2
10.	Standart funksiyalar bilan ishlash.	Standart funksiyalar bilan ishlashni o‘rganish.	7- hafta	2
11.	Interpretatorlar va kompilyatorlar.	Interpretatorlar va kompilyatorlar haqidagi ma’lumotlarni o‘rganish.	8- hafta	2
12.	Chiziqli va tarmoqlanuvchi dasturlar.	Paskal dasturlash tilidan chiziqli va tarmoqlanuvchi dasturlar tuzish va ularni tahlil qilish.	8, 9- haftalar	4
13.	Takrorlanuvchi dasturlar	a) Dars jadvalini chiqaruvchi.	10- hafta	4

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

		b) Alfavit bo‘yicha saralovchi dastur. v) Test topshiriqlarini bajaruvchi (To‘g‘ri javobni belgilovchi) dasturlar tuzing.		
14.	Tanlash operatorlari bilan ishlash	Tanlash operatorlari vazifasini o‘rganish.	11- hafta	2
15.	Paskalda bir o‘lchovli va ikki o‘lchovli massivlar bilan ishlash.	Mustaqil ravishda 10 dan dastur tuzing va uning bajarilishini asoslab bering.	11, 12- haftalar	4
16.	Paskalda satriy kattaliklar bilan ishlash.	Sana-vaqt turi bilan ishlashga doir masalalar yechish.	12, 13- haftalar	4
17.	Qism dasturlar-funksiyalar yordamida dasturlar tuzish.	Funksiyalardan foydalanib dastur tuzish.	13, 14- haftalar	4
18.	Qism dasturlar-protseduralar yordamida dasturlar tuzish	Bir necha xil Protседura va funksiyalarning qo‘llanishiga doir misollar.	14- hafta	4
19.	Paskalda fayllar bilan ishlash.	Paskalda fayllar bilan ishlashga doir dasturlar tuzish.	15- hafta	4
20.	Ma‘lumotlarni fayllardan o‘qish. Ma‘lumotlarni fayllarga yozish.	Mustaqil ravishda 10 dan dastur tuzing va uning bajarilishini asoslab bering. Ma‘lumotlarni fayllarga yozish bo‘yicha dasturlar tuzish.	16- hafta	4
21.	Grafika moduli va ular bilan ishlash.	Pascal dasturlash tilida grafik imkoniyatlar va ulardan foydalanish.	17- hafta	2
22.	Paskal tilida sodda geometrik shakllar chizish.	Figuralarni harakatga keltiruvchi dasturlar tuzish.	18- hafta	2
23.	Paskalda kichik hajmdagi amaliy dasturlar yaratish.	Paskalda kichik hajmdagi amaliy dasturlar yaratish.	18- hafta	4
24.	Paskalda amaliy dasturlar yaratish.	Paskalda amaliy dasturlar yaratish.	19- hafta	2
	JAMI:			66

IV.GLOSSARIY

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

<p>Atamaning o‘zbek, ingliz, rus tillarida nomlanishi</p>	<p>Atamaning nomlanishi</p>
<p>algoritm algorithm алгоритм</p>	<p>Vazifani bajarishga qaratilgan aniq belgilangan qoidalarining tartiblangan chekli to‘plami. Dastlabki ma'lumotlarni oxirgi natijaga o‘tkazuvchi hisoblash jarayoni orqali masala echimini ko‘rsatuvchi amallar mazmuni va ketma-ketligi. Algoritm atamasi o‘rta asrlardayashabijodetganbuyukzzbekmatematigi Al-Xorazmiy Nomidan kelib chiqqan. U IX asrning 825 yilidayoq o‘zikashf etgan o‘nli sanoq tizimida to‘rt arifmetika amallarini bajarish Qoidalarini bergan. Arifmetika amallarini Bajarish jarayoni esa Al xorazm deb atalgan. Bu atama 1747 yildanboshlabalgorismus, 1950 yilga kelib algorifm deb amataldi.</p>
<p>Ta’rif Definition Определение</p>	<p>Muayyan tushuncha yoki terminning unga xos muhim belgilarini aks ettiruvchi qisqacha ifodasi; biron predmet yoki voqea hodisaning mazmun mohiyatini ochib, tushuntirib berish</p>
<p>To‘plam Set множества</p>	<p>To‘plam tushunchasi matematikanin gboshlang‘ich tushunchasidir.</p>
<p>schema Sheme схема</p>	<p>(yun. schema — qiyofa, tashqiko‘rinish) —1) muayyan qurilma, inshoot, mashina va boshqalarning umumiy muhim tomonlari shartli belgilar bilan masshtabsiz ifodalangan chizma;2)biron narsaning umumiy tasvir, bayoni, ifodasi.</p>
<p>Chiziq line линия</p>	<p>geom.ning asosiy tushunchalaridan biri. To‘g‘ri chiziq geometriyada boshlang‘ich (ta’riflanmaydigan) tushuncha deb olinadi.</p>
<p>Konstruksiya design Конструкция</p>	<p>(lot. constructs -tuzilish, qurilish) — 1) mashina, inshootyoki o‘zeldetallarningtuzilishsxemasi, shuningdek, mashina, inshoot, o‘zellarhamdaularningdetallari. K. da kerakliqismvaelementlariningshaklihamdao‘zarojoylashishi, ularnibiriktirishusullari, o‘zarota’sirivakandaymateriallardanyasalishihisobgaolinadi</p>
<p>Aniqlik vatusunarlilik Accurate and understandable Точная и понятно</p>	<p>algoritmda ijroshiga berilayotgan ko‘rsatmalar aniq mazmunda bo‘lishi tushuniladi. SHunki ko‘rsatmalardagi noaniqliklar mo‘ljallangan maqsadga erishishga olib kelmaydi. Ijroshiga tavsiya etiladigan ko‘rsatmalar tushunarli mazmunda bo‘lishi shart, aks holda ijroshi uni bajara olmaydi.</p>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

<p>Natijaviylik The resulting Результирующий</p>	<p>algoritmda chekli qadamlardan so‘ng albatta natija bo‘lishi tushuniladi. Shuni ta’kidlash joizki, algoritm avvalfdan ko‘zlangan maqsadga erishishga olib kelmasligi ham mumkin. Bunga ba’zan algoritmning noto‘g‘ri tuzilgani yoki boshqa xatolik sabab bo‘lishi mumkin, ikkinshi tomondan, qo‘yilgan masala ijodiy yeshimga ega bo‘lmasligi ham mumkin. Lekin salbiy natija ham deb qabul qilinadi.</p>
<p>Diskretlik Sample Дискретность</p>	<p>algoritmlarni chekli qadamlardan tashkil qilib bo‘laklash imkoniyati tushuniladi.</p>
<p>Samaradorligi The effectiveness эффективность</p>	<p>Har xil masalalarni yechish uchun mo‘ljallangan, turli xil algoritmlar, samaradorligi bo‘yicha sezilarli darajada farq qiladi.</p>
<p>Masala qo‘yilishi question put to вопрос поставлен</p>	<p>Masalani yechishdan oldin, uni berilishini aniq shakllantirib olish zarur. Bu jarayon to‘g‘ri savollarni aniqlash bo‘lib, savollar quyidagicha bo‘lishi mumkin:</p> <ol style="list-style-type: none"> 7. Dastlabki berilgan masala shartlarida hamma iboralar tushunarlimi? 8. Nima berilgan? 9. Nimani topish kerak? 10. Yechimni qanday ta’riflash kerak? 11. Qaysi berilganlar yetarli emas va hammasi kerakmi? 12. Qanaqa mumkinliklar qabul qilingan? <p>Albatta, bulardan tashqari boshqa savollarni ham ishlatish mumkin, yoki ayrim savollarni bir necha bor takror ishlatishga to‘g‘ri keladi.</p>
<p>Bayt Byte Байт</p>	<p>Sakkiz bitga teng bo‘lgan axborot miqdorining asosiy o‘lchov birligi</p>
<p>Belgi Symbol СИМВОЛ</p>	<p>1.Kodlashtirilishi, kompyuterga kiritilishi va uning displey ekranida ko‘rsatilishi</p> <p>mumkin bo‘lgan hisoblash tizimi tomonidan ruxsat etilgan belgilar to‘plami elementi. Belgilarga harf, raqam, qavs va ayirish belgilari hamda arifmetika amallari belgilari, shuningdek maxsus, boshqaruv va soxta grafik belgilar kiradi. Hisoblash tizimi foydalanish mumkin bo‘lgan deyarli barcha belgilar kompyuter xotirasiga klaviatura yordamida bitta yoki bir guruh tugmalar bosilishi orqali kiritilishi mumkin.</p> <p>2.Biror narsani belgilash uchun ishlatiladigan bitta yoki birin-ketin joylashgan belgilar ketma-ketligi</p>
<p>Buyruq Instruction Команда</p>	<p>Bajarilishi zarur bo‘lgan amalning ta’rifi. Topshiriqlarni boshqarish tilini ko‘rsatish, dastur operatori, boshqaruvchi signal va foydalanuvchi talablari buyruq (ko‘rsatma) hisoblanadi. Barcha hollarda, buyruqlar yordamida ma’lumotlarni qayta ishlash tizimida ma’lumotlar jo‘natish yoki tizimlar orasida ma’lumotlar uzatish jarayonlarini boshqarish amalga oshiriladi. Har bir buyruq amal kodidan tashkil topib, qaysi ob’ektga va nima qilish kerakligini, olingan natijani qaerga yuborish kerakligini xabar qiladi. Amallarni bajarishda ishlatiladigan buyruqlar yig‘masi buyruq tili bilan aniqlanadi. Buyruqlar arifmetik,</p>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

	<p>mantiqiy, kiritish-chiqazish, ma’lumotlarni uzatish turlariga bo‘linadi. O‘zaro bog‘langan buyruqlar ketma-ketligi makrobuyruq deb ataladi.</p>
<p style="text-align: center;">Dastur Software Программа</p>	<p>1.Hisoblash mashinasiga algoritmnini beruvchi ko‘rsatmalarning ketma-ketligi. Dastur kompyuter tomonidan qaysi tartibda, qaysi ma’lumotlar ustidan va qaysi amallar bajarilishi va natija qaysi shaklda taqdim etilishini ko‘rsatadi. Kompyuterning boshqarish qurilmasi mashina buyruqlari ketma-ketligi shaklida berilgan dasturni qabul qiladi.</p> <p>2.Ma’lum natijani olish uchun kompyuter va boshqa kompyuter qurilmalari faoliyati uchun mo‘ljallangan jami ma’lumot va buyruqlarni taqdim etishning ob’ektiv shakli</p>
<p style="text-align: center;">Dasturchi Programmer Программист</p>	<p>Kompyuter ta’minotini yozuvchi kishi.Kompyuter dasturchisi deb kompyuter dasturlash bo‘yicha mutaxassis yoki turli xil dasturiy ta’minot uchun kodlar yozuvchi mutaxassislarni atashadi. Amaliyotda dasturlashni formal tarzda o‘qitadigan insonlarni ham dasturiy tahlilchi deb nomlash mumkin. Dasturchining asosiy ishlov tilini (Lisp, Java, Delphi, C++, h.k.) nomiga qo‘shib xuddi veb muhitida ishlovchilarni veb nomlari bilan ishlatganday nomlashadi. Dasturchi atamasi ta’minot yozuvchisiga, ta’minot muhandisiga, kompyuter olimi yoki ta’minot tahlilchisiga nisbatan ishlatilishi mumkin.</p>
<p style="text-align: center;">Dasturiy ta’minot Software Программное обеспечение</p>	<p>Axborotga ishlov berish tizimining barcha yoki ba’zi dasturlari, tartiblari, qoidalari va ularga tegishli hujjatlar. Dasturiy vositalar ular yozilgan tashuvchidan qat’iy nazar intellektual mahsulot hisoblanadi</p>
<p style="text-align: center;">Dasturlash tili Programmi ng language Язык программирования</p>	<p>Kompyuterlar uchun dasturlar (ko‘rsatmalar yig‘masi) yoziladigan, uni u yoki bu harakatlarni bajarishga majbur qiladigan rasmiy til. Dasturlash tilida yozilgan ko‘rsatmalar dastlabki kod deb ataladi.</p> <p>Dastlabki kod kompyuterda amalda bajarilishidan avval, uni mashina kodiga yo bo‘laklab talqin qilish, yo batamom talqin qilish zarur. Dasturlash tilining ta’rifi quyidagilarni o‘z ichiga oladi:</p> <ul style="list-style-type: none"> - mumkin bo‘lgan belgilar ro‘yxatini; - zaxiralangan so‘zlar ro‘yxatini; - sintaksisni (belgilarni va zaxiralangan so‘zlarni birikmalash usullarini); - semantikani (dasturlash tilining birikmalar ma’nosi).
<p style="text-align: center;">Dasturlash Programmi ng Программирован ие</p>	<p>1.Dasturni yaratish jarayoni. Dasturlash o‘z ichiga dasturga bo‘lgan talablar tahlili va uni ishlab chiqish va yaratishning barcha bosqichlarini oladi: algoritm, ma’lumotlar tuzilmasi va dasturlash tizimini tanlash; dasturni yozish (kodlash) va ma’lumotlarni tayyorlash; dasturni sozlash va sinovdan o‘tkazish; dastur uchun qo‘shimcha hujjatlarni yaratish.</p> <p>2.Hisoblash mashinalari uchun dasturlar tuzish va amalga oshirish</p>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

	uslublari va vositalarini ishlab chiqish bilan shug‘ullanuvchi fan.
Ekran dasturi Screen program Экранная программа	Ekkranda aks ettirilgan qurilma harakatlarini uning ishiga taqlid natijasi sifatida ko‘rsatadigan amaliy dastur. Ekran dasturi qator vazifalar bajarilishini juda sodda va ko‘rgazmali namoyish qiladi. Masalan, dastur kalkulyatorning klaviaturasini kompyuter ekranida aks ettiradi. Shu usulda yaratilgan taqvim ekranda varaqlanayotgan sahifalarni vaqt va sana bilan birga aks ettiriladi.
Ekran Screen Экран	1.Axborotni aks ettirish uchun ishlatiladigan yuzaning qismi yoki qurilma. q: monitor 2.Xavf. Tarmoqlararo muhofazalash vositasi. q: brandmauer. 3.Telekom. O‘tkazgichlar atrofida ularning ostidan o‘rnatilgan o‘tkazgichlarga qarshi maydonlar tomonidan ko‘rsatiladiganbelektrostatik yoki elektromagnit ta’sirni bartaraf etish uchun joylashtirilgan metall qatlam. 4.App.Elekt va magnit maydonlar o‘tishini sezilarli kamaytiruvchi yoki ob’ektlar yoxud odamlarning xavfli kuchlanish ostida ishlayotgan sxema komponentlari yoki elementlariga tasodifan tegib ketishning oldini oluvchi ekran yoki korpus
Menyu Menu Меню	1.Foydalanuvchi grafik interfeysining (GUI) qismi. Dasturning yoki operatsion tizimning mumkin bo‘lgan harakatlari ro‘yxati. Dastur menyusi qatori oynaning ustki qismida joylashgan. Matnbop menyu ma’lum ob’ektga sichqonchanning o‘ng tugmasi bosilganda paydo bo‘ladi. Har bir ob’ekt alohida amallar ro‘yxatiga ega. 2.Ovozni qayta ishlash tizimi aytadigan dastur amallarini davom ettirish variantlari. Foydalanuvchi o‘z tanlovini bildirish uchun javoban bir yoki bir necha so‘z aytishi lozim.
Chiziqli dasturlar Linear programs Линейные программы	Chiziqli dasturlar tuzilish jihatidan aniq ketma-ketlikdagi amallardan iborat. Bu amallarning bajarilishi dasturda keltirilish tartibi bilan bog‘liq, ya’ni yozilgan ketma-ketlikda bajariladi. Har bir masalani yechishdan oldin algoritmi tuzib olinadi.
Tarmoqlanuvchi dasturlar Branching programs Ветвящихся программы	Shartga muvofiq bajariladigan ko‘rsatmalar bilan tuziladigan dasturlar tarmoqlanuvchi dasturlar deyiladi.
Blok – sxema Flow Diagram Блок-схема	Blok-sxema- keng tarqalgan sxema turi bo‘lib, har bir alohida qadam bir-birini bog‘lovchi chiziqlar bilan bog‘langan turli xil formadagi bloklar bilan algoritm yoki jarayonlarni tasvirlaydi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Chiziqli algoritmlar Linear algorithms Линейные алгоритмы	Hech qanday shart tekshirilmaydigan va tartib bilan faqat ketma – ket bajariladigan algoritmlar Chiziqli algoritmlar deyiladi.
Tanlash (Case) operatori (Case) operator Case оператор	Bu operator bir necha yo‘nalish bo‘yicha tarmoqlanishni ta’minlab beruvchi(tanlashni amalga oshiruvchi) operator hisoblanadi. Case <tanlash indeksi-kalid> of <tanlash holatlari ro‘yxati-elementlari> else <operatorlar >end;
While sikli While-do loop Цикл While	While sikli (takrorlanadigan amallar bajarilishi uchun avval shart tekshiriladi).
for-do sikli for-do loop Цикл for-do	Dasturlash jarayonida ayrim hollarda bir yoki bir necha amallarni bir necha marotaba takrorlab bajarish zarurati tug‘iladi. Parametrik sikllarning umumiy ko‘rinishi quyidagicha: <i>For <Sikl parametri>:= <a> to do</i> <operator yoki operatorlar> Bu yerda a-parametr bosh qiymatiga, b-parametr oxirgi qiymatiga teng.
Repeat sikli Repeat loop for-do Repeat	Repeat sikli (takrorlanadigan amallar kamida bir marotaba bajarilib so‘ngra shart tekshiriladi).
Yozuv Record Запись	1.Ko‘rilayotgan ob’ektni (masalan, buyum – uning tartib raqami, nomlanishi, o‘lchamlari, narxi, uni yasash uchun ishlatilgan modda) ifodalab beruvchi ma’lumot elementlari majmui. 2.Tashqi va amaliy xotira orasidagi ma’lumotlar almashuvining tuzilmaviy birligi.
Irearxiya Hierarchy Иерархия	Bog‘liq (bog‘langan) ob’ektlarning tartiblashtirilgan majmui. Ularning o‘zaro bog‘liqligini belgilovchi bir necha pog‘onasini o‘z ichiga oladi.
Natija Result Результат	Dasturning ishlashi
Click Click Click	Klavishani bosish va qo‘yib yuborish
Matrisa Matrix Матрица	F maydonning mn ta a_{ij} ($i = 1, m, j = 1, n$) elementlaridan tuzilgan jadval F maydon ustidagi mxn tartibli matrisa deyiladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

<p>Massiv Massive Массив</p>	<p>Massiv-jadval ko‘rinishidagi miqdor bo‘lib, u ma‘lum sondagi bir turli va tartiblangan elementlar majmuasidan iborat. Massiv nomga ega-bir xil tipga tegishli bo‘lgan va tartiblangan elementlar to‘plamidir.</p>
<p>Element indeksi The index element Индекс элемента</p>	<p>Elementning tartib raqami</p>
<p>Massiv elementlari The elements of the array Элементы массива</p>	<p>Ma‘lum sondagi bir turli va tartiblangan elementlar majmuasi</p>
<p>Vektor Vector Вектор</p>	<p>Yo‘nalishga ega bo‘lgan kesma</p>
<p>Satriy kattaliklar String type Сатривая типъ</p>	<p>Turbo Pascal dasturlash tilida String tipidan matnli ma‘lumotlar bilan ishlashda foydalaniladi. Bu tipning har bir elementi xotirada 1 bayt joy egallovchi belgilardan iborat bo‘ladi.</p>
<p>Belgi Symbol Символ</p>	<p>1.Kodlashtirilishi, kompyuterga kiritilishi va uning displey ekranida ko‘rsatilishi mumkin bo‘lgan hisoblash tizimi tomonidan ruxsat etilgan belgilar to‘plami elementi. Belgilarga harf, raqam, qavs va ayirish belgilari hamda arifmetika amallari belgilari, shuningdek maxsus, boshqaruv va soxta grafik belgilar kiradi. Hisoblash tizimi foydalanish mumkin bo‘lgan deyarli barcha belgilar kompyuter xotirasiga klaviatura yordamida bitta yoki bir guruh tugmalar bosilishi orqali kiritilishi mumkin.</p> <p>2.Biror narsani belgilash uchun ishlatiladigan bitta yoki birin-ketin joylashgan belgilar ketma-ketligi</p>
<p>Buyruq Instruction Команда</p>	<p>Bajarilishi zarur bo‘lgan amalning ta‘rifi. Topshiriqlarni boshqarish tilini ko‘rsatish, dastur operatori, boshqaruvchi signal va foydalanuvchi talablari buyruq (ko‘rsatma) hisoblanadi. Barcha hollarda, buyruqlar yordamida ma‘lumotlarni qayta ishlash tizimida ma‘lumotlar</p>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

	<p>jo‘natish yoki tizimlar orasida ma’lumotlar uzatish jarayonlarini boshqarish amalga oshiriladi. Har bir buyruq amal kodidan tashkil topib, qaysi ob’ektga va nima qilish kerakligini, olingan natijani qaerga yuborish kerakligini xabar qiladi. Amallarni bajarishda ishlatiladigan buyruqlar yig‘masi buyruq tili bilan aniqlanadi. Buyruqlar arifmetik, mantiqiy, kiritish-chiqazish, ma’lumotlarni</p> <p>uzatish turlariga bo‘linadi. O‘zaro bog‘langan buyruqlar ketma-ketligi makrobuyruq deb ataladi.</p>
<p style="text-align: center;">Dastur Software Программа</p>	<p>1. Hisoblash mashinasiga algoritmnini beruvchi ko‘rsatmalarning ketma-ketligi. Dastur</p> <p>kompyuter tomonidan qaysi tartibda, qaysi ma’lumotlar ustidan va qaysi amallar</p> <p>bajarilishi va natija qaysi shaklda taqdim etilishini ko‘rsatadi. Kompyuterning</p> <p>boshqarish qurilmasi mashina buyruqlari ketma-ketligi shaklida berilgan dasturni qabul qiladi. Dasturni mashina tilida</p> <p>yaratish – noqulay va katta mehnat talab qiluvchi jarayon. Shuning uchun kompyuter uchun dastur inson tomonidan dasturlash</p> <p>tillaridan birida yaratilib, keyin esa kompyuterning o‘zi ushbu dasturni mashina tiliga o‘giradi.</p> <p>2. Ma’lum natijani olish uchun kompyuter va boshqa kompyuter qurilmalari faoliyati uchun mo‘ljallangan jami ma’lumot va buyruqlarni taqdim etishning ob’ektiv shakli</p>
<p style="text-align: center;">Funksiya function функция</p>	<p>dasturning bir qismi bo‘lib, unga dasturda bir necha marotaba murojaat qilinishi mumkin. Umuman dasturning bir qismi bo‘lib, unga dasturda bir necha marotaba murojaat qilinishi mumkin. Umuman:</p> <p>a) agarda parametrlar qiymatlari yordamida yagona natija olinadigan bo‘lsa, funksiyadan foydalanish mumkin;</p> <p>b) qism dasturning yakuniy natijasi albatta funksiya nomiga o‘zlashtirilishi shart.</p>
<p style="text-align: center;">Buyruq Instruction Команда</p>	<p>Bajarilishi zarur bo‘lgan amalning ta’rifi. Topshiriqlarni boshqarish tilini ko‘rsatish, dastur operatori, boshqaruvchi signal va foydalanuvchi talablari buyruq (ko‘rsatma) hisoblanadi. Barcha hollarda,</p> <p>buyruqlar yordamida ma’lumotlarni qayta ishlash tizimida ma’lumotlar jo‘natish yoki tizimlar orasida ma’lumotlar uzatish jarayonlarini boshqarish amalga oshiriladi. Har bir buyruq amal kodidan tashkil topib, qaysi ob’ektga va nima qilish kerakligini, olingan natijani qaerga yuborish kerakligini xabar qiladi. Amallarni bajarishda ishlatiladigan buyruqlar yig‘masi buyruq tili bilan aniqlanadi. Buyruqlar arifmetik, mantiqiy, kiritish-chiqazish, ma’lumotlarni</p> <p>uzatish turlariga bo‘linadi. O‘zaro bog‘langan buyruqlar ketma-ketligi</p>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

	makrobuyruq deb ataladi.
Dastur Software Программа	1.Hisoblash mashinasiga algoritmni beruvchi ko‘rsatmalarning ketma-ketligi. Dastur kompyuter tomonidan qaysi tartibda, qaysi ma’lumotlar ustidan va qaysi amallar bajarilishi va natija qaysi shaklda taqdim etilishini ko‘rsatadi. Kompyuterning boshqarish qurilmasi mashina buyruqlari ketma-ketligi shaklida berilgan dasturni qabul qiladi. 2.Ma’lum natijani olish uchun kompyuter va boshqa kompyuter qurilmalari faoliyati uchun mo‘ljallangan jami ma’lumot va buyruqlarni taqdim etishning ob’ektiv shakli
Ekran dasturi Screen program Экранная программа	Ekran aks ettirilgan qurilma harakatlarini uning ishiga taqlid natijasi sifatida ko‘rsatadigan amaliy dastur. Ekran dasturi qator vazifalar bajarilishini juda sodda va ko‘rgazmali namoyish qiladi. Masalan, dastur kalkulyatorning klaviaturasini kompyuter ekranida aks ettiradi. Shu usulda yaratilgan taqvim ekranida varaqlanayotgan sahifalarni vaqt va sana bilan birga aks ettiriladi.
Menyu Menu Меню	1.Foydalanuvchi grafik interfeysining (GUI) qismi. Dasturning yoki operatsion tizimning mumkin bo‘lgan harakatlari ro‘yxati. Dastur menyusi qatori oynaning ustki qismida joylashgan. Matnbop menyu ma’lum ob’ektga sichqonchani o‘ng tugmasi bosilganda paydo bo‘ladi. Har bir ob’ekt alohida amallar ro‘yxatiga ega. 2.Ovozni qayta ishlash tizimi aytadigan dastur amallarini davom ettirish variantlari. Foydalanuvchi o‘z tanlovini bildirish uchun javoban bir yoki bir necha so‘z aytishi lozim.
Ta’rif Definition Определение	muayyan tushuncha yoki terminning unga xos muhim belgilarini aks ettiruvchi qisqacha ifodasi; biron predmet yoki voqeahodisaning mazmunmohiyatini ochib, tushuntirib berish
To‘plam Set множества	To‘plam tushunchasi matematikaning boshlang‘ich tushunchasidir.
schema Scheme схема	(yun. schema — qiyofa, tashqi ko‘rinish) — 1) muayyan qurilma, inshoot, mashina va boshqalarning umumiy muhim tomonlari shartli belgilar bilan masshtabsiz ifodalangan chizma; 2) biror narsaning umumiy tasviri, bayoni, ifodasi.
Chiziq line линия	geom.ning asosiy tushunchalaridan biri. To‘g‘ri chiziq geometriyada boshlang‘ich (ta’riflanmaydigan) tushuncha deb olinadi.
Konstruksiya design Конструкция	(lot. constructs -tuzilish, qurilish) — 1) mashina, inshoot yoki o‘zel va detallarning tuzilish sxemasi, shuningdek, mashina, inshoot, o‘zellar hamda ularning detallari. K. da kerakli qism va elementlarining shakli hamda o‘zaro joylashishi, ularni birlashtirish usullari, o‘zaro ta’siri va qanday materiallardan yasalishi hisobga olinadi

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

insident intsident инцидент	Agar x_1 va x_2 lar qandaydir qirraga (x_i, x_j) ga tegishli bo‘lsa, u holda ushbu qirra x_i va x_j “insident” deyiladi, x_i va x_j lar esa qo‘shni nuqtalar deyiladi.
Qisman graf partial graph частичный граф	Graf $G_0(x_0, E_0)G(x, E)$ ning qisman grafi deb ataladi, agarda u berilgan grafning barcha uchlariga ega bo‘lib, ammo barcha qirralariga ega bo‘lmasa, balki qisman qirralariga ega bo‘lsa, ya’ni $x_0 = x, E_0 \in E$
Planar graf Planar graph Планарный граф	Graf (tekis) planardeyiladi, agardaushbugrafgaizomorfo‘lgangrafnitekislikdaqirralarikesishmaganh oldatasvirlashmumkinbo‘lsa.
Graf yo‘li Line graph Маршрут	m uzunlikdagi marshrut deb grafning qirralarini shunday ketma ketligiga aytiladiki yonma-yon bo‘lgan qirralarini uchlari uchma-uch tushishlari kerak.
Bog‘langan graf Connected graph Связанный граф	Grafning ikki uchi bog‘langan deyiladi, agar shu uchlarni birlashtiruvchi yo‘l bo‘lsa. Agar grafning har qanday uchini birlashtiruvchi marshrut mavjud bo‘lsa, bunday graf bog‘langan graf deyiladi
zanjir chain цепь	Barcha qirralari turli bo‘lgan (yo‘l) marshrut zanjir deb ataladi. Agar zanjir turli uchlardan o‘tsa, u oddiy zanjir deb ataladi.
daraxt tree дерево	Siklga ega bo‘lmagan bog‘langan graf daraxt deb ataladi, uning qirralari esa shoxlaridir.
O‘rmon forest лес	Siklsiz bog‘lanmagan graf o‘rmon deb ataladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

V. ILOVALAR

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM VAZIRLIGI

Ro‘yxatga olindi:
№ BD 5110700 2.04

201__ - yil _____

Oliy va o‘rta maxsus ta’lim vazirligi

201__ - yil _____

ALGORITMLAR FAN DASTURI

Bilim sohasi:	100000 – Gumanitar
Ta’lim sohasi:	110000 – Pedagogika
Ta’lim yo‘nalishi:	5110700 – Informatika o‘qitish metodikasi

Toshkent – 2019

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

O‘zbekiston Respublikasi oliy va o‘rta maxsus ta’lim vazirligining 201__-yil ____
dagi ____-sonli buyrug‘ining ____-ilovasi bilan fan dasturi ro‘yxati tasdiqlangan.

Fan dasturi Oliy va o‘rta maxsus, kasb-hunar ta’limi yo‘nalishlari bo‘yicha O‘quv-
uslubiy birlashmalar faoliyatini Muvofiqlashtiruvchi Kengashning 201__-yil ____
dagi ____-sonli bayonnomasi bilan ma’qullangan.

Fan dasturi Nizomiy nomidagi Toshkent davlat pedagogika universitetida ishlab
chiqildi

Tuzuvchilar:

M.O'.Ashurov- «Informatika o‘qitish metodikasi» kafedrasida katta o‘qituvchisi

Sh.A.Sattarova - «Informatika o‘qitish metodikasi» kafedrasida o‘qituvchisi

Taqrizchilar:

Axatov A.R.

– SamDU Amaliy matematika va informatika fakulteti
dekani, texnika fanlari doktori, professor

Yuldasheva U.T.

– TTESI qoshidagi akademik litsey direktori, texnika fanlari
nomzodi

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

I. O‘quv fanning dolzarbligi va oliy kasbiy ta’limdagi o‘rni

Ta’lim sohasidagi tub islohatlarning asosiy maqsadi jahon andozalari asosida bilimlar berish va raqobatdosh kadrlar tayyorlashdir. SHuning uchun ta’lim tizimidagi 5110700 – Informatika o‘qitish metodikasi yo‘nalishida o‘qitiladigan fanlar ham zamonaviy fanlardan hisoblanadi. Ushbu namunaviy dastur bugungi kunning zamonaviy bilimlari bilan yangilangan va qayta ishlangan dastur bo‘lib, unda fanning nazariy va amaliy jihatlariga alohida e’tibor qaratilgan. Mazkur fan dasturi bakalavriat yo‘nalishi: 5110700 – Informatika o‘qitish metodikasi ta’lim yo‘nalishida taxsil olayotgan talabalarning o‘zlashtirishi lozim bo‘lgan bilimlari va unga qo‘yiladigan talablar asosida tuzilgan bo‘lib, bo‘lajak fan o‘qituvchisi egallashi kerak bo‘lgan bilimlar va ko‘nikmalar majmuini o‘z ichiga oladi:

- algoritmik tillar, algoritm va uning xossalari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari;

- steklar, navbatlar, daraxtlar, algoritmlarning tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, saralashning Sheyker, Shella usullari, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, kattaliklar va ularning tiplari, buruqlar tizimi va operatorlari, massivlar, chiziqli, tarmoqlanuvchi va takrorlanuvchi operatorlar, funktsiya va protseduralar, fayllar bilan ishlash, tilning grafik imkoniyatlari;

«Algoritmlar» fani insonlarda axborot muhitida ma’lum bir dunyoqarashni shakllantirishga hizmat qilishi bilan bir qatorda, uning axboriy madaniyatni egallashida asosiy rol o‘ynaydi. Bugungi «Axborot» asrida yoshlarning kompyuter savodxonligini oshiribgina qolmay, balki ma’lumotlar bazalari bilan ishlash imkoniyatlarini oshiradi. Umumiy o‘rta ta’lim maktablari, akademik litsey va kasb – hunar kollejlarda «Informatika» mutahassislaridagi fanlarni o‘qitish uchun kadrlarni tayyorlab beradi.

II. O‘quv fanining maqsadi va vazifasi

Fanni o‘qitishdan maqsad - bo‘lajak informatika o‘qituvchisining kasbiy sohasida egallashi lozim bo‘lgan bilimlar va amalda qo‘llash uchun ko‘nikma va makalalarni shakllantirish va rivojlantirishdan iborat. Ushbu dasturda har bir kasb egasi uning faoliyat ko‘rsatish turidan qat’iy nazar egallashi kerak bo‘lgan tayanch nazariy va amaliy ma’lumotlarni o‘z ichiga oladi.

Fanning o‘qitishning vazifasi - algoritm tushunchasi va uning xossalari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar bilan tanishtirish;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

- algoritmik tillarning asosiy tushunchalari: steklar, navbatlar, daraxtlar, algoritmlar tahlili kabilar haqida ma’lumotlar berish.

- saralash algoritmlari va ulardan foydalanish usullari haqida tasavvurlar hosil qilish.

borasidagi bilimlar va amalda qo‘llash uchun ko‘nikma va makalalarni shakllantirish va rivojlantirishdan iborat.

Fan bo‘yicha talabalarning bilim, ko‘nikma va malakalariga quydagi talablar qo‘yiladi: Algoritmlar o‘quv fanini o‘zlashtirish jarayonida amalga oshiriladigan masalalar doirasida bakalavr:

- algoritm va uning xossalari, algoritmik tillar, steklar, navbatlar, daraxtlar, algoritmlar tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, Sheyker usulida saralash, Shell usulida saralash, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari, chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlarga doir *to‘g‘risida tasavvurga ega bo‘lishi*;

- algoritmlar, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar, steklar, navbatlar, daraxtlar, algoritmlar tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, Sheyker usulida saralash, Shell usulida saralash, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari, chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlarni *bilishi va ulardan foydalana olishi*;

-algoritmik tillarning asosiy tushunchalaridan foydalanib sodda va tadbqiqiy masalalar yechish *ko‘nikmalariga ega bo‘lishi lozim*.

III. Asosiy nazariy qism (ma’ruza mashg‘ulotlari)

Fanning nazariy mashg‘ulotlari mazmuni

1-Modul. Algoritmlar tushunchasi va mohiyati

1-mavzu. Algoritm tushunchasi va ulardan foydalanish

Algoritm tushunchasi va ular haqida ma’lumotlar. Algoritmlar tarixi. Algoritm xossalari, turlari va uning berilish usullari.

2-mavzu. Algoritmlar samaradorligini baholash

Xotiraviy samara, vaqt samarasi. Algoritmlarning murakkablik darajasi. Algotimlarning taqqoslash usullari.

3-mavzu. Tanlash va joylashtirish turkumidagi murakkablikga ega saralash algoritmlari

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Saralash tushunchasi. Saralash algoritmlari. Tanlash va joylashtirish usulida saralash, O‘shib borish va kamayish tartibida saralash, qo‘shish usulida saralash, Joyida abstrakt qo‘shib saralash, Yuqoridan pastga qo‘shib saralash.

4-mavzu. Almashish usulida saralash, saralashning Sheyker usuli

Almashish usuli mohiyati, saralashning Sheyker, pufakcha va piramida usullari.

5-mavzu. Algoritmlar tahlili

Algoritmlarning baholash va ularning tahlili. Kiruvchi berilganlar sinfi. Xotira bo‘yicha murakkablik. Tahlil va maqsad.

6-mavzu. Algoritmlarni ishlab chiqish metodlari

Algoritmlarni ishlab chiqishning o‘liga xos jihatlari. Strukturaviy algoritmlar, Prosedurali algoritmlar metodi, Algoritmlar konstruksiyasi. Algoritmlar ekvivalent qayta ishlash. Toraytiruvchi o‘zgartirishlar. Formal usulni matematikaga bog‘liq bo‘lmagan muammoga qo‘llash.

7-mavzu. Rekursiya va rekursiv funksiyalar

Rekursiya tushunchasi. Funksiya va rekursiv funksiyalar. Rekursiv funksiyalar turlari. Oddiy va vositali rekursiya. Matematikada rekursiyning qo‘llanilishi. Fibonachchi funksiyasi.

8-mavzu. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv

Qidiruv masalasi, Qidiruv usullari. Yozuvlarni oddiy ko‘rib chiqish usuli, Ketma-ketlik usulida qidiruv, Binar qidiruv, Interpolyatsiya usulida qidiruv, Binar daraxt va Fibonachchi qidiruvlar, Muvozanatlashgan (Balansirlangan) daraxt bo‘yicha qidiruv, Bor usulida qidiruv, h-Hashlash usulida qidiruv, Interval bo‘yicha izlash.

9-mavzu. Rabin-Karp algoritmi

Rabin-Karp algoritmi haqida. Qidiruv algorimlari. Rabin-Karp algoritmi xesh-funksiya. Algoritm ahamiyati.

10-mavzu. Graflar bilan ishlovchi sodda algoritmlar

Graflar nazariyasi, Graflarni tipik qo‘llanilishi, Graflar terminologiyasi, Qism graf, orentirlangan va orentirlanmagan graflar, Graflar izomorfizmi.

11-mavzu. Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari

Paskal tili dasturlash tilining alifbosi, Pascal dasturida asosiy fayllar va funksional tugmalar, O‘zgarimas va o‘zgaruvchi miqdorlar, Standart funksiyalar va algebraik ifodalar, O‘zlashtirish va ma’lumotlarni ekranga chiqarish operatorlari, Ekran bilan ishlash operatorlari.

12-mavzu. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar

Pascalda chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar haqida ma’lumotlar, qiymat berish, tarmoqlanuvchi operatorlar, shartli va shartsiz o‘tish operatorlari, siklik operatorlar.

2-Modul. Paskal dasturlash tilining imkoniyatlari

13-mavzu. Paskalda massivlar

Massivlar haqida tushuncha, bir o‘lchovli va ikki o‘lchovli massivlar, dinamik massivlar.

14-mavzu. Paskalda satriy kattaliklar bilan ishlash

Satriy kattaliklar, satriy kattaliklar bilan ishlovchi funksiya va proseduralar, satriy kattaliklardan foydalanishning o‘ziga xos xususiyatlari.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

15-mavzu. Qism dasturlar-funksiyalar

Qism dastur, qism dasturlar-funksiyalar, qismdastur-funksiya tuzilishi, formal, lokal va haqiqiy parametrlar, dasturda qism dasturlardan foydalanish.

16-mavzu. Qism dasturlar-proseduralar

Qism dastur, qism dasturlar-proseduralar, qismdastur-proseduralar tuzilishi, formal, lokal va haqiqiy parametrlar, dasturda qism dasturlardan foydalanish.

17-mavzu. Paskalda fayllar bilan ishlash

Paskalda fayllar bilan ishlash, tiplashtirilgan va tiplashtirilmagan fayllar, fayllar bilan ishlovchi standart funksiya va proseduralar, fayllarga yozish, fayllardan o‘qish.

18-mavzu. Grafika moduli va ular bilan ishlash

Paskal dasturlash tilining grafik imkoniyatlari, grafik adapterlar, graph moduli funksiya va proseduralari.

19-mavzu. Paskal tilida sodda shakllar chizish

Paskalda sodda shakllar chizuvchi proseduralar, sodda shakllar chizishning o‘ziga xos xususiyatlari.

IV. Amaliy mashg‘ulotlar bo‘yicha ko‘rsatma va tavsiyalar

Amaliy mashg‘ulotlar bo‘yicha quyidagi mavzular tavsiya etiladi:

Fan bo‘yicha rejalashtirilgan amaliy mashg‘ulotlar davomida nazariy bilimlar mustahkamlanadi. Amaliy mashg‘ulotlarda tadbirlar keng qo‘llanilayotgan asosiy matematik usullarni o‘rgatishga e‘tiborni kuchaytirish lozim. Ma‘lum sababalarga ko‘ra ma‘ruzaga kirmagan va murakkab bo‘lgan tushunchalarni amaliy mashg‘ulotlarda ko‘rib o‘tish maqsadga muvofiqdir.

Amaliy mashg‘ulotlarni o‘tkazishda quyidagi didaktik taomiyllarga amal qilinadi:

amaliy mashg‘ulotlarining maqsadini aniq belgilab olish;

o‘qituvchining innovatsion pedagogik faoliyati bo‘yicha bilimlarni chuqurlashtirish imkoniyatlariga talabalarda qiziqish uyg‘otish;

talabada natijani mustaqil ravishda qo‘lga kiritish imkoniyatni ta‘minlash;

talabani nazariy-metodik jihatdan tayyorlash;

amaliy mashg‘ulotlarda nafaqat aniq mavzu bo‘yicha bilimlarni yakunlash, balki talabalarni tarbiyalash manbai hamdir. Amaliy mashg‘ulotlarda talabalar muayyan masala bo‘yicha mavjud bo‘lgan yoki mustaqil tarzda kichik ishchi guruhlari yordamida hosil qilingan algoritmlarni muhokama qiladilar. Mazkur mavzularga oid test masalalar tuzib, ular asosida tuzilgan dasturlar majmuasini tuzadilar va kompyuterda olingan natijalarni birgalikda tahlil qiladilar. Amaliy mashg‘ulotlarning taxminiy tavsiya etiladigan mavzulari:

1. Algoritm tushunchasi va ulardan foydalanish
2. Algoritm samaradorligini baholash
3. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.
4. Almashish usulida saralash, saralashning Sheyker usuli
5. Algoritm tahlili
6. Algoritmni ishlab chiqish metodlari
7. Rekursiya va rekursiv funksiyalar
8. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv
9. Rabin-Karp algoritmi
10. Graflar bilan ishlovchi sodda algoritmlar
11. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.
12. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar
13. Paskalda massivlar
14. Paskalda satriy kattaliklar bilan ishlash
15. Qism dasturlar-funksiyalar

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

16. Qism dasturlar-protseduralar
17. Qism dasturlar-funksiya va proseduralarga doir dasturlar tuzish
18. Paskalda fayllar bilan ishlash.
19. Grafika moduli va ular bilan ishlash.
20. Paskal tilida sodda shakllar chizish

Amaliy mashg‘ulotlarni tashkil etish bo‘yicha kafedra professor-o‘qituvchilari tomonidan ko‘rsatma va tavsiyalar ishlab chiqiladi. Unda talabalar asosiy ma‘ruza mavzulari bo‘yicha olgan bilim va ko‘nikmalarini amaliy masalalarga dasturlar tuzish orqali bilimlarini yanada boyitadilar. Shuningdek, darslik va o‘quv qo‘llanmalar asosida talabalar bilimlarini mustaxkamlashga erishish, tarqatma materiallardan foydalanish, ilmiy maqolalar va tezislarni chop etish orqali talabalar bilimini oshirish, masalalarning dasturini tuzish, mavzular bo‘yicha ko‘rgazmali qurollar tayyorlash va boshqalar tavsiya etiladi.

V. Laboratoriya mashg‘ulotlar bo‘yicha ko‘rsatma va tavsiyalar

Laboratoriya mashg‘ulotlarida talabalar kompyuter yordamida grafik ma‘lumotlarni hosil qilish va amalda uning natijalarini ko‘rib, ularni tahlil qiladi va xulosalar chiqaradilar. Laboratoriya mashg‘ulotlari uchun tavsiya etiladigan mavzular:

1. Algoritm tushunchasi va ulardan foydalanish
2. Algoritm samaradorligini baholash
3. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.
4. Almashish usulida saralash, saralashning Sheyker usuli
5. Algoritm tahlili
6. Algoritmni ishlab chiqish metodlari
7. Rekursiya va rekursiv funksiyalar
8. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv
9. Rabin-Karp algoritmi
10. Graflar bilan ishlovchi sodda algoritmlar
11. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.
12. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar
13. Paskalda massivlar
14. Paskalda satriy kattaliklar bilan ishlash.
15. Qism dasturlar-funksiyalar
16. Qism dasturlar-protseduralar
17. Paskalda fayllar bilan ishlash.
18. Grafika moduli va ular bilan ishlash.
19. Paskal tilida sodda shakllar chizish

VI. Mustaqil ta‘lim va mustaqil ishlar

Mustaqil ta‘lim uchun tavsiya etiladigan mavzulari:

1. Algoritm tushunchasi va ulardan foydalanish
2. Algoritm va uning xossalari
3. Algoritm turlari
4. Algoritm nazariyasiga asos solgan olimlar
5. Tarmoqlanuvchi algoritmlar
6. Takrorlanuvchi algoritmlar
7. Algoritm samaradorligini baholash

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

8. Saralash usullari
9. Pufakcha usulida saralash
10. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.
11. Qo‘shish usulida saralash
12. Piramida usida saralash
13. Almashish usulida saralash, saralashning Sheyker usuli
14. Algoritmlar tahlili
15. Eyler sikllari va zanjirlari. Eyler teoremasi.
16. Algoritmlarni ishlab chiqish metodlari
17. Evklid algoritmi
18. Rekursiya va rekursiv funksiyalar
19. Qidiruv usullari: binar qidiruv
20. Fibonachchi qidiruv
21. Binar daraxt bo‘yicha qidiruv
22. Rabin-Karp algoritmi
23. Graflar bilan ishlovchi sodda algoritmlar
24. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.
25. Standart funksiyalar bilan ishlash
26. Tanlash operatorlari bilan ishlash
27. Chiziqli va tarmoqlanuvchi dasturlar
28. Takrorlanuvchi dasturlar
29. Paskalda bir o‘lchovli massivlar.
30. Paskalda ikki o‘lchovli massivlar.
31. Paskalda satriy kattaliklar bilan ishlash.
32. Qism dasturlar-funksiyalar yordamida dasturlar tuzish
33. Qism dasturlar-protseduralar yordamida dasturlar tuzish
34. Paskalda fayllar bilan ishlash.
35. Ma’lumotlarni fayllardan o‘qish.
36. Ma’lumotlarni fayllarga yozish.
37. Grafika moduli va ular bilan ishlash.
38. Paskal tilida sodda geometrik shakllar chizish
39. Chiziqli ro‘yhatlar
40. Tarmoqlar. Daraxtlar.
41. Izlash algoritmlari.
42. Rekursiv funksiyalar.
43. Interpretatorlar va kompilyatorlar
44. Tanlash operatori bilan ishlash
45. Grahp moduli bilan ishlash.
46. Paskalda sodda shakll chizish
47. Paskalda kichik hajmdagi amaliy dasturlar yaratish
48. Paskalda amaliy dasturlar yaratish

Fan bo‘yicha kurs ishi. Fan bo‘yicha kurs ishi rejalashtirilmagan

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Foydalaniladigan adabiyotlar ro‘yxati

Asosiy adabiyotlar:

1. M.Aripov, M.Muhammadiyev. Informatika, informasion texnologiyalar. Darslik. T.: TDYuI, 2005 y.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
3. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyo”, 2011y.
4. T. X. Holmatov, N. I. Tayloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta‘minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
5. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T. ;, “O‘qituvchi”, 2011 y.
6. B.Mo‘minov. Informatika. O‘quv qo‘llanma. T.: “Tafakkur-bo‘stoni”, 2014 y.

Qo‘shimcha adabiyotlar

1. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига киришиш тантанали маросимига бағишланган Олий Мажлис палаталарининг кўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Шавкат Миромонович. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қондаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий яқунлари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
3. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
4. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (*Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда*)
5. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт-Петербург, “БХВ-Петербург” 2003 г.
6. В. М. Пестиков, А. Н. Маслобоев. Turbo PASKAL 7. 0. Изучаем на примерах. Санкт-Петербург. : “БХВ-Петербург”, 2004 г.
7. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : “Питер”, 2003 г.
8. В. Т. Безручко. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.
9. Меняев Михаил Федорвич. Информационные технология управления. Москва, «Издательский ОмегаЛ», 2003 г.
10. В.Д.Колдаев. Основы алгоритмизации и программирования. Учебный пособие, Москва ИД “Форум”- ИНФРА-М 2006 г.
11. Thomas H. Cormen. Intruduction to algorithms. Third Edition. Massachusetts Institute of Technology. The MIT Press. London 2009. 1292-p.
12. Algorithms, Fourth Edition (Deluxe): Book and 24-Part Lecture Series 1st Edition ,

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Addison-Wesley Professional , USA, 2015

13. Цой, Маргарита и др. Создание электронных учебников. Т. : “Ўзбекистон миллий энциклопедияси”, 2007 г.

Internet saytlari

1. www.ziyonet.uz – Axborot ta’lim portali
2. www.edu.uz – Oliy va o‘rta maxsus ta’lim vazirligi portali
3. www.tdpu.uz – Nizomiy nomidagi TDPU rasmiy sayti
4. [http:// corel.Deamiart.ru/](http://corel.Deamiart.ru/).
5. www.amazon.com
6. <http://www.ctc.msiu.ru/materials/Book1,2/index1.html>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM VAZIRLIGI

NAVOIY DAVLAT PEDAGOGIKA INSTITUTI

Ro‘yxatga olindi:

№ _____
« ____ » _____ 2019 yil.

«Tasdiqlayman»

O‘quv ishlar bo‘yicha prorektor

A.Kushakov _____

« ____ » _____ 2019 yil.

ALGORITMLAR FANINING ISHCHI O‘QUV DASTURI

Bilim sohasi:	100000 – Gumanitar
Ta’lim sohasi:	110000 – Pedagogika
Ta’lim yo‘nalishi:	5110700- Informatika o‘qitish metodikasi

Navoiy-2019

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Ushbu ishchi o‘quv dasturi O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligining va 201__-yil __-avgustdagi __-sonli buyrug‘i bilan fan dasturi tasdiqlangan va Oliy va o‘rta maxsus, kasb-hunar ta’limi yo‘nalishlari bo‘yicha O‘quv-uslubiy birlashmalar faoliyatini Muvofiqlashtiruvchi Kengashning 201__-yil __-avgustdagi __-sonli bayonnomasi bilan ma’qullangan.

Tuzuvchilar:

Ibragimov Alimjan Artikbayevich	- Navoiy davlat pedagogika instituti «Informatika o‘qitish metodikasi» kafedrasida dotsenti, fizika-matematika fanlari nomzodi;
Xamroyeva Dilafro‘z Namozovna	- Navoiy davlat pedagogika instituti «Informatika o‘qitish metodikasi» kafedrasida katta o‘qituvchisi;
Absobirov Samariddin qurbonovich	- Navoiy davlat pedagogika instituti «Informatika o‘qitish metodikasi» kafedrasida o‘qituvchisi.

Taqrizchilar:

Yodgorov G‘.R.	- NavDPI “Informatika o‘qitish metodikasi” kafedrasida mudiri, fizika-matematika fanlari nomzodi;
Ro‘ziyev R.A.	- NavDPI “Informatika o‘qitish metodikasi” kafedrasida dotsenti, fizika-matematika fanlari nomzodi.

Fanning ishchi o‘quv dasturi “Informatika o‘qitish metodikasi” kafedrasining 201__-yil _____dagi __-son yig‘ilishida muhokamadan o‘tgan va fakultet kengashida muhokama qilish uchun tavsiya etilgan.

Kafedra mudiri:

f.-m.f.n. G‘.R.Yodgorov

Fanning ishchi o‘quv dasturi “Fizika-matematika” fakulteti Kengashida muhokama qilinib, foydalanishga tavsiya etilgan (201__-yil _____dagi __-son bayonnomasi).

Fakultet kengashi raisi:

t.f.n., dots.I.R.Kamolov

Navoiy davlat pedagogika institutining 2019-yil _____-avgustdagi 1-sonli Ilmiy uslubiy kengashida muhokama qilinib tasdiqlangan.

Kelishildi: O‘quv-uslubiy boshqarma boshlig‘i:

Xolmirzayev N.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

I. O‘quv fanning dolzarbligi va Oliy kasbiy ta’limdagi o‘rni

Ta’lim sohasidagi tub islohatlarning asosiy maqsadi jahon andozalari asosida bilimlar berish va raqobatdosh kadrlar tayyorlashdir. Shuning uchun ta’lim tizimidagi 5110700 – Informatika o‘qitish metodikasi yo‘nalishida o‘qitiladigan fanlar ham zamonaviy fanlardan hisoblanadi. Ushbu namunaviy dastur bugungi kunning zamonaviy bilimlari bilan yangilangan va qayta ishlangan dastur bo‘lib, unda fanning nazariy va amaliy jihatlariga alohida e’tibor qaratilgan. Mazkur fan dasturi bakalavriat yo‘nalishi: 5110700 – Informatika o‘qitish metodikasi ta’lim yo‘nalishida tahsil olayotgan talabalarning o‘zlashtirishi lozim bo‘lgan bilimlari va unga qo‘yiladigan talablar asosida tuzilgan bo‘lib, bo‘lajak fan o‘qituvchisi egallashi kerak bo‘lgan bilimlar va ko‘nikmalar majmuini o‘z ichiga oladi:

- algoritmik tillar, algoritm va uning xossalari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari;

- steklar, navbatlar, daraxtlar, algoritmlarning tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, saralashning Sheyker, Shella usullari, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, kattaliklar va ularning tiplari, buruqlar tizimi va operatorlari, massivlar, chiziqli, tarmoqlanuvchi va takrorlanuvchi operatorlar, funksiya va protseduralar, fayllar bilan ishlash, tilning grafik imkoniyatlari;

“Algoritmlar” fani insonlarda axborot muhitida ma’lum bir dunyoqarashni shakllantirishga xizmat qilishi bilan bir qatorda, uning axboriy madaniyatni egallashida asosiy rol o‘ynaydi. Bugungi “Axborot” asrida yoshlarning kompyuter savodxonligini oshiribgina qolmay, balki ma’lumotlar bazalari bilan ishlash imkoniyatlarini oshiradi. Umumiy o‘rta ta’lim maktablari, akademik litsey va kasb–hunar kollejlarda “Informatika” mutahassislaridagi fanlarni o‘qitish uchun kadrlarni tayyorlab beradi.

II. O‘quv fanining maqsadi va vazifasi

Fanni o‘qitishdan maqsad - bo‘lajak informatika o‘qituvchisining kasbiy sohasida egallashi lozim bo‘lgan bilimlar va amalda qo‘llash uchun ko‘nikma va malakalarni shakllantirish va rivojlantirishdan iborat. Ushbu dasturda har bir kasb egasi uning faoliyat ko‘rsatish turidan qat’iy nazar egallashi kerak bo‘lgan tayanch nazariy va amaliy ma’lumotlarni o‘z ichiga oladi.

Fanning o‘qitishning vazifasi - algoritm tushunchasi va uning xossalari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar bilan tanishtirish;

- algoritmik tillarning asosiy tushunchalari: steklar, navbatlar, daraxtlar, algoritmlar tahlili kabilar haqida ma’lumotlar berish.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

- saralash algoritmlari va ulardan foydalanish usullari haqida tasavvurlar hosil qilish borasidagi bilimlar va amalda qo‘llash uchun ko‘nikma va makalalarni shakllantirish va rivojlantirishdan iborat.

Fan bo‘yicha talabalarning bilimi, ko‘nikma va malakalariga qo‘yiladigan talablar

Fan bo‘yicha talabalarning bilim, ko‘nikma va malakalariga quyidagi talablar qo‘yiladi: algoritmlar o‘quv fanini o‘zlashtirish jarayonida amalga oshiriladigan masalalar doirasida bakalavr:

- algoritm va uning xossalari, algoritmik tillar, steklar, navbatlar, daraxtlar, algoritmlar tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, Sheyker usulida saralash, Shell usulida saralash, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari, chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlarga doir *to‘g‘risida tasavvurga ega bo‘lishi*;

- algoritmlar, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar, steklar, navbatlar, daraxtlar, algoritmlar tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, Sheyker usulida saralash, Shell usulida saralash, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari, chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlarni *bilishi va ulardan foydalana olishi*;

-algoritmik tillarning asosiy tushunchalaridan foydalanib sodda va tadbiiy masalalar yechish *ko‘nikmalariga ega bo‘lishi lozim*.

Fanning o‘quv rejadagi boshqa fanlar bilan o‘zaro bog‘liqligi, uslubiy jihatdan uzviyligi va ketma-ketligi

“Algoritmlar” fani asosiy kasbiy fanlardan biri hisoblanib, 1, 2-semestrlarda o‘qitiladi. Dasturni amalga oshirish o‘quv rejasidagi rejalashtirilgan Matematika, Fizika, Informatika fanlaridan olingan nazariy va amaliy bilimlarga tayanadi.

Fanning ta’limdagi o‘rni

Algoritmlar fanidan olingan bilim, ko‘nikma va malakalar umumiy o‘rta ta’lim maktablari, akademik litsey va kasb-hunar kollejlariidagi kasbiy faoliyatida asosiy o‘rin tutib, yoshlarga bilim berishda va mutaxassis tayyorlashda muhim ahamiyat kasb etadi. Umumiy o‘rta ta’lim maktablari, akademik litsey va kasb-hunar kollejlariida “Informatika” ta’limi yo‘nalishidagi fanlarni o‘qitish uchun kadrlarni tayyorlashda muhim ahamiyatga ega.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Fanni o‘qitishda foydalaniladigan zamonaviy axborot va pedagogik texnologiyalar

Fanning o‘qitilishida yangi pedagogik va axborot texnologiyalaridan unumli foydalanish ko‘zda tutiladi. Nazariy ma‘lumotlar amaliyot va ko‘rgazmalilik bilan mustahkamlangandagina, talabalar chuqur bilimga ega bo‘lishi mumkin. Shuning uchun dasturda amaliy va laboratoriya ishlariga katta e‘tibor qaratilgan.

Talabalarning «Algoritmlar» fanini o‘zlashtirishlari uchun o‘qitishning ilg‘or va zamonaviy usullaridan foydalanish, yangi axborot-kommunikatsiya va pedagogik texnologiyalarni tadbiq qilish muhim ahamiyatga ega.

Fanni o‘qitishda zamonaviy pedagogik texnologiyalarni qo‘llashda ta‘lim jarayonini optimallashtirish uchun omil bo‘ladigan pedagogik texnologiyalardan “Fikrlar hujumi”, “Klaster”, “Aqliy hujum”, “Blis-so‘rov”, “Blis-o‘yin”, “Yelpig‘ich”, kichik guruhlar musobaqalari, guruhli fikrlash va boshqalardan foydalaniladi.

Fanni o‘zlashtirishda darslik, o‘quv va uslubiy qo‘llanmalar, elektron materiallar, tarqatma materiallar, virtual kartochkalardan foydalaniladi. Nazariy ma‘lumotlar amaliyot va ko‘rgazmalilik bilan mustahkamlashi kerak. Amaliy va laboratoriya ishlari kompyuter yordamida o‘tkaziladi.

Shaxsga yo‘naltirilgan ta‘lim. Bu ta‘lim o‘z mohiyatiga ko‘ra ta‘lim jarayonining barcha ishtirokchilarini to‘laqonli rivojlanishlarini ko‘zda tutadi. Bu esa ta‘limni loyihalashtirilayotganda, albatta, ma‘lum bir ta‘lim oluvchining shaxsini emas, avvalo, kelgusidagi mutaxassislik faoliyati bilan bog‘liq o‘qish maqsadlaridan kelib chiqqan holda yondoshilishni nazarda tutadi.

Tizimli yondashuv. Ta‘lim texnologiyasi tizimning barcha belgilarini o‘zida mujassam etmog‘i lozim: jarayonning mantiqiyliqi, uning barcha bo‘g‘inlarini o‘zaro bog‘langanligi, yaxlitligi.

Faoliyatga yo‘naltirilgan yondashuv. Shaxsning jarayonli sifatlarini shakllantirishga, ta‘lim oluvchining faoliyatni aktivlashtirish va intensivlashtirish, o‘quv jarayonida uning barcha qobiliyati va imkoniyatlari, tashabbuskorligini ochishga yo‘naltirilgan ta‘limni ifodalaydi.

Dialogik yondoshuv. Bu yondashuv o‘quv munosabatlarini yaratish zaruriyatini bildiradi. Uning natijasida shaxsning o‘z-o‘zini faollashtirishi va o‘z-o‘zini ko‘rsata olishi kabi ijodiy faoliyati kuchayadi.

Hamkorlikdagi ta‘limni tashkil etish. Demokratik, tenglik, ta‘lim beruvchi va ta‘lim oluvchi faoliyat mazmunini shakllantirishda va erishilgan natijalarni baholashda birgalikda ishlashni joriy etishga e‘tiborni qaratish zarurligini bildiradi.

Muammoli ta‘lim. Ta‘lim mazmunini muammoli tarzda taqdim qilish orqali ta‘lim oluvchi faoliyatini faollashtirish usullaridan biri. Bunda ilmiy bilimni ob‘yektiv qarama-qarshiligi va uni hal etish usullarini, dialektik mushohadani shakllantirish va rivojlantirishni, amaliy faoliyatga ularni ijodiy tarzda qo‘llashni mustaqil ijodiy faoliyati ta‘minlanadi.

Axborotni taqdim qilishning zamonaviy vositalari va usullarini qo‘llash - yangi kompyuter va axborot texnologiyalarini o‘quv jarayoniga qo‘llash.

O‘qitishning usullari va texnikasi. Amaliy mashg‘ulot (kirish, mavzuga oid, vizuallashtirish), muammoli ta‘lim, keys-stadi, pinbord, paradoks va loyihalash usullari, amaliy ishlar.

O‘qitishni tashkil etish shakllari: dialog, muloqot hamkorlik va o‘zaro o‘rganishga asoslangan frontal, kollektiv va guruh.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

O‘qitish vositalari: o‘qitishning an’anaviy shakllari (darslik, ma’ruza matni) bilan bir qatorda – kompyuter va axborot texnologiyalari.

Kommunikatsiya usullari: talabalar bilan operativ teskari aloqaga asoslangan bevosita o‘zaro munosabatlar.

Teskari aloqa usullari va vositalari: kuzatish, blits-so‘rov, oraliq va joriy va yakunlovchi nazorat natijalarini tahlili asosida o‘qitish diagnostikasi.

Boshqarish usullari va vositalari: o‘quv mashg‘uloti bosqichlarini belgilab beruvchi texnologik karta ko‘rinishidagi o‘quv mashg‘ulotlarini rejalashtirish, qo‘yilgan maqsadga erishishda o‘qituvchi va talabaning birgalikdagi harakati, nafaqat auditoriya mashg‘ulotlari, balki auditoriyadan tashqari mustaqil ishlarning nazorati.

Monitoring va baholash: o‘quv mashg‘ulotida ham butun kurs davomida ham o‘qitishning natijalarini rejali tarzda kuzatib borish. Kurs oxirida test topshiriqlari yoki yozma ish variantlari yordamida talabalarning bilimlari baholanadi. “Algoritmlar” fanini o‘qitish jarayonida kompyuter texnologiyasidan, “Paskal” dasturidan foydalaniladi. Ayrim mavzular bo‘yicha talabalar bilimni baholash test asosida va kompyuter yordamida bajariladi.

“Algoritmlar” fanidan mashg‘ulotlarning mavzular va soatlar bo‘yicha taqsimlanishi:

T/r	Mavzular nomi	Jami soat	Jami auditoriya soati	Ma’ruza	Amaliy mashg‘ulot	Laboratoriya mashg‘uloti	Mustaqil ta’lim
I SEMESTR							
1.	Algoritmlar tushunchasi va mohiyati	36	20	6	8	6	16
2.	Pascal dasturlash tilining imkoniyatlari	106	56	20	18	18	50
JAMI:		142	76	26	26	24	66
II SEMESTR							
3.	Algoritmlar tushunchasi va mohiyati	70	38	12	14	12	32
Jami:		70	38	12	14	12	32
JAMI:		212	114	38	40	36	98

Asosiy qism

Fanning uslubiy jihatdan uzviy ketma-ketligi Asosiy nazariy qism (ma’ruza mashg‘ulotlari) I SEMESTR

1-Modul. Algoritmlar tushunchasi va mohiyati.

1-mavzu. Algoritm tushunchasi va ulardan foydalanish.

Algoritm tushunchasi va ular haqida ma’lumotlar. Algoritmlar tarixi. Algoritm xossalari, turlari va uning berilish usullari.

Qo‘llaniladigan ta’lim texnologiyalari: ma’ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

2-mavzu. Algoritmlar samaradorligini baholash.

Xotiraviy samara, vaqt samarasi. Algoritmlarning murakkablik darajasi. Algotimlarning taqqoslash usullari.

Qo‘llaniladigan ta‘lim texnologiyalari: ma‘ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

3-mavzu. Algoritmlar tahlili.

Algoritmlarning baholash va ularning tahlili. Kiruvchi berilganlar sinfi. Xotira bo‘yicha murakkablik. Tahlil va maqsad.

Qo‘llaniladigan ta‘lim texnologiyalari: dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q11, I1, I2, I3, I4, I5.

4-mavzu. Algoritmlarni ishlab chiqish metodlari.

Algoritmlarni ishlab chiqishning o‘sga xos jihatlari. Strukturaviy algoritmlar, Prosedurali algoritmlar metodi, Algoritmlar konstruktsiyasi. Xotiraviy samara, vaqt samarasi. Algoritmlarning murakkablik darajasi. Algotimlarning taqqoslash usullari.

Algoritmlar ekvivalent qayta ishlash. Toraytiruvchi o‘zgartirishlar. Formal usulni matematikaga bog‘liq bo‘lmagan muammoga qo‘llash.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q12, I1, I2, I3, I4.

2-Modul. Paskal dasturlash tilining imkoniyatlari.

5-mavzu. Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.

Paskal tili dasturlash tilining alifbosi, Pascal dasturida asosiy fayllar va funksional tugmalar, O‘zgarmas va o‘zgaruvchi miqdorlar, Standart funksiyalar va algebraik ifodalar, O‘zlashtirish va ma‘lumotlarni ekranga chiqarish operatorlari, Ekran bilan ishlash oreratorlari.

Qo‘llaniladigan ta‘lim texnologiyalari: ma‘ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

6-mavzu. Chiziqli, tarmoqlanuvchi va takrorlanuvchi jarayonlarni dasturlash.

Pascalda chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar haqida ma‘lumotlar, qiymat berish, tarmoqlanuvchi operatorlar, shartli va shartsiz o‘tish operatorlari, siklik operatorlar.

Qo‘llaniladigan ta‘lim texnologiyalari: dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, I1, I2, I3, I4.

7-mavzu. Paskalda massivlar

Massivlar haqida tushuncha, bir o‘lchovli va ikki o‘lchovli massivlar, dinamik massivlar.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q11, I1, I2, I3, I4, I5.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

8-mavzu. Paskalda satriy kattaliklar bilan ishlash.

Satriy kattaliklar, satriy kattaliklar bilan ishlovchi funksiya va proseduralar, satriy kattaliklardan foydalanishning o‘ziga xos xususiyatlari.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, blis-savol, aqliy hujum, pinbord

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q12, I1, I2, I3, I4.

9-mavzu. Qism dasturlar-funksiyalar.

Qism dastur, qism dasturlar-funksiyalar, qismdastur-funksiya tuzilishi, formal, lokal va haqiqiy parametrlar, dasturda qism dasturlardan foydalanish.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, aqliy hujum, pinbord, blis-o‘yin.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q12, I1, I2, I3, I4.

10-mavzu. Qism dasturlar-proseduralar.

Qism dastur, qism dasturlar-proseduralar, qismdastur-proseduralar tuzilishi, formal, lokal va haqiqiy parametrlar, dasturda qism dasturlardan foydalanish.

Qo‘llaniladigan ta‘lim texnologiyalari: ma‘ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

11-mavzu. Paskalda fayllar bilan ishlash.

Paskalda fayllar bilan ishlash, tiplashtirilgan va tiplashtirilmagan fayllar, fayllar bilan ishlovchi standart funksiya va proseduralar, fayllarga yozish, fayllardan o‘qish.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, dialogli yondashuv, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q12, I1, I2, I3, I4.

12-mavzu. Grafika moduli va ular bilan ishlash.

Paskal dasturlash tilining grafik imkoniyatlari, grafik adapterlar, graph moduli funksiya va proseduralari.

Qo‘llaniladigan ta‘lim texnologiyalari: ma‘ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

13-mavzu. Paskal tilida sodda shakllar chizish.

Paskalda sodda shakllar chizuvchi proseduralar, sodda shakllar chizishning o‘ziga xos xususiyatlari.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, dialogli yondashuv, aqliy hujum, pinbord, sinkveyn.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q12, I1, I2, I3, I4.

II SEMESTR

1-mavzu. Tanlash va joylashtirish turkumidagi saralash algoritmlari.

Saralash tushunchasi. Saralash algoritmlari. Tanlash va joylashtirish usulida saralash, o‘sib borish va kamayish tartibida saralash, qo‘shish usulida saralash, Joyida abstrakt qo‘shib saralash, yuqoridan pastga qo‘shib saralash.

Qo‘llaniladigan ta‘lim texnologiyalari: dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord, blis-o‘yin.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q11, I1, I2, I3, I4, I5.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

2-mavzu. Almashish usulida saralash, saralashning Sheyker usuli.

Almashish usuli mohiyati, saralashning Sheyker, pufakcha va piramida usullari.

Qo‘llaniladigan ta‘lim texnologiyalari: klaster, dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q12, I1, I2, I3, I4.

3-mavzu. Rekursiya va iteratsiya.

Rekursiya tushunchasi. Funksiya va rekursiv funksiyalar. Rekursiv funksiyalar turlari. Oddiy va vositali rekursiya. Matematikada rekursiyning qo‘llanilishi. Fibonachchi funksiyasi.

Qo‘llaniladigan ta‘lim texnologiyalari: ma‘ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

4-mavzu. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv.

Qidiruv masalasi, qidiruv usullari. Yozuvlarni oddiy ko‘rib chiqish usuli, Ketma-ketlik usulida qidiruv, Binar qidiruv, Interpolyatsiya usulida qidiruv, Binar daraxt va Fibonachchi qidiruvlar, Muvozanatlashgan (Balansirlangan) daraxt bo‘yicha qidiruv, Bor usulida qidiruv, h-Hashlash usulida qidiruv, Interval bo‘yicha izlash.

Qo‘llaniladigan ta‘lim texnologiyalari: ma‘ruzali chiqishlar, dialogli yondashuv, taqdimot usuli, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A6, Q1, Q2, Q3, Q6, Q10, Q11, Q12, I1, I2, I3.

5-mavzu. Rabin-Karp algoritmi.

Rabin-Karp algoritmi haqida. Qidiruv algoritmlari. Rabin-Karp algoritmi xesh-funksiya. Algoritm ahamiyati.

Qo‘llaniladigan ta‘lim texnologiyalari: dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q11, I1, I2, I3, I4, I5.

6-mavzu. Graflar bilan ishlovchi sodda algoritmlar.

Graflar nazariyasi, Graflarni tipik qo‘llanilishi, Graflar terminologiyasi, Qism graf, oriyentirlangan va oriyentirlanmagan graflar, Graflar izomorfizmi.

Qo‘llaniladigan ta‘lim texnologiyalari: dialogli yondashuv, muammoli ta‘lim, aqliy hujum, pinbord.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q11, I1, I2, I3, I4, I5.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

“Algoritmlar” fani bo‘yicha ma’ruza mashg‘ulotlarining kalendar tematik rejasi

№	Ma’ruza mavzulari	Soati
I SEMESTR		
1-Modul. Algoritmlar tushunchasi va mohiyati.		
1.	Algoritm tushunchasi va ulardan foydalanish.	2
2.	Algoritmlar samaradorligini baholash.	2
3.	Algoritmlar tahlili.	2
4.	Algoritmlarni ishlab chiqish metodlari.	2
2-Modul. Paskal dasturlash tilining imkoniyatlari.		
5.	Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.	2
6.	Chiziqli, tarmoqlanuvchi va takrorlanuvchi jarayonlarni dasturlash.	2
7.	Paskal dasturlash tilida massivlar bilan ishlash.	2
8.	Paskalda satriy kattaliklar bilan ishlash.	2
9.	Qism dasturlar-funksiyalar.	2
10.	Qism dasturlar-protseduralar.	2
11.	Paskalda fayllar bilan ishlash.	2
12.	Grafika moduli va ular bilan ishlash.	2
13.	Paskal tilida sodda shakllar chizish.	2
JAMI:		26

№	Ma’ruza mavzulari	Soati
II SEMESTR		
1-modul. Algoritmlar tushunchasi va mohiyati.		
1.	Tyuring mashinasi va Markovning normal algoritmlari.	2
2.	Tanlash va joylashtirish turkumidagi saralash algoritmlari. Almashish usulida saralash, saralashning Sheyker usuli.	2
3.	Rekursiya va iteratsiya.	2
4.	Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv.	2
5.	Rabin-Karp algoritmi.	2
6.	Graflar bilan ishlovchi sodda algoritmlar.	2
JAMI:		12

Amaliy mashg‘ulotlar bo‘yicha tavsiya etiladigan mavzulari

Fan bo‘yicha rejalashtirilgan amaliy mashg‘ulotlar davomida nazariy bilimlar mustahkamlanadi. Amaliy mashg‘ulotlarda tadbiqlar keng qo‘llanilayotgan asosiy matematik usullarni o‘rgatishga e’tiborni kuchaytirish lozim. Ma’lum sababalarga ko‘ra ma’ruzaga kirmagan va murakkab bo‘lgan tushunchalarni amaliy mashg‘ulotlarda ko‘rib o‘tish maqsadga muvofiqdir.

Amaliy mashg‘ulotlarni o‘tkazishda quyidagi didaktik taomiyillarga amal qilinadi:

- amaliy mashg‘ulotlarining maqsadini aniq belgilab olish;

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

- o‘qituvchining innovatsion pedagogik faoliyati bo‘yicha bilimlarni chuqurlashtirish imkoniyatlariga talabalarda qiziqish uyg‘otish;
- talabada natijani mustaqil ravishda qo‘lga kiritish imkoniyatini ta‘minlash;
- talabani nazariy-metodik jihatdan tayyorlash;
- amaliy mashg‘ulotlarda nafaqat aniq mavzu bo‘yicha bilimlarni yakunlash, balki talabalarni tarbiyalash manbai hamdir.

Amaliy mashg‘ulotlarda talabalar muayyan masala bo‘yicha mavjud bo‘lgan yoki mustaqil tarzda kichik ishchi guruhlarida yordamida hosil qilingan algoritmlarni muhokama qiladilar. Mazkur mavzularga oid test masalalar tuzib, ular asosida tuzilgan dasturlar majmuasini tuzadilar va kompyuterda olingan natijalarni birgalikda tahlil qiladilar.

IV. Amaliy mashg‘ulotlar bo‘yicha ko‘rsatma va tavsiyalar

Amaliy mashg‘ulotlar bo‘yicha quyidagi mavzular tavsiya etiladi:

Fan bo‘yicha rejalashtirilgan amaliy mashg‘ulotlar davomida nazariy bilimlar mustahkamlanadi. Amaliy mashg‘ulotlarda tadbirlar keng qo‘llanilayotgan asosiy matematik usullarni o‘rgatishga e‘tiborni kuchaytirish lozim. Ma‘lum sabablarga ko‘ra ma‘ruzaga kirmagan va murakkab bo‘lgan tushunchalarni amaliy mashg‘ulotlarda ko‘rib o‘tish maqsadga muvofiqdir.

Amaliy mashg‘ulotlarni o‘tkazishda quyidagi didaktik tamoyillarga amal qilinadi:

- amaliy mashg‘ulotlarining maqsadini aniq belgilab olish;
- o‘qituvchining innovatsion pedagogik faoliyati bo‘yicha bilimlarni chuqurlashtirish imkoniyatlariga talabalarda qiziqish uyg‘otish;
- talabada natijani mustaqil ravishda qo‘lga kiritish imkoniyatini ta‘minlash;
- talabani nazariy-metodik jihatdan tayyorlash.

Amaliy mashg‘ulotlarda nafaqat aniq mavzu bo‘yicha bilimlarni yakunlash, balki talabalarni tarbiyalash manbai hamdir. Amaliy mashg‘ulotlarda talabalar muayyan masala bo‘yicha mavjud bo‘lgan yoki mustaqil tarzda kichik ishchi guruhlarida yordamida hosil qilingan algoritmlarni muhokama qiladilar. Mazkur mavzularga oid test masalalar tuzib, ular asosida tuzilgan dasturlar majmuasini tuzadilar va kompyuterda olingan natijalarni birgalikda tahlil qiladilar. Amaliy mashg‘ulotlarning taxminiy tavsiya etiladigan mavzulari:

1. Algoritm tushunchasi va ulardan foydalanish.
2. Algoritm samaradorligini baholash.
3. Tanlash va joylashtirish turkumidagi murakkablikka ega saralash algoritmlari.
4. Almashish usulida saralash, saralashning Sheyker usuli.
5. Algoritm tahlili.
6. Algoritmni ishlab chiqish metodlari.
7. Rekursiya va rekursiv funksiyalar.
8. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv.
9. Rabin-Karp algoritmi.
10. Graflar bilan ishlovchi sodda algoritmlar.
11. Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.
12. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar.
13. Paskalda massivlar.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

14. Paskalda satriy kattaliklar bilan ishlash.
15. Qism dasturlar-funksiyalar.
16. Qism dasturlar-proseduralar.
17. Qism dasturlar-funksiya va proseduralarga doir dasturlar tuzish.
18. Paskalda fayllar bilan ishlash.
19. Grafika moduli va ular bilan ishlash.
20. Paskal tilida sodda shakllar chizish.

1-Modul. Algoritmlar tushunchasi va mohiyati

Algoritm tushunchasi va ulardan foydalanish. Algoritmlar samaradorligini baholash. Algoritmlar tahlili. Algoritmlarni ishlab chiqish metodlari. Algoritmning asosiy tiplari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar va ularning blok-sxemalari. Tyuring mashinasi. Markovning normal algoritmlari va ular asosida masalalar yechish. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari. Almashish usulida saralash, saralashning Sheyker usuli. Rekursiya va rekursiv funksiyalar. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv. Rabin-Karp algoritmi. Graflar bilan ishlovchi sodda algoritmlar.

Qo‘llaniladigan ta’lim texnologiyalari: dialogik yondashuv, muammoli ta’lim, aqliy hujum, pinbord, klaster, blis-o‘yin, fikrlar hujumi.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q8, Q10, Q11, Q12, E1, E2, E3, E4, E5, E6.

2-Modul. Paskal dasturlash tilining imkoniyatlari

Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar. Paskalda massivlar. Paskalda satriy kattaliklar bilan ishlash. Qism dasturlar-funksiyalar va proseduralar. Qism dasturlar-funksiya va proseduralarga doir dasturlar tuzish. Paskalda fayllar bilan ishlash. Grafika moduli va ular bilan ishlash. Paskal tilida sodda shakllar chizish.

Qo‘llaniladigan ta’lim texnologiyalari: dialogik yondashuv, muammoli ta’lim, aqliy hujum, pinbord, blis-o‘yin, fikrlar hujumi, blis-savol.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, E1, E2, E3, E4.

“Algoritmlar” fani bo‘yicha amaliy mashg‘ulotlarning kalendar tematik rejasi:

№	Amaliy mashg‘ulot mavzulari	Soati
I semestr		
1-Modul. Algoritmlar tushunchasi va mohiyati.		
1.	Algoritm tushunchasi va ulardan foydalanish.	2
2.	Algoritmlar samaradorligini baholash.	2
3.	Algoritmlar tahlili. Algoritmlarni ishlab chiqish metodlari.	2
4.	Algoritmning asosiy tiplari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar va ularning blok-sxemalari.	2
II modul. Pascal dasturlash tili asoslari.		
5.	Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.	2
6.	Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar.	2

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

7.	Paskalda massivlar.	2
8.	Paskalda satriy kattaliklar bilan ishlash.	2
9.	Qism dasturlar-funksiyalar va protseduralar..	2
10.	Qism dasturlar-funksiya va proseduralarga doir dasturlar tuzish.	2
11.	Paskalda fayllar bilan ishlash.	2
12.	Grafika moduli va ular bilan ishlash.	2
13.	Paskal tilida sodda shakllar chizish.	2
Jami:		26

№	Amaliy mashg‘ulot mavzulari	Soati
II semestr		
1-Modul. Algoritm tushunchasi va mohiyati.		
1.	Tyuring mashinasi. Markovning normal algoritmlari va ular asosida masalalar yechish.	2
2.	Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.	2
3.	Almashish usulida saralash, saralashning Sheyker usuli. Rekursiya va rekursiv funksiyalar.	2
4.	Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv.	2
6.	Rabin-Karp algoritmi.	2
7.	Graflar bilan ishlovchi sodda algoritmlar.	2
Jami:		14

Amaliy mashg‘ulotlarni tashkil etish bo‘yicha kafedra professor-o‘qituvchilari tomonidan ko‘rsatma va tavsiyalar ishlab chiqiladi. Unda talabalar asosiy ma’ruza mavzulari bo‘yicha olgan bilim va ko‘nikmalarini amaliy masalalarga dasturlar tuzish orqali bilimlarini yanada boyitadilar. Shuningdek, darslik va o‘quv qo‘llanmalar asosida talabalar bilimlarini mustahkamlashga erishish, tarqatma materiallardan foydalanish, ilmiy maqolalar va tezislarni chop etish orqali talabalar bilimini oshirish, tanlab olimgan masalalar uchun dasturlar tuzish, mavzular bo‘yicha ko‘rgazmali qurollar tayyorlash va boshqalar tavsiya etiladi.

Laboratoriya mashg‘ulotlari bo‘yicha tavsiya etiladigan mavzulari

Laboratoriya mashg‘ulotlarida talabalar kompyuter yordamida grafik ma’lumotlarni hosil qilish va amalda uning natijalarini ko‘rib, ularni tahlil qiladi va xulosalar chiqaradilar. Laboratoriya mashg‘ulotlari uchun tavsiya etiladigan mavzular:

1. Algoritm tushunchasi va ulardan foydalanish.
2. Algoritm samaradorligini baholash.
3. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.
4. Almashish usulida saralash, saralashning Sheyker usuli.
5. Algoritm tahlili.
6. Algoritmni ishlab chiqish metodlari.
7. Rekursiya va rekursiv funksiyalar.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

8. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv
9. Rabin-Karp algoritmi.
10. Graflar bilan ishlovchi sodda algoritmlar.
11. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.
12. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar.
13. Paskalda massivlar.
14. Paskalda satriy kattaliklar bilan ishlash.
15. Qism dasturlar-funksiyalar.
16. Qism dasturlar-protseduralar.
17. Paskalda fayllar bilan ishlash.
18. Grafika moduli va ular bilan ishlash.
19. Paskal tilida sodda shakllar chizish.

1-Modul. Algoritmlar tushunchasi va mohiyati

Algoritm tushunchasi va ulardan foydalanish. Algoritmlar samaradorligini baholash. Tyuring mashinasi. Markovning normal algoritmlari va ular asosida masalalar yechish. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari. Almashish usulida saralash, saralashning Sheyker usuli. Algoritmlar tahlili. Algoritmlarni ishlab chiqish metodlari. Rekursiya va rekursiv funksiyalar. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv. Rabin-Karp algoritmi. Graflar bilan ishlovchi sodda algoritmlar.

Qo‘llaniladigan ta’lim texnologiyalari: dialogik yondashuv, muammoli ta’lim, aqliy hujum, pinbord, keys-stadi, fikrlar hujumi.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q7, Q8, Q10, Q11, Q12, I1, I2, I3, I4, I5, I6.

2-Modul. Paskal dasturlash tilining imkoniyatlari

Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar. Paskalda massivlar. Paskalda satriy kattaliklar bilan ishlash. Qism dasturlar-funksiyalar. Qism dasturlar-protseduralar. Qism dasturlar-funksiya va proseduralarga doir dasturlar tuzish. Paskalda fayllar bilan ishlash. Grafika moduli va ular bilan ishlash. Paskal tilida sodda shakllar chizish.

Qo‘llaniladigan ta’lim texnologiyalari: dialogik yondashuv, muammoli ta’lim, aqliy hujum, pinbord, keys-stadi, fikrlar hujumi.

Adabiyotlar: A1, A2, A3, A4, A5, A6, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, I1, I2, I3, I4.

“Algoritmlar” fani bo‘yicha laboratoriya mashg‘ulotlarining kalendar tematik rejasi

№	Laboratoriya mashg‘ulotlarining mavzulari	Soati
I SEMESTR		
I modul. Algoritmlar tushunchasi va mohiyati.		
1.	Algoritm tushunchasi va ulardan foydalanish.	2
2.	Algoritmlar samaradorligini baholash.	2
3.	Algoritmlar tahlili. Algoritmlarni ishlab chiqish metodlari.	2
II modul. Pascal dasturlash tilining imkoniyatlari.		

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

4.	Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.	2
5.	Chiziqli va tarmoqlanuvchi jarayonlarni dasturlash.	2
6.	Takrorlanuvchi jarayonlarga doir masalalar va ularning hisoblash dasturlarini yaratish.	2
7.	Paskalda massivlar bilan ishlash.	2
8.	Paskalda satriy kattaliklar bilan ishlash.	2
9.	Qism dasturlar-funksiyalar va protseduralarga doir dasturlar tuzish.	2
10.	Paskalda fayllar bilan ishlash.	2
11.	Grafika moduli va ular bilan ishlash.	2
12.	Paskal tilida sodda shakllar chizish.	2
JAMI:		24
II SEMESTR		
№	Laboratoriya mashg‘ulotlarining mavzulari	Soati
I modul. Algoritmlar tushunchasi va mohiyati.		
1.	Tyuring mashinasi. Markovning normal algoritmlari.	2
2.	Tanlash va joylashtirish turkumidagi murakkablikga ega saralash algoritmlari.	2
3.	Almashish usulida saralash, saralashning Sheyker usuli. Rekursiya va rekursiv funksiyalar.	2
4.	Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv.	2
5.	Rabin-Karp algoritmi.	2
6.	Graflar bilan ishlovchi sodda algoritmlar.	2
JAMI:		12

Kurs ishi (loyihasi) tarkibi, ularga qo‘yiladigan talablar

O‘quv rejasida mazkur fandan kurs ishi rejalashtirilmagan.

Mustaqil ishlar mavzulari, mazmuni va ularga ajratilgan soatlar

Talaba mustaqil ta’limni tayyorlashda fanning xususiyatlarini hisobga olgan holda quyidagi shakllardan foydalanish tavsiya etiladi:

- darslik va o‘quv qo‘llanmalar bo‘yicha fan boblari va mavzularini o‘rganish;
- tarqatma materiallar bo‘yicha ma’ruzalar qismini o‘zlashtirish;
- avtomatlashtirilgan o‘rgatuvchi va nazorat qiluvchi dasturlar bilan ishlash;
- maxsus adabiyotlar bo‘yicha fanlar bo‘limlari yoki mavzulari ustida ishlash;
- talabaning o‘quv va ilmiy-tadqiqot ishlarini bajarish bilan bog‘liq bo‘lgan fanlar bo‘limlari va mavzularni chuqur o‘rganish;
- zamonaviy dasturlash muhitidan foydalanishga mustaqil o‘rganishlari;

Tavsiya etilayotgan mustaqil ishlarning mavzulari:

T/r	Mustaqil ta’lim mavzulari	Berilgan topshiriqlar	Bajarish muddati	Hajmi (soatda)
I SEMESTR				
25.	Algoritm tushunchasi va ulardan foydalanish.	Algoritm tushunchasi va ulardan foydalanishni o‘rganish.	1- hafta	2

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

26.	Algoritm va uning xossalari. Algoritm turlari.	Algoritmning xossalari va turlarini o‘rganish.	2- hafta	2
27.	Algoritm nazariyasiga asos solgan olimlar.	Algoritm nazariyasiga asos solgan olimlar haqidagi ma’lumotlarni o‘rganish.	3- hafta	2
28.	Algoritmlar samaradorligini baholash.	Murakkab turdagi algoritmlar va ularning qo‘llanilishiga doir masalalar yechish.	4- hafta	2
29.	Algoritmlar tahlili.	Algoritmlar tahlilini o‘rganish.	4- hafta	2
30.	Algoritmlarni ishlab chiqish metodlari.	Algoritmlarni ishlab chiqish metodlarini o‘rganish.	5- hafta	2
31.	Chiziqli va tarmoqlanuvchi algoritmlar.	Chiziqli, tarmoqlanuvchi jarayonlarni algoritmlashni o‘rganish.	6- hafta	2
32.	Takrorlanuvchi algoritmlar.	Takrorlanuvchi jarayonlarni algoritmlashni o‘rganish.	6- hafta	2
33.	Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari.	Pascal dasturlash tilining kelib chiqishi va qo‘llanishini o‘rganish.	7- hafta	2
34.	Standart funksiyalar bilan ishlash.	Standart funksiyalar bilan ishlashni o‘rganish.	7- hafta	2
35.	Interpretatorlar va kompilyatorlar.	Interpretatorlar va kompilyatorlar haqidagi ma’lumotlarni o‘rganish.	8- hafta	2
36.	Chiziqli va tarmoqlanuvchi dasturlar.	Pascal dasturlash tilidan chiziqli va tarmoqlanuvchi dasturlar tuzish va ularni tahlil qilish.	8, 9- haftalar	4
37.	Takrorlanuvchi dasturlar	a) Dars jadvalini chiqaruvchi. b) Alfavit bo‘yicha saralovchi dastur. v) Test topshiriqlarini bajaruvchi (To‘g‘ri javobni belgilovchi) dasturlar tuzing.	10- hafta	4
38.	Tanlash operatorlari bilan ishlash	Tanlash operatorlari vazifasini o‘rganish.	11- hafta	2
39.	Paskalda bir o‘lchovli va ikki o‘lchovli massivlar bilan ishlash.	Mustaqil ravishda 10 dan dastur tuzing va uning bajarilishini asoslab bering.	11, 12- haftalar	4
40.	Paskalda satriy kattaliklar bilan ishlash.	Sana-vaqt turi bilan ishlashga doir masalalar yechish.	12, 13- haftalar	4
41.	Qism dasturlar-funksiyalar yordamida dasturlar tuzish.	Funksiyalardan foydalanib dastur tuzish.	13, 14- haftalar	4
42.	Qism dasturlar-protseduralar yordamida	Bir necha xil Protседura va funksiyalarning qo‘llanishiga	14- hafta	4

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

	dasturlar tuzish	doir misollar.		
43.	Paskalda fayllar bilan ishlash.	Paskalda fayllar bilan ishlashga doir dasturlar tuzish.	15- hafta	4
44.	Ma'lumotlarni fayllardan o'qish. Ma'lumotlarni fayllarga yozish.	Mustaqil ravishda 10 dan dastur tuzing va uning bajarilishini asoslab bering. Ma'lumotlarni fayllarga yozish bo'yicha dasturlar tuzish.	16- hafta	4
45.	Grafika moduli va ular bilan ishlash.	Pascal dasturlash tilida grafik imkoniyatlar va ulardan foydalanish.	17- hafta	2
46.	Paskal tilida sodda geometrik shakllar chizish.	Figuralarni harakatga keltiruvchi dasturlar tuzish.	18- hafta	2
47.	Paskalda kichik hajmdagi amaliy dasturlar yaratish.	Paskalda kichik hajmdagi amaliy dasturlar yaratish.	18- hafta	4
48.	Paskalda amaliy dasturlar yaratish.	Paskalda amaliy dasturlar yaratish.	19- hafta	2
	JAMI:			66

T/r	Mustaqil ta'lim mavzulari	Berilgan topshiriqlar	Bajarish muddati	Hajmi (soatda)
II SEMESTR				
1.	Saralash usullari.	Saralash usullarini o'rganish.	1- hafta	2
2.	Tanlash va joylashtirish turkumidagi saralash algoritmlari.	Tanlash va joylashtirish turkumidagi murakkablikga ega saralash algoritmlarini o'rganish.	2-hafta	2
3.	Qo'shish usulida saralash.	Qo'shish usulida saralashga doir dastur tuzish.	3-hafta	2
4.	Almashish usulida saralash, saralashning Sheyker usuli.	Almashish usulida saralash, saralashning Sheyker usullaridan foydalanib masalalar yechish.	4,5- haftalar	2
5.	Piramida va pufakcha usulida saralash.	Piramida va pufakcha saralash usulidan foydalanib dastur tuzish.	6,7- haftalar	2
6.	Algoritm tushunchasini formallashtirish: Tyuring mashinasi.	Algoritm tushunchasini formallashtirish: Tyuring mashinasini ishlash algoritmini o'rganish.	8-hafta	2
7.	Eyler sikllari va zanjirlari. Eyler teoremasi.	Eyler sikllari va zanjirlari. Eyler teoremasini o'rganish.	9,10- haftalar	2
8.	Evklid algoritmi.	Evklid algoritmi.	11-hafta	2
9.	Rekursiya va rekursiv	Rekursiya va rekursiv	12-hafta	2

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

	funksiyalar	funksiyalarni o‘rganish.		
10.	Qidiruv usullari: binar qidiruv.	Qidiruv usullari: binar qidiruvdan dasturlar tuzish.	13-hafta	2
11.	Fibonachchi qidiruv.	Fibonachchi qidiruvga doir dasturlar tuzish.	14- hafta	2
12.	Binar daraxt bo‘yicha qidiruv.	Binar daraxt bo‘yicha qidiruv.	15-hafta	2
13.	Chiziqli ro‘yhatlar.	Chiziqli ro‘yhatlar haqida nazariy o‘rganish.	16- hafta	2
14.	Tarmoqlar. Daraxtlar.	Tarmoqlar va daraxtlar haqida o‘rganish..	17-hafta	2
15.	Rabin-Karp algoritmi.	Rabin-Karp algoritmini tuzish.	18- hafta	2
16.	Graflar bilan ishlovchi sodda algoritmlar.	Graflar bilan ishlovchi sodda algoritmlarni o‘rganish.	19-hafta	2
	JAMI:			32

Dasturning informatsion uslubiy ta‘minoti

1. **Jihozlar va uskunalar, moslamalar:** elektron doska-Hitachi, LCD-monitor, elektron ko‘rsatgich (ukazka).
2. **Video–audio uskunalar:** video va audiomagnitofon, mikrofon, kolonkalar.
3. **Kompyuter va mul‘timediali vositalar:** komp’yuter, Dell tipidagi proektor, DVD-diskovod, Web-kamera, video-ko‘z (videoglaz).

II. Baholash turlari va shakllari

II.1. Oraliq nazorat turi

Oliy ta‘lim muassasalarida talabalar bilimini nazorat qilish oraliq va yakuniy nazorat turlarini o‘tkazish orqali amalga oshiriladi.

Oraliq nazorat semestr davomida ishchi fan dasturining tegishli bo‘limi tugagandan keyin talabaning bilim va amaliy ko‘nikmalarini baholash maqsadida o‘quv mashg‘ulotlari davomida o‘tkaziladi.

Oraliq nazorat turi har bir fan bo‘yicha fanning hususiyatidan kelib chiqqan holda 2 martagacha o‘tkazilishi mumkin.

Oraliq nazorat turini o‘tkazish shakli (*yozma, og‘zaki, test va hokazo*) va muddati fanning xususiyati va fanga ajratilgan soatlardan kelib chiqib tegishli kafedra tomonidan belgilanadi.

Oraliq nazorat turining topshiriqlari tegishli kafedra professor-o‘qituvchilari tomonidan ishlab chiqiladi va mazkur kafedra mudiri tomonidan tasdiqlanadi.

Semestr davomida haftasiga 2 akademik soatdan kam bo‘lgan fanlar bo‘yicha oraliq nazorat turi o‘tkazilmaydi.

Talabaning amaliy, seminar, laboratoriya mashg‘ulotlari va mustaqil ta‘lim topshiriqlarini bajarishi, shuningdek uning ushbu mashg‘ulotlardagi faolligi fan o‘kituvchisi tomonidan baholab boriladi. Baholash O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta‘lim vazirligining 2018 yil 9 avgustdagi 19-2018-son buyrug‘ining ya‘ni “Oliy talim muassasalarida talabalar bilimini nazorat qilish va baholash tizimi to‘g‘risida”gi Nizomning 15-bandida nazarda tutilgan mezonlar asosida amalga oshiriladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda faolligi uchun talabani baholash mezonida quyiladigan talablar qo‘yiladi.

Baho		Talabning bilim darajasi va malakasiga talablar
5	A'lo	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda faol ishtirok etadi, berilgan topshiriqlarni mustaqil fikr asosida to‘g‘ri bajaradi, javoblarni izohlab ularning amaliy ahamiyatini anglay oladi, topshiriqlarni bajarishda ijodiy yondoshadi va ijodiy fikrlay oladi, o‘z fikrini to‘la ifodalay oladi, laboratoriya mashg‘ulotlarini o‘z vaqtida bajarib topshiradi, mustaqil ta’lim mavzularini to‘liq o‘zlashtiradi, talaba mustaqil xulosa va qaror qabul qiladi, mustaqil mushohada yuritadi, olgan bilimini amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunda, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda.
4	Yaxshi	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda faol ishtirok etadi, berilgan topshiriq va mashqlarni to‘g‘ri bajaradi, javoblarni izohlay oladi, fikrini mustaqil ifodalay oladi, topshiriq mohiyatini to‘la tushunadi, laboratoriya mashg‘ulotlarini o‘z vaqtida bajarib topshiradi, mustaqil ta’lim mavzularini 70%dan 90%gacha o‘zlashtiradi, talaba mustaqil mushohada yuritadi, olgan bilimini amalda qullay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda.
3	Qoniqarli	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda ishtirok etib topshiriqlarni o‘qituvchi yordamida to‘g‘ri bajaradi, yechimlardan olingan javoblarni mohiyatini tushunadi, masalani yechish jarayonini tushuntira oladi, mustaqil berilgan laboratoriya mashg‘ulotlarini 60% yoki 70% o‘z vaqtida bajaradi, mustaqil ta’limni qisman o‘zlashtiradi, talaba olgan bilimini amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda.
2	Qoniqarsiz	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlardagi topshiriqlarning shartini to‘g‘ri tushunib ularni to‘liq bajara olmaydi, topshiriqlarni ba’zilar to‘g‘risida aniq tasavvurga ega bo‘lmaydi, o‘qituvchi ko‘magida ham topshiriqlarni bajarishga qiynaladi, nazariy va amaliy bilimlarini bog‘lay olmaydi, laboratoriya mashg‘ulotlarini o‘z vaqtida bajarmaydi va topshira olmaydi, mustaqil ta’limni o‘zlashtirmaydi, talaba fan dasturini o‘zlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas deb topilganda.

Talabani oraliq nazorat turi bo‘yicha baholashda, uning o‘quv mashg‘ulotlari davomida olgan baholari inobatga olinadi.

Talabalar bilimni baholash 5 baholik tizimda amalga oshiriladi. Oraliq nazorat turini o‘tkazish va mazkur nazorat turi bo‘yicha talabning bilimni baholash tegishli fan bo‘yicha o‘quv mashg‘ulotlarini olib borgan professor-o‘qituvchi tomonidan amalga oshiriladi.

Talaba tegishli fan bo‘yicha yakuniy nazorat turi o‘tkaziladigan muddatga qadar oraliq nazorat turini topshirgan bo‘lishi shart.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Oraliq nazorat turini topshirmagan, shuningdek ushbu nazorat turi bo‘yicha “2” (qoniqarsiz) baho bilan baholangan talaba yakuniy nazorat turiga kiritilmaydi.

Oraliq (yozma, og‘zaki) nazoratda talaba bilimini baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Savollardagi mavzularning barchasiga asoslangan, ilmiy xatoliklarga yo‘l qo‘yilmagan holda javoblar beradi, mavzu material mohiyatini to‘la tushunib yetgan bo‘ladi, ijodiy fikr yuritadi, mustaqil mushohada qiladi, nazariy bilimlarni amalda qo‘llashga misollar keltira oladi, xulosa va qarorlar qabul qilishda faol bo‘ladi, material bo‘yicha to‘la tasavvurga ega bo‘ladi va talaba ilmiy-uslubiy maqolalar yozishga loyiq bo‘ladi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi shuningdek mashg‘ulotlarda faol ishtirok etagan deb topilganda.
4	Yaxshi	Savollarning barchasiga to‘liq javob beradi, juz‘iy xatoliklarga yo‘l qo‘ymaydi. Material mohiyatni tushunib yetgan bo‘ladi, ijodiy fikr yurita oladi, nazariy bilimlarni amaliy ahamiyatini anglab yetgan bo‘ladi, material bo‘yicha tasavvurga ega bo‘lsa qo‘shimcha adabiyotlardan mustaqil foydalana olish qobiliyatiga hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi yaxshi ishtirokni inobatga olgan deb topilganda
3	Qoniqarli	Savollarga javoblar yozadi, yo‘l qo‘ygan xatolari juz‘iy xatolar bo‘lmaydi, material mohiyatini tushungan bo‘ladi, nazariy bilimlarni amaliy ahamiyatini anglagan bo‘ladi, mavzular bo‘yicha tasavvurga ega bo‘ladi va auditoriya mashg‘ulotlariga to‘liq qatnashgan bo‘ladi hamda fan (mavzu) bo‘yicha tasavvurga ega hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlarda ishtirok etib topshiriqlarni o‘qituvchi yordamida to‘g‘ri bajaradi deb topilganda
2	Qoniqarsiz	Savollarga javob berishga qiynaladi, material mohiyatini tushunmaydi, tasavvuri sayoz bo‘ladi, nazariy bilimlarni amaldagi ahamiyatni anglab yetmaydi, savollarni ko‘pchiligiga javob bera olmaydi va darslarga muntazam qatnashmagan bo‘laydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi topshiriqlarning shartini to‘g‘ri tushunib ularni bajara olmaydi deb topilganda

Oraliq (test) nazoratda talaba bilimini baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Talaba test savollarining 90%dan yuqorisiga to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi shuningdek mashg‘ulotlarda faol ishtirok etagan deb topilganda.
4	Yaxshi	Talaba test savollarining 89,9%dan 70% gacha to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi yaxshi ishtirokni inobatga olgan deb topilganda

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3	Qoniqarli	Talaba test savollarining 69,9%dan 60% gacha to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlarda ishtirok etib topshiriqlarni o‘qituvchi yordamida to‘g‘ri bajaradi deb topilganda
2	Qoniqarsiz	Talaba test savollarining 59,9% dan kamiga to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi topshiriqlarning shartini to‘g‘ri tushunib ularni bajara olmaydi deb topilganda

Fan bo‘yicha oraliq nazorat turlarida talaba olgan baho yakuniy nazorat turiga kirish uchun.

Nizomda ko‘rsatib o‘tilgandek, talaba uzrli sabablarga ko‘ra oraliq nazorat turiga kirmagan taqdirda ushbu talabaga oraliq nazorat turini qayta topshirishga fakultet dekanining farmoyishi asosida ruxsat beriladi.

Talabaga oraliq nazorat turini qayta topshirish uchun berilgan muddat davomida talaba tomonidan qayta topshirishlar soni 2 martadan ko‘p bo‘lmasligi kerak.

Talaba oraliq nazorat turini birinchi marta qayta topshirishdan o‘ta olmagan taqdirda, fakultet dekani tomonidan komissiya tuziladi. Komissiya tarkibi tegishli fan bo‘yicha professor-o‘qituvchi va soha mutaxassisleri orasidan shakllantiriladi.

Ikkinchi marta oraliq nazorat turini o‘tkazish va talabani baholash mazkur komissiya tomonidan amalga oshiriladi.

Berilgan muddat davomida mavjud bo‘lgan qarzdorlikni topshira olmagan talaba bo‘yicha fakultet dekani bildirgi bilan oliy ta‘lim muassasasi rektorini xabardor qiladi va ushbu talaba rektor buyrug‘i asosida kursdan qoldiriladi.

II.2. Yakuniy nazorat turi

Yakuniy nazorat turi semestr yakunida tegishli fan bo‘yicha talabaning nazariy bilim va amaliy ko‘nikmalarini o‘zlashtirish darajasini aniqlash maqsadida o‘tkaziladi.

Yakuniy nazorat turini o‘tkazish shakli tegishli fan bo‘yicha kafedra tomonidan belgilanadi.

Yakuniy nazorat turi oliy ta‘lim muassasasining tegishli fakultet dekani yoki o‘quv-uslubiy bo‘lim tomonidan ishlab chiqiladigan hamda o‘quv ishlari bo‘yicha prorektor tomonidan tasdiqlanadigan Yakuniy nazorat turlarini o‘tkazish jadvalga muvofiq o‘tkaziladi.

Yakuniy(yozma, og‘zaki) nazoratda talaba bilimni baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Savollardagi mavzularning barchasiga asoslangan, ilmiy xatoliklarga yo‘l qo‘yilmagan holda javoblar beradi, mavzu material mohiyatini to‘la tushunib yetgan bo‘ladi, ijodiy fikr yuritadi, mustaqil mushohada qiladi, nazariy bilimlarni amalda qo‘llashga misollar keltira oladi, xulosa va qarorlar qabul qilishda faol bo‘ladi, material bo‘yicha to‘la tasavvurga ega deb topilganda.
4	Yaxshi	Savollarning barchasiga to‘liq javob beradi, juz‘iy xatoliklarga yo‘l qo‘ymaydi. Material mohiyatni tushunib yetgan bo‘ladi, ijodiy fikr yurita oladi, nazariy bilimlarni amaliy ahamiyatini anglab yetgan bo‘ladi, material bo‘yicha tasavvurga ega bo‘lsa qo‘shimcha adabiyotlardan mustaqil foydalana olish qobiliyatiga ega deb topilganda
3	Qoniqarli	Savollarga javoblar yozadi, yo‘l qo‘ygan xatolari juz‘iy xatolar bo‘lmaydi, material mohiyatini tushungan bo‘ladi, nazariy bilimlarni amaliy ahamiyatini anglagan bo‘ladi, mavzular

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

		bo‘yicha tasavvurga ega bo‘ladi va auditoriya mashg‘ulotlariga to‘liq qatnashgan bo‘ladi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda
2	Qoniqarsiz	Savollarga javob berishga qiynaladi, material mohiyatini tushunmaydi, tasavvuri sayoz bo‘ladi, nazariy bilimlarni amaldagi ahamiyatni anglab yetmaydi, savollarni ko‘pchiligiga javob bera olmaydi va darslarga muntazam qatnashmagan bo‘laydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas deb topilganda

Yakuniy(test) nazoratda talaba bilimni baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Talaba test savollarining 90%dan yuqorisiga to‘g‘ri javob berganda. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test material mohiyatini to‘la tushunib yetgan bo‘ladi, ijodiy fikr yuritadi, mustaqil mushohada qiladi, nazariy bilimlarni amalda qo‘llaydi, xulosa va qarorlar qabul qilishda faol bo‘ladi, test material bo‘yicha to‘la tasavvurga ega deb topilganda.
4	Yaxshi	Talaba test savollarining 89,9%dan 70%gachasiga to‘g‘ri javob berishi lozim. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test material mohiyatini tushunib yetgan bo‘ladi, test savollariga javob berishda o‘zining fikr va mulohazalariga tayanadi, nazariy bilimlarni amaliy ahamiyatini anglab yetgan bo‘ladi, test material bo‘yicha yaxshi tasavvurga ega deb topilganda
3	Qoniqarli	Talaba test savollarining 69,9%dan 60%gachasiga to‘g‘ri javob berishi lozim. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test material mohiyatini tushungan bo‘ladi, nazariy bilimlarni amaliy ahamiyatini anglagan bo‘ladi, test material bo‘yicha o‘rtacha tasavvurga ega deb topilganda
2	Qoniqarsiz	Talaba test savollarining 59,9%dan kamiga to‘g‘ri javob berganda. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test savollarga javob berishga qiynaladi, material mohiyatini tushunmaydi, tasavvuri sayoz bo‘ladi, nazariy bilimlarni amaldagi ahamiyatni anglab yetmaydi, test savollarni kattf qismiga javob bera olmaydi deb topilganda

Yakuniy nazorat turini o‘tkazish va mazkur nazorat turi bo‘yicha talabning bilimni baholash o‘quv mashg‘ulotlarini olib bormagan professor-o‘qituvchi tomonidan amalga oshiriladi.

Tegishli fan bo‘yicha o‘quv mashg‘ulotlarini olib borgan professor-o‘qituvchi yakuniy nazorat turini o‘tkazishda ishtirok etishi taqiqlanadi.

Yakuniy nazorat turini o‘tkazishda kelishuv asosida boshqa oliy ta‘lim muassasalarining tegishli fan bo‘yicha professor-o‘qituvchilari jalb qilinishi mumkin.

Oliy ta‘lim muassasasida nazorat turlarini o‘tkazilishi tegishli oliy ta‘lim muassasasining ta‘lim sifatini nazorat qilish bo‘limi tomonidan doimiy ravishda o‘rganib boriladi. Bunda nazorat turlarini o‘tkazilish tartibi buzilganligi aniqlangan hollarda, o‘tkazilgan nazorat turlarining natijalari bekor qilinishi hamda tegishli nazorat turi qaytadan o‘tkazilishi mumkin.

Yakuniy nazorat turiga kirmagan yoki kiritilmagan, shuningdek ushbu nazorat turi bo‘yicha “2” (qoniqarsiz) baho bilan baholangan talaba akademik qarzdor hisoblanadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Nizomning 23 – bandiga asosan, talaba uzrli sabablarga ko‘ra yakuniy nazorat turiga kirmagan taqdirda ushbu talabaga yakuniy nazorat turini qayta topshirishga fakultet dekanining farmoyishi asosida ruxsat beriladi.

Bir kunda 1 tadan ortiq fan bo‘yicha yakuniy nazorat turi o‘tkazilishiga yo‘l qo‘yilmaydi. Yakuniy nazorat turlarini o‘tkazish kamida 2 kun oralig‘ida belgilanishi lozim.

Yuqorida keltirib o‘tganimizdek, bitiruvchi kurs bo‘lmagan talabalar kuzgi semestr natijalari bo‘yicha 3 tagacha fandan (fanlardan) akademik qarzdorligi bo‘lgan hollarda talabaga bir oygacha, bahorgi semestr natijalari bo‘yicha 3 tacha fandan (fanlardan) akademik qarzdorligi bo‘lgan talabaga tegishli fan (fanlar) bo‘yicha oralik va (yoki) yakuniy nazorat turlarini yangi o‘quv yili boshidan qayta topshirish uchun 1 oy muddat beriladi.

Bitiruvchi kurs talabalariga bahorgi semestr natijalari bo‘yicha o‘zlashtirmagan fandan (fanlardan) qayta topshirish uchun yakuniy davlat attestatsiyasi boshlangunga qadar ruxsat beriladi.

Fanlardan akademik qarzdorligi 4 ta va undan ko‘p bo‘lgan talabalarga qayta topshirishga ruxsat berilmaydi va ular oliy ta‘lim muassasasi rektorining buyrug‘i bilan kursdan qoldiriladi.

Talabaga yakuniy nazorat turini qayta topshirish uchun berilgan muddat davomida talaba tomonidan qayta topshirishlar soni 2 martadan ko‘p bo‘lmasligi kerak.

Talaba yakuniy nazorat turini birinchi marta qayta topshirishdan o‘ta olmagan taqdirda, fakultet dekani tomonidan komissiya tuziladi. Komissiya tarkibi tegishli fan bo‘yicha professor-o‘qituvchi va soha mutaxassisleri orasidan shakllantiriladi.

Ikkinchi marta yakuniy nazorat turini o‘tkazish va talabani baholash mazkur komissiya tomonidan amalga oshiriladi.

Berilgan muddat davomida mavjud bo‘lgan qarzdorlikni topshira olmagan talaba bo‘yicha fakultet dekani bildirgi bilan oliy ta‘lim muassasasi rektorini xabardor qiladi va ushbu talaba rektor buyrug‘i asosida kursdan qoldiriladi.

Baholash natijasidan norozi bo‘lan talabalar fakultet dekani tomonidan tashkil etiladigan Apellyatsiya komissiyasiga apellyatsiya berish huquqiga ega.

Apellyatsiya komissiyasi tarkibiga talabani baholashda ishtirok etmagan tegishli fan professor-o‘qituvchilari orasidan komissiya raisi va kamida to‘rt nafar a‘zo kiritiladi.

Talaba baxolash natijasidan norozi bo‘lgan taqdirda, baholash natijasi e‘lon kilingan vaqtdan boshlab 24 soat davomida apellyatsiya berishi mumkin. Talaba tomonidan berilgan apellyatsiya Apellyatsiya komissiyasi tomonidan 2 kun ichida kurib chiqilishi lozim.

Talabaning apellyatsiyasini ko‘rib chiqishda talaba ishtirok etish huquqiga ega.

Apellyatsiya komissiyasi talabaning apellyatsiyasini ko‘rib chiqib, uning natijasi bo‘yicha tegishli qaror qabul qiladi. Qarorda talabaning tegishli fanni o‘zlashtirgani yoki o‘zlashtira olmagani ko‘rsatiladi.

Apellyatsiya komissiyasi tegishli qarorni fakultet dekani va talabaga yetkazilishini ta‘minlaydi.

III. Baxolash natijalarini qayd qilish

Nizomning 35-bandida keltirilganidek, talabalar bilimini baholash tegishli fan bo‘yicha professor-o‘qituvchi tomonidan Talabalarning fanlarni o‘zlashtirishini hisobga olish jurnalida (bundan buyon matnda Jurnal deb yuritiladi) qayd etib boriladi. Professor-o‘qituvchi qo‘shimcha ravishda talabalar bilimini baholashni elektron tizimda ham yuritishi mumkin.

Professor-o‘qituvchi Jurnalda talabaga qo‘yilgan baholarni shu kunning o‘zida qayd etib boradi. Agar talabaning bilimini baholash yozma ish shaklida o‘tkazilgan bo‘lsa, bunda professor-o‘qituvchi talabalarning natijalarini 3 kundan ko‘p bo‘lmagan muddatda Jurnalga qayd etishi lozim.

Nazorat turi bo‘yicha talabaning bilimi “3” (qoniqarli) yoki “4” (yaxshi) yoxud “5” (a‘lo) baho bilan baholanganda, nazorat turini qayta topshirishga yo‘l qo‘yilmaydi.

Talaba nazorat turi o‘tkazilgan vaqtda uzrli sabablarsiz qatnashmagan hollarda Jurnalga “0” belgisi yozib qo‘yiladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Jurnal tegishli fan bo‘yicha o‘quv mashg‘ulotlarini olib borgan professor-o‘qituvchi, kafedra mudiri va fakultet tomonidan imzolanadi hamda fakultet dekanatida saqlanadi. Jurnalning saqlanishi uchun fakultet dekani mas‘ul hisoblanadi.

Talabalarning yakuniy nazorat turi bo‘yicha baholari Jurnalga kayd etilganda, shu kunning o‘zida talabani Baholash daftariga ham yozib qo‘yilishi kerak.

Yakuniy nazorat turi bo‘yicha talabani baholash bilimi “2” (qoniqarsiz) baho bilan baholangan yoki Jurnalga “0” belgisi yozib qo‘yilgan hollarda ushbu baho yoki belgi talabani Baholash daftariga yozilmaydi.

Jurnalning o‘z vaqtida, to‘g‘ri va to‘liq yuritilishi, shuningdek undagi baho va boshqa ma‘lumotlarga asossiz o‘zgartirishlar kiritilmasligi uchun fakultet dekani va tegishli fan bo‘yicha professor-o‘qituvchi mas‘ul hisoblanadi.

Tegishli o‘quv yili yakuni bo‘yicha ishchi o‘quv rejagi fanlar bo‘yicha “3” (qoniqarli) yoki “4” (yaxshi) yoxud “5” (a‘lo) baho bilan baholangan talaba oliy ta‘lim muassasasi rektorining buyrug‘iga asosan keyingi kursga o‘tkazadi.

Baholash natijalari kafedra yig‘ilishlari, fakultet va oliy ta‘lim muassasasi Kengashlarida muntazam ravishda muhokama etib boriladi va tegishli qarorlar qabul qilinadi.

Foydalaniladigan adabiyotlar ro‘yxati

Asosiy adabiyotlar:

1. M.Aripov, M.Muhammadiyev. Informatika, informasion texnologiyalar. Darslik. T.: TDYuI, 2005 y.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T.: “Cho‘lpon”, 2013 y.
3. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T.: “Ilm Ziyos”, 2011y.
4. T. X. Holmatov, N. I. Tayloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta‘minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
5. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T.: “O‘qituvchi”, 2011 y.
6. B.Mo‘minov. Informatika. O‘quv qo‘llanma. T.: “Tafakkur-bo‘stoni”, 2014 y.

Qo‘shimcha adabiyotlar

1. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига киришиш тантанали маросимига бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Шавкат Миромонович. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қонидаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий яқунлари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
3. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
4. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (*Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда*)
5. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт-Петербург, “БХВ-Петербург” 2003 г.
6. В. М. Пестиков, А. Н. Маслобоев. Turbo PASKAL 7. 0. Изучаем на примерах. Санкт-Петербург. : “БХВ-Петербург”, 2004 г.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

7. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : “Питер”, 2003 г.
8. В. Т. Безручко. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.
9. Меняев Михаил Федорович. Информационные технология управления. Москва, «Издательский ОмегаЛ», 2003 г.
10. В.Д.Колдаев. Основы алгоритмизации и программирования. Учебный пособый, Москва ИД “Форум”- ИНФРА-М 2006 г.
11. Thomas H. Cormen. Intruduction to algorithms. Third Edition. Massachusetts Institute of Technology. The MIT Press. London 2009. 1292-p.
12. Algorithms, Fourth Edition (Deluxe): Book and 24-Part Lecture Series 1st Edition , Addison-Wesley Professional , USA, 2015
13. Цой, Маргарита и др. Создание электронных учебников. Т. : “Ўзбекистон миллий энциклопедияси”, 2007 г.

Internet saytlari

1. www.ziyonet.uz – Axborot ta’lim portali
2. www.edu.uz – Oliy va o’rta maxsus ta’lim vazirligi portali
3. www.tdpu.uz – Nizomiy nomidagi TDPU rasmiy sayti
4. [http:// corel.Deamiart.ru/](http://corel.Deamiart.ru/).
5. www.amazon.com
6. <http://www.ctc.msiu.ru/materials/Book1,2/index1.html>

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA
TARQATMA MATERIALLAR

1- Bilet

1. Koordinatalar tekisligida ikkita nuqta orasidagi masofani aniqlovchi dastur tuzing.
2. 1 dan 100 gacha bo‘lgan sonlarning yig‘indisini toping.

3. $y = \frac{\sum_{j=1}^6 \ln|x_i + 2,5|}{\prod_{i=1}^6 x_i^2}$ ni hisoblang

2- Bilet

1. To‘rtta son berilgan. Shu sonlarning o‘rta arifmetigi va o‘rta geometrigini hisoblovchi dastur tuzing.
2. 1 dan 50 gacha bo‘lgan sonlarning ko‘paytmasini toping.

3. $y = \sum_{i=1}^5 \frac{iz_i}{2^i}$ ni hisoblang

3- Bilet

2. To‘rtta son berilgan. Shu sonlarning o‘rta arifmetigi va o‘rta geometrigini hisoblovchi dastur tuzing.
3. Raqamlari yig‘indisining kubiga teng bo‘lgan barcha uch xonali sonlarni toping.

3. $y_i = \prod_{i=1}^3 a_i^2 b_i$ ni hisoblang

4- Bilet

1. Quyidagi formula bilan berilgan uchburchakka tashqi chizilgan aylananing radiusini hisoblash dasturini tuzing:

$$R = \frac{a * b * c}{4 * \sqrt{p * (p - a) * (p - b) * (p - c)}}$$

bunda p – uchburchakning yarim perimetri.

2. 2,3,4,5,6,7,8,9 sonlariga ko‘paytirilganda raqmlarining yig‘indisi o‘zgamaydigan ikki xonali sonlarni toping.

3. $A(10)$ vektor elementlarining yig‘indisini va o‘rta arifmetigini hisoblash dasturini toping.

5- Bilet

1. 1 dan m gacha natural sonlar berilgan. Shu sonlarning arifmetik progressiya yordamida yig‘indisini hisoblash dasturini tuzing.

2. Do‘konda 16 kgli, 17 kgli va 21 kgli pol buyoqlari yashiklarda mavjud. Ularni ochib ko‘rmasdan 185 kg pol buyoq tanlang. Barcha variantlarni ko‘rib chiqing.

3. $A(10)$ vektor elementlarining ko‘paytmasining va o‘rta geometrigini hisoblash dasturini toping.

6- Bilet

- 1.

$$g = \frac{1 + \cos(x + y)}{|e^x - 2y / (1 + x^2 * y^2)|} * x^3 + \arcsin(y)$$

2. Raqamlari yig‘indisi berilgan butun songa teng uch xonali barcha sonlarni toping.

3. $A(10)$ vektor elementlarini eng kichigini topish dasturini chizing

7- Bilet

1. Rombning tomoni va o‘tkir burchagi berilgan uning yuzasini hisoblash dasturini tuzing:

$$S = a^2 \sin \alpha$$

bunda a – rombning tomoni, α -esa o‘tkir burchagi

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

2. $A(10)$ vektor elementlarini eng kattasini toppish dasturini chizing
3. Barcha shunday uch xonali sonlarni topingki, ularni kvadratga ko‘targanda uchata raqami bir xil bo‘lsin, nol bundan mustasno.

8- Bilet

1. Arifmetik progressiyaning birinchi hadi va ayirmasi berilgan. Uning 30-chi hadi va dastlabki 40 ta hadining yig‘indisini hisoblash dasturini tuzing.
2. $42 \cdot 4$ yulduzchalar o‘rniga shunday raqamlarni tanlang, hosil bo‘lgan besh xonali son 72ga bo‘linsin.
3. $A(5)$ vektor elementlarini o‘tib borish tartibida joylashtiring

9- Bilet

1. Kesmaning koordinatalari berilgan. Shu kesmani o‘rtasining koordinatalarini hisoblash dasturini tuzing.
2. $A(5)$ vektor elementlarini kamayib boorish tartibida joylashtiring
3. O‘zi va raqamlari yig‘indisi 7ga karrali uch xonali sonlarni toping.

10- Bilet

1. Uchburchakning berilgan uchta burchaklari va tomoni orqali uning yuzini hisoblash dastur tuzing:

$$S = \frac{a^2 * \sin(B) * \sin(C)}{2 * \sin(A)}$$

2. Uchunchi tartibli kvadrat matrisani teskari matritsasini toppish dasturini tuzing
3. A va B vektorlar berilgan. C vektorni shunday tashkil qiling, unung elementlari A da ham, B da ham mavjud bo‘lsin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

“ALGORITMLAR” FANIDAN TEST SAVOLLARI

1. Paskalda dastur qanday xizmatchi soʻz bilan boshlanadi?
 - a) Uses
 - b) Programm
 - c) Program
 - d) Label
2. Paskalda dastur necha qismdan tashkil topgan?
 - a) 3
 - b) 4
 - c) 2
 - d) 1
3. Oʻzgaruvchilarni tavsiflash qanday xizmatchi soʻzdan keyin keladi?
 - a) crt
 - b) uses
 - c) var
 - d) label
4. Apostrof ichiga olingan bitta belgi- harf, raqam yoki maxsus belgi nima deyiladi?
 - a) satrli oʻzgarmaslar
 - b) mantiqiy oʻzgarmaslar
 - c) sonli oʻzgarmaslar
 - d) belgili oʻzgarmaslar
5. Uzunligi 255 ta belgidan oshmagan va apostrof ichiga olingan belgi- harf, raqam yoki maxsus belgilar ketma-ketligi nima deyiladi?
 - a) satrli oʻzgarmaslar
 - b) mantiqiy oʻzgarmaslar
 - c) sonli oʻzgarmaslar
 - d) belgili oʻzgarmaslar
6. Faqat True(rost) yoki False(yolgʻon) qiymatlarni qabul qiluvchilar nima deyiladi?
 - a) satrli oʻzgarmaslar
 - b) mantiqiy oʻzgarmaslar
 - c) sonli oʻzgarmaslar
 - d) belgili oʻzgarmaslar
7. Butun yoki haqiqiy qiymatlarni qabul qiluvchi oʻzgarmaslar nima deyiladi?
 - a) satrli oʻzgarmaslar
 - b) mantiqiy oʻzgarmaslar
 - c) sonli oʻzgarmaslar
 - d) belgili oʻzgarmaslar
8. Satrli oʻzgaraslarni tavsivlash uchun qanday xizmatchi soʻzdan foydalaniladi?
 - a) string
 - b) real
 - c) integer
 - d) byte
9. Dastur taʼminoti tarkibia kiritilgan funksiyalar qaysilar?
 - a) standart funksiyalar
 - b) standart protseduralar
 - c) standart ifodalar
 - d) matematik funksiyalar
10. Sonning modulini hisoblovchi funksiyani koʻrsating?
 - a) Sin(x)
 - b) sqr(x)
 - c) frac(x)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

- d) $\text{abs}(x)$
11. Sonning kvadrat ildizini hisoblovchi funksiyani ko'rsating?
- a) $\text{Sin}(x)$
b) $\text{sqrt}(x)$
c) $\text{frac}(x)$
d) $\text{abs}(x)$
11. Sonning eksponentasini hisoblovchi funksiyani ko'rsating?
- a) $\text{Sin}(x)$
b) $\text{sqrt}(x)$
c) $\text{Exp}(x)$
d) $\text{abs}(x)$
12. Sonning natural logarifmini hisoblovchi funksiyani ko'rsating?
- a) $\text{Sin}(x)$
b) $\text{sqrt}(x)$
c) $\text{frac}(x)$
d) $\text{Ln}(x)$
13. Sonning kasr qismini hisoblovchi funksiyani ko'rsating?
- a) $\text{Frac}(x)$
b) $\text{sqrt}(x)$
c) $\text{frac}(x)$
d) $\text{abs}(x)$
14. Sonning butun qismini hisoblovchi funksiyani ko'rsating?
- a) $\text{Sin}(x)$
b) $\text{Int}(x)$
c) $\text{frac}(x)$
d) $\text{abs}(x)$
15. Paskalda $\text{sqrt}(\text{sqrt}(a)-\text{sqrt}(b))$ nimani anglatadi?
- a) $\sqrt{a^2 - b^2}$
b) $\sqrt{a - b^2}$
c) $\sqrt{a^2 - b}$
d) $\sqrt{(a - b)^2}$
16. Paskalda to'g'ri yozilgan javobni toping?
- a) $\text{sin}x - \text{cos}x$
b) $\text{sqrt}(\text{sqrt}(a)-\text{sqrt}(b))$
c) $\text{sqrt}(\text{sqrt}(\text{sin}(x))-\text{sqrt}(b))$
d) $\text{tan}(x) + \text{cos}(x)$
17. O'zlashtirish operatorini ko'rsating?
- a) read
b) write
c) :=
d) readln
18. Paskalda chiqarish operatorini ko'rsating?
- a) read
b) write
c) :=
d) readln
19. Paskalda ma'lumotlarni muloqot usulda kiritish operatorini ko'rsating?
- a) read
b) write
c) :=

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

- d) WriteLn
20. Paskalda qora rangining kodi qanday?
- a) 1
 - b) 2
 - c) 0
 - d) 4
21. Paskalda och qizil rangining kodi qanday?
- a) 1
 - b) 12
 - c) 6
 - d) 4
22. TextColor(rang) operatori nima vazifa bajaradi?
- a) fon rangini belgilaydi
 - b) matn foni rangini belgilaydi
 - c) matn rangini belgilaydi
 - d) ekran rangini belgilaydi
23. TextBackGround(rang) operatori nima vazifa bajaradi?
- a) fon rangini belgilaydi
 - b) matn foni rangini belgilaydi
 - c) matn rangini belgilaydi
 - d) ekran rangini belgilaydi
24. ClrScr operatori nima vazifa bajaradi?
- a) fon rangini belgilaydi
 - b) matn foni rangini belgilaydi
 - c) matn rangini belgilaydi
 - d) yurgichni ekran boshiga keltiradi
25. GoToXY(x,y) operatori nima vazifa bajaradi?
- a) fon rangini belgilaydi
 - b) matn foni rangini belgilaydi
 - c) yurgichni ekranning kerakli joyiga keltiradi
 - d) ekran rangini belgilaydi
26. Aylanani uzunligini hisoblash formulasi Paskalda qanday yoziladi?
- a) $l:=2pR$
 - b) $l:=2*pi*r$
 - c) $l=2*pi*r$
 - d) $l:= 2*pi*r^2$
27. Paskalda o`tish operatorining ko`rinishini toping?
- a) Label
 - b) Go To
 - c) WriteLn
 - d) ReadLn
28. O`tish operatori ishlatilayotganda dasturda qanday xizmatchi so`z bo`lishi kerak?
- a) Label
 - b) Go To
 - c) WriteLn
 - d) ReadLn
29. Tarmoqlanish operatorida qo`lanilishi mumkin bo`lgan xizmatchi so`zlarni toping?
- a) Go To
 - b) label
 - c) then
 - d) for

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

30. *if <shart> then <operator yoki operatorlar ro`yxati>* operatori qanday dasturlarda qo`llaniladi?
- a) takrorlanuvchi
 - b) chiziqli
 - c) murakkab
 - d) tarmoqlanuvchi
31. Quyidagilardan xato yozilganini toping?
- a) GoTo 10
 - b) GoTo _10
 - c) GoTo 1A
 - d) GoTo-10
32. Tarmoqlanish operatori uchun quyidagi shartlardan xato yozilganini toping?
- a) $a < b$
 - b) $a < -b$
 - c) $a > b$
 - d) $-a > 0$
33. Takrorlash operatorlari necha xil bo`ladi?
- a) 2
 - b) 4
 - c) 3
 - d) 6
34. *For i:=n1 to n2 Do <operator>* operatori qanday operator hisoblanadi?
- a) parametrli takrorlash operatori
 - b) sharti avval tekshiriladigan takrorlash operatori
 - c) sharti keyin tekshiriladigan takrorlash operatori
 - d) shartsiz takrorlash operatori
35. *While <shart> Do <operator>* operatori qanday operator hisoblanadi?
- a) parametrli takrorlash operatori
 - b) sharti avval tekshiriladigan takrorlash operatori
 - c) sharti keyin tekshiriladigan takrorlash operatori
 - d) shartsiz takrorlash operatori
36. *Repeat <operatorlar> Until <shart>* operatori qanday operator hisoblanadi?
- a) parametrli takrorlash operatori
 - b) sharti avval tekshiriladigan takrorlash operatori
 - c) sharti keyin tekshiriladigan takrorlash operatori
 - d) shartsiz takrorlash operatori
37. Satrlar ketma-ketligini bir-biriga ulash amalini bajaruvchi funksiyani toping?
- a) Concat(s1,s2,...,sn)
 - b) Length(s)
 - c) Pos(b,s)
 - d) Copy(s,n1,n2)
38. Satrni uzunligi o`lchovchi funksiyani toping?
- a) Concat(s1,s2,...,sn)
 - b) Length(s)
 - c) Pos(b,s)
 - d) Copy(s,n1,n2)
39. Satrlar ko`rsatilgan belgini izlash amalini bajaruvchi funksiyani toping?
- a) Concat(s1,s2,...,sn)
 - b) Length(s)
 - c) Pos(b,s)
 - d) Copy(s,n1,n2)

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

40. Satrning ko`rsatilgan qismidan nusxa olish ulash amalini bajaruvchi funktsiyani toping?
- a) Concat(s1,s2,...,sn)
 - b) Length(s)
 - c) Pos(b,s)
 - d) Copy(s,n1,n2)
41. a:=’ Institut’ bo`lsa, Length(a) nimani aniqlaydi?
- a) 6 b) 8 c) ‘t’ d) ‘tut’
42. Pos(‘a’, ‘axborot’) nimani aniqlaydi?
- a) 6 b) 2 c) 1 d) 4
43. Copy(‘internet’,1,2)+Copy(‘Informatsiya’,3,3)+copy(‘matematika’,5,6) funktsiyasi bajarilgach natija nimaga teng bo`ladi?
- a) Infortika b) Informa c) Infmatika d) Informatika
44. a:=’Internet’ bo`lsa, Delete(a, 1,5) ning natijasini aniqlang.
- a) Inter
 - b) Net
 - c) 6
 - d) 3
45. Satrning ko`rsatilgan qismini olib tashlash protsedurasini toping.
- a) Insert(s1,s,n)
 - b) Insert(s,n)
 - c) Delete(s,n1,n2)
 - d) Delete(s,n)
46. Bir satrning ko`rsatilgan joyiga ikkinchi satrni joylashtirish protsedurasini aniqlang.
- a) Insert(s1,s,n)
 - b) Insert(s,n)
 - c) Delete(s,n1,n2)
 - d) Delete(s,n)
47. Ekranni grafik holatga o`tkazish uchun qanday ko`rsatmadan foydalaniladi?
- a) CloseGraph
 - b) InitGraph(gd,gm,<yo`l>)
 - c) InGraph
 - d) InitGraph(gd,gm)
48. Ekranning (x,y) koordinatali nuqtasini rangga bo`yash operatorini ko`rsating.
- a) PutPixel(x,y,rang) b) GetPixel(x,y) c) GetmaxX d) GetMaxY
49. Ekranning (x,y) koordinatali nuqtasi qanday rangda ekanini aniqlash operatorini ko`rsating.
- a) PutPixel(x,y,rang) b) GetPixel(x,y) c) GetmaxX d) GetMaxY
50. Ekranning maksimal koordinatali nuqtasini aniqlash operatorini ko`rsating.
- a) PutPixel(x,y,rang)
 - b) GetPixel(x,y)
 - c) GetmaxX va GetMaxY
 - d) GetmaxXY

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

“ALGORITMLAR” FANIDAN NAZORAT TOPSHIRIQLARI

1-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b x^7 - ctgx^3$

2. $y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar } x^2 + a^2 < 4, \\ x + a, & \text{agar } x^2 + a^2 = 4, \\ \ln x, & \text{agar } x^2 + a^2 > 4. \end{cases}$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

4. $s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k + n^k}$

5. 100 va 200 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing.

2-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b (x + 4)^7 - ctgx^3$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

4. $s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{tgk + n^k}$

5. 4 va 25 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

3-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b (x + c)^7 - tg(x + 7)^3$

2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

4. $s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

4-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b (x + n)^7 - ctg^m x$

2. Klaviaturadan kiritilgan butun sonlarni ishorasi va absolyut qiymatini chiqaradigan, agar nol son kiritsa, nol degan so‘z chiqadigan dastur tuzing.

3. 3 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

4. $s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\sin n^3}{ctgk + n^k}$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

5-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b^4 x - ctg^n x$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$$2. y = \begin{cases} \sin \frac{x}{\sqrt{3x+x^2}} & \text{agar } |x| < \frac{\pi}{2}, \\ \sin(\operatorname{tg}x), & \text{agar } |x| \geq \frac{\pi}{2}. \end{cases}$$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k+n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

6-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x+4)^7 - \operatorname{tg} \frac{a}{b} x$$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{\operatorname{tg} k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

7-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x + \frac{a}{b}c)^7 - \operatorname{tg}(x+7)^3$$

2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k+n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

8-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b x^7 - \operatorname{ctg} x^3$$

$$2. y = \begin{cases} \sqrt[3]{x^2+a^2}, & \text{agar } x^2+a^2 < 4, \\ x+a, & \text{agar } x^2+a^2 = 4, \\ \ln x, & \text{agar } x^2+a^2 > 4. \end{cases}$$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k+n^k}$$

5. 100 va 200 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

9-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x+4)^7 - \operatorname{ctg} x^3$$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{\operatorname{tg} k + n^k}$$

5. 4 va 25 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

10-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b(x + c)^7 - \operatorname{tg}(x + 7)^3$

2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

11-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b(x + n)^7 - \operatorname{ctg}^m x$

2. Klaviaturadan kiritilgan butun sonlarni ishorasi va absolyut qiymatini chiqaradigan, agar nol son kiritsa, nol degan so‘z chiqadigan dastur tuzing.

3. 3 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\sin n^3}{\operatorname{ctg} k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

12-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b^4 x - \operatorname{ctg}^n x$

$$2. y = \begin{cases} \frac{\sin \frac{x}{\sqrt{3x + x^2}}}{\sin(\operatorname{tg} x)}, & \text{agar } |x| < \frac{\pi}{2}, \\ \sin(\operatorname{tg} x), & \text{agar } |x| \geq \frac{\pi}{2}. \end{cases}$$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

13-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b(x + 4)^7 - \operatorname{tg} \frac{a}{b} x$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{\operatorname{tg} k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

14-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b \left(x + \frac{a}{b}c\right)^7 - \operatorname{tg}(x + 7)^3$$

2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

15-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b x^7 - \operatorname{ctg} x^3$$

$$2. y = \begin{cases} \sqrt[3]{x^2 + b^2}, & \text{agar } x^2 + b^2 < 2, \\ x + a, & \text{agar } x^2 + b^2 = 2, \\ \ln x, & \text{agar } x^2 + b^2 > 2. \end{cases}$$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k + n^k}$$

5. 100 va 200 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

16-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b (x + 4)^7 - \operatorname{tg}^2 x^3$$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{\operatorname{tg} k + n^k}$$

5. 4 va 25 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

17-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b (x + c)^7 - \operatorname{tg}(x + 7)^3$$

2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

18-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

1. $s = \log_b(x+n)^7 - ctg^m x$
2. Klaviaturadan kiritilgan butun sonlarni ishorasi va absolyut qiymatini chiqaradigan, agar nol son kiritsa, nol degan soʻz chiqadigan dastur tuzing.
3. 3 dan 20 gacha boʻlgan sonlarning koʻpaytmasini toping.
4. $s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\sin n^3}{ctgk + n^k}$
5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

19-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b^4 x - ctg^n x$
2. $y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar } x^2 + a^2 < 4, \\ x + a, & \text{agar } x^2 + a^2 = 4, \\ \ln x + 1, & \text{agar } x^2 + a^2 > 4. \end{cases}$
3. 1 dan 20 gacha boʻlgan sonlarning koʻpaytmasini toping.
4. $s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k + n^k}$
5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

20-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b(x+4)^7 - tg \frac{a}{b} x$
2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.
3. 12 dan 20 gacha boʻlgan sonlarning koʻpaytmasini toping.
4. $s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{ctgk + n^k}$
5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

21-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b(x + \frac{a}{b}c)^7 - tg(x+7)^3$
2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.
3. 10 dan 30 gacha boʻlgan sonlarning koʻpaytmasini toping.
4. $s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$
5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

22-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

1. $s = \log_b x^7 - ctgx^3$

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

$$2. y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar } x^2 + a^2 < 4, \\ x + 2 * a, & \text{agar } x^2 + a^2 = 4, \\ \ln x, & \text{agar } x^2 + a^2 > 4. \end{cases}$$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k + n^k}$$

5. 100 va 200 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

23-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x + 4)^7 - ctgx^3$$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos^3 n^3}{tgk + n^k}$$

5. 4 va 25 oraliqda tub sonlar jadvalini chiqaruvchi dastur tuzing

24-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x + c)^7 - tg(x + 7)^3$$

2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

25-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x + n)^7 - tg^m x$$

2. Klaviaturadan kiritilgan butun sonlarni ishorasi va absolyut qiymatini chiqaradigan, agar nol son kiritsa, nol degan so‘z chiqadigan dastur tuzing.

3. 3 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\sin n^3}{ctgk + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

26-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b^4 x - ctg^n x$$

$$2. y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar } x^2 + a^2 < 4, \\ x + a, & \text{agar } x^2 + a^2 = 4, \\ \ln x, & \text{agar } x^2 + a^2 > 4. \end{cases}$$

3. 1 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

27-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x + 4)^7 - \operatorname{tg} \frac{a}{b} x$$

2. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.

3. 12 dan 20 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \prod_{k=1}^4 \prod_{n=2}^6 \frac{\cos n^3}{\operatorname{tg} k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

28-bilet

Paskal dasturlash tilida quyidagi masalalarni dasturini tuzing:

$$1. s = \log_b(x + \frac{a}{b}c)^7 - \operatorname{tg}(x + 7)^3$$

$$2. y = \begin{cases} \sin^2 x & \text{agar } |x| < \frac{\pi}{4}, \\ \sin(\operatorname{tg} x), & \text{agar } |x| = \frac{\pi}{4}. \end{cases}$$

3. 10 dan 30 gacha bo‘lgan sonlarning ko‘paytmasini toping.

$$4. s = \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k + n^k}$$

5. Berilgan 2 ta sonni EKUBni hisoblovchi dastur tuzing.

Kafedra mudiri:

Fan o‘qituvchilari:

f.-m.f.n. A.A.Ibragimov

k.o‘q. D.N. Xamroyeva

o‘q.S.Q.Absobirov

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI

NAVOIY DAVLAT PEDAGOGIKA INSTITUTI

FIZIKA-MATEMATIKA FAKULTETI

“INFORMATIKA O‘QITISH METODIKASI” KAFEDRASI

“ALGORITMLAR” FANI BO‘YICHA

BAHOLASH MEZONI

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligining 2018 yil 9 avgustdagi 19-2018-son buyrug‘iga ilova

BAHOLASH MEZONI

I. Umumiy talablar

1. Namunaviy o‘quv reja va ishchi o‘quv rejada mavjud fanlardan talabalar bilimini O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligining 2018 yil 9 avgustdagi 19-2018-son buyrug‘iga asosan baholash ularning o‘qitilayotgan fanni chuqur egallashi, topshiriqlarga ijodiy yondoshuvi, mustaqil fikrlash, o‘z bilimini muntazam ravishda oshirishga intilishi hamda adabiyotlardan keng foydalanish kabi xususiyatlarni rivojlantirishi va shu tariqa raqobatbardosh mutaxassislarini tayyorlashga erishishdan iboratdir.

2. O‘quv rejasidagi har bir fanidan baholash mezon quyidagi vazifalarni bajarishga qaratilgan:

- talabalar fanni o‘zlashtirishini muntazam ravishda nazorat qilib borish, ularni semestr (o‘quv yili) davomida o‘z ustlarida uzluksiz faol ishlashlarini ta’minlash;

- talabalar bilimini haqqoniy, aniq, adolatli va shaffof baholash hamda natijalarini ularga muntazam ravishda ma’lum qilish;

- talabalarda mustaqil ishlash ko‘nikmalarini keng rivojlantirish;

- professor-o‘qituvchilarda ma’ruza, amaliy, laboratoriya va seminar mashg‘ulotlarga puxta tayyorgarlik ko‘rish, baholash savollarini tuzishda mas’uliyatini oshirish.

3. Fan bo‘yicha maksimal “5” (a‘lo) baho quyiladi. O‘quv rejasida aynan shu fanga ajratilgan soatlar miqdori bilan belgilanadi.

Oliy talim muassasalarida talabalar bilimini nazorat qilish va baholash tizimi to‘g‘risidagi Nizom va uning har bir bandi namunaviy o‘quv rejadagi va ishchi o‘quv rejadagi har bir fanning birinchi mashg‘ulotida talabalarga e‘lon qilinadi. Fan bo‘yicha talabalarning bilim saviyasi va o‘zlashtirish darajasining **Davlat ta’lim standartlari va malakaviy talablar**ga muvofiqligini ta’minlash uchun quyidagi oraliq va yakuniy nazorat turlari o‘tkaziladi.

II. Baholash turlari va shakllari

II.1. Oraliq nazorat turi

Oliy ta’lim muassasalarida talabalar bilimini nazorat qilish oraliq va yakuniy nazorat turlarini o‘tkazish orqali amalga oshiriladi.

Oraliq nazorat semestr davomida ishchi fan dasturining tegishli bo‘limi tugagandan keyin talabaning bilim va amaliy ko‘nikmalarini baholash maqsadida o‘quv mashg‘ulotlari davomida o‘tkaziladi.

Oraliq nazorat turi har bir fan bo‘yicha fanning hususiyatidan kelib chiqqan holda 2 martagacha o‘tkazilishi mumkin.

Oraliq nazorat turini o‘tkazish shakli (*yozma, og‘zaki, test va hokazo*) va muddati fanning xususiyati va fanga ajratilgan soatlardan kelib chiqib tegishli kafedra tomonidan belgilanadi.

Oraliq nazorat turining topshiriqlari tegishli kafedra professor-o‘qituvchilari tomonidan ishlab chiqiladi va mazkur kafedra mudiri tomonidan tasdiqlanadi.

Semestr davomida haftasiga 2 akademik soatdan kam bo‘lgan fanlar bo‘yicha oraliq nazorat turi o‘tkazilmaydi.

Talabaning amaliy, seminar, laboratoriya mashg‘ulotlari va mustaqil ta’lim topshiriqlarini bajarishi, shuningdek uning ushbu mashg‘ulotlardagi faolligi fan o‘qituvchisi tomonidan baholab boriladi. Baholash O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligining 2018 yil 9 avgustdagi 19-2018-son buyrug‘ining ya’ni “Oliy talim muassasalarida talabalar bilimini nazorat qilish va baholash tizimi to‘g‘risida”gi Nizomning 15-bandida nazarda tutilgan mezonlar asosida amalga oshiriladi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda faolligi uchun talabani baholash mezonida quyiladigan talablar qo‘yiladi.

Baho		Talabning bilim darajasi va malakasiga talablar
5	A‘lo	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda faol ishtirok etadi, berilgan topshiriqlarni mustaqil fikr asosida to‘g‘ri bajaradi, javoblarni izohlab ularning amaliy ahamiyatini anglay oladi, topshiriqlarni bajarishda ijodiy yondoshadi va ijodiy fikrlay oladi, o‘z fikrini to‘la ifodalay oladi, laboratoriya mashg‘ulotlarini o‘z vaqtida bajarib topshiradi, mustaqil ta’lim mavzularini to‘liq o‘zlashtiradi, talaba mustaqil xulosa va qaror qabul qiladi, mustaqil mushohada yuritadi, olgan bilimini amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunda, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda.
4	Yaxshi	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda faol ishtirok etadi, berilgan topshiriq va mashqlarni to‘g‘ri bajaradi, javoblarni izohlay oladi, fikrini mustaqil ifodalay oladi, topshiriq mohiyatini to‘la tushunadi, laboratoriya mashg‘ulotlarini o‘z vaqtida bajarib topshiradi, mustaqil ta’lim mavzularini 70%dan 90%gacha o‘zlashtiradi, talaba mustaqil mushohada yuritadi, olgan bilimini amalda qullay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda.
3	Qoniqarli	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlarda ishtirok etib topshiriqlarni o‘qituvchi yordamida to‘g‘ri bajaradi, yechimlardan olingan javoblarni mohiyatini tushunadi, masalani yechish jarayonini tushuntira oladi, mustaqil berilgan laboratoriya mashg‘ulotlarini 60% yoki 70% o‘z vaqtida bajaradi, mustaqil ta’limni qisman o‘zlashtiradi, talaba olgan bilimini amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda.
2	Qoniqarsiz	Amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta’lim topshiriqlari bajarishi hamda mashg‘ulotlardagi topshiriqlarning shartini to‘g‘ri tushunib ularni to‘liq bajara olmaydi, topshiriqlarni ba’zilar to‘g‘risida aniq tasavvurga ega bo‘lmaydi, o‘qituvchi ko‘magida ham topshiriqlarni bajarishga qiynaladi, nazariy va amaliy bilimlarini bog‘lay olmaydi, laboratoriya mashg‘ulotlarini o‘z vaqtida bajarmaydi va topshira olmaydi, mustaqil ta’limni o‘zlashtirmaydi, talaba fan dasturini o‘zlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas deb topilganda.

Talabani oraliq nazorat turi bo‘yicha baholashda, uning o‘quv mashg‘ulotlari davomida olgan baholari inobatga olinadi.

Talabalar bilimni baholash 5 baholik tizimda amalga oshiriladi. Oraliq nazorat turini o‘tkazish va mazkur nazorat turi bo‘yicha talabanning bilimni baholash tegishli fan bo‘yicha o‘quv mashg‘ulotlarini olib borgan professor-o‘qituvchi tomonidan amalga oshiriladi.

Talaba tegishli fan bo‘yicha yakuniy nazorat turi o‘tkaziladigan muddatga qadar oraliq nazorat turini topshirgan bo‘lishi shart.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

Oraliq nazorat turini topshirmagan, shuningdek ushbu nazorat turi bo‘yicha “2” (qoniqarsiz) baho bilan baholangan talaba yakuniy nazorat turiga kiritilmaydi.

Oraliq (yozma, og‘zaki) nazoratda talaba bilimini baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Savollardagi mavzularning barchasiga asoslangan, ilmiy xatoliklarga yo‘l qo‘yilmagan holda javoblar beradi, mavzu material mohiyatini to‘la tushunib yetgan bo‘ladi, ijodiy fikr yuritadi, mustaqil mushohada qiladi, nazariy bilimlarni amalda qo‘llashga misollar keltira oladi, xulosa va qarorlar qabul qilishda faol bo‘ladi, material bo‘yicha to‘la tasavvurga ega bo‘ladi va talaba ilmiy-uslubiy maqolalar yozishga loyiq bo‘ladi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi shuningdek mashg‘ulotlarda faol ishtirok etgan deb topilganda.
4	Yaxshi	Savollarning barchasiga to‘liq javob beradi, juz‘iy xatoliklarga yo‘l qo‘ymaydi. Material mohiyatni tushunib yetgan bo‘ladi, ijodiy fikr yurita oladi, nazariy bilimlarni amaliy ahamiyatini anglab yetgan bo‘ladi, material bo‘yicha tasavvurga ega bo‘lsa qo‘shimcha adabiyotlardan mustaqil foydalana olish qobiliyatiga hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi yaxshi ishtirokni inobatga olgan deb topilganda
3	Qoniqarli	Savollarga javoblar yozadi, yo‘l qo‘ygan xatolari juz‘iy xatolar bo‘lmaydi, material mohiyatini tushungan bo‘ladi, nazariy bilimlarni amaliy ahamiyatini anglagan bo‘ladi, mavzular bo‘yicha tasavvurga ega bo‘ladi va auditoriya mashg‘ulotlariga to‘liq qatnashgan bo‘ladi hamda fan (mavzu) bo‘yicha tasavvurga ega hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlarda ishtirok etib topshiriqlarni o‘qituvchi yordamida to‘g‘ri bajaradi deb topilganda
2	Qoniqarsiz	Savollarga javob berishga qiynaladi, material mohiyatini tushunmaydi, tasavvuri sayoz bo‘ladi, nazariy bilimlarni amaldagi ahamiyatni anglab yetmaydi, savollarni ko‘pchiligiga javob bera olmaydi va darslarga muntazam qatnashmagan bo‘laydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi topshiriqlarning shartini to‘g‘ri tushunib ularni bajara olmaydi deb topilganda

Oraliq (test) nazoratda talaba bilimini baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Talaba test savollarining 90%dan yuqorisiga to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi shuningdek mashg‘ulotlarda faol ishtirok etgan deb topilganda.
4	YAxshi	Talaba test savollarining 89,9%dan 70% gacha to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi yaxshi ishtirokni inobatga olgan deb topilganda

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

3	Qoniqarli	Talaba test savollarining 69,9%dan 60% gacha to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlarda ishtirok etib topshiriqlarni o‘qituvchi yordamida to‘g‘ri bajaradi deb topilganda
2	Qoniqarsiz	Talaba test savollarining 59,9% dan kamiga to‘g‘ri javob beradi hamda amaliy, laboratoriya, seminar mashg‘ulotlar va mustaqil ta‘lim topshiriqlari bajarishi hamda mashg‘ulotlardagi topshiriqlarning shartini to‘g‘ri tushunib ularni bajara olmaydi deb topilganda

Fan bo‘yicha oraliq nazorat turlarida talaba olgan baho yakuniy nazorat turiga kirish uchun.

Nizomda ko‘rsatib o‘tilgandek, talaba uzrli sabablarga ko‘ra oraliq nazorat turiga kirmagan taqdirda ushbu talabaga oraliq nazorat turini qayta topshirishga fakultet dekanining farmoyishi asosida ruxsat beriladi.

Talabaga oraliq nazorat turini qayta topshirish uchun berilgan muddat davomida talaba tomonidan qayta topshirishlar soni 2 martadan ko‘p bo‘lmasligi kerak.

Talaba oraliq nazorat turini birinchi marta qayta topshirishdan o‘ta olmagan taqdirda, fakultet dekani tomonidan komissiya tuziladi. Komissiya tarkibi tegishli fan bo‘yicha professor-o‘qituvchi va soha mutaxassisleri orasidan shakllantiriladi.

Ikkinchi marta oraliq nazorat turini o‘tkazish va talabani baholash mazkur komissiya tomonidan amalga oshiriladi.

Berilgan muddat davomida mavjud bo‘lgan qarzdorlikni topshira olmagan talaba bo‘yicha fakultet dekani bildirgi bilan oliy ta‘lim muassasasi rektorini xabardor qiladi va ushbu talaba rektor buyrug‘i asosida kursdan qoldiriladi.

II.2. YAkuniy nazorat turi

Yakuniy nazorat turi semestr yakunida tegishli fan bo‘yicha talabaning nazariy bilim va amaliy ko‘nikmalarini o‘zlashtirish darajasini aniqlash maqsadida o‘tkaziladi.

Yakuniy nazorat turini o‘tkazish shakli tegishli fan bo‘yicha kafedra tomonidan belgilanadi.

Yakuniy nazorat turi oliy ta‘lim muassasasining tegishli fakultet dekani yoki o‘quv-uslubiy bo‘lim tomonidan ishlab chiqiladigan hamda o‘quv ishlari bo‘yicha prorektor tomonidan tasdiqlanadigan YAkuniy nazorat turlarini o‘tkazish jadvalga muvofiq o‘tkaziladi.

Yakuniy(yozma, og‘zaki) nazoratda talaba bilimni baholashda quyidagi talablar qo‘yiladi:

Baho		Talaba bilimi va malakasiga talablar
5	A‘lo	Savollardagi mavzularning barchasiga asoslangan, ilmiy xatoliklarga yo‘l qo‘yilmagan holda javoblar beradi, mavzu material mohiyatini to‘la tushunib yetgan bo‘ladi, ijodiy fikr yuritadi, mustaqil mushohada qiladi, nazariy bilimlarni amalda qo‘llashga misollar keltira oladi, xulosa va qarorlar qabul qilishda faol bo‘ladi, material bo‘yicha to‘la tasavvurga ega deb topilganda.
4	YAxshi	Savollarning barchasiga to‘liq javob beradi, juz‘iy xatoliklarga yo‘l qo‘ymaydi. Material mohiyatni tushunib yetgan bo‘ladi, ijodiy fikr yurita oladi, nazariy bilimlarni amaliy ahamiyatini anglab yetgan bo‘ladi, material bo‘yicha tasavvurga ega bo‘lsa qo‘shimcha adabiyotlardan mustaqil foydalana olish qobiliyatiga ega deb topilganda
3	Qoniqarli	Savollarga javoblar yozadi, yo‘l qo‘ygan xatolari juz‘iy xatolar bo‘lmaydi, material mohiyatini tushungan bo‘ladi, nazariy

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

		bilimlarni amaliy ahamiyatini anglagan bo‘ladi, mavzular bo‘yicha tasavvurga ega bo‘ladi va auditoriya mashg‘ulotlariga to‘liq qatnashgan bo‘ladi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda
2	Qoniqarsiz	Savollarga javob berishga qiynaladi, material mohiyatini tushunmaydi, tasavvuri sayoz bo‘ladi, nazariy bilimlarni amaldagi ahamiyatni anglab yetmaydi, savollarni ko‘pchiligiga javob bera olmaydi va darslarga muntazam qatnashmagan bo‘laydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas deb topilganda

Yakuniy(test) nazoratda talaba bilimini baholashda quyidagi talablar qo‘yiladi:

Baho	Talaba bilimi va malakasiga talablar	
5	A‘lo	Talaba test savollarining 90%dan yuqorisiga to‘g‘ri javob berganda. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test materiali mohiyatini to‘la tushunib yetgan bo‘ladi, ijodiy fikr yuritadi, mustaqil mushohada qiladi, nazariy bilimlarni amalda qo‘llaydi, xulosa va qarorlar qabul qilishda faol bo‘ladi, test material bo‘yicha to‘la tasavvurga ega deb topilganda.
4	Yaxshi	Talaba test savollarining 89,9%dan 70%gachasiga to‘g‘ri javob berishi lozim. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test materiali mohiyatini tushunib yetgan bo‘ladi, test savollariga javob berishda o‘zining fikr va mulohazalariga tayanadi, nazariy bilimlarni amaliy ahamiyatini anglab yetgan bo‘ladi, test material bo‘yicha yaxshi tasavvurga ega deb topilganda
3	Qoniqarli	Talaba test savollarining 69,9%dan 60%gachasiga to‘g‘ri javob berishi lozim. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test material mohiyatini tushungan bo‘ladi, nazariy bilimlarni amaliy ahamiyatini anglagan bo‘ladi, test material bo‘yicha o‘rtacha tasavvurga ega deb topilganda
2	Qoniqarsiz	Talaba test savollarining 59,9%dan kamiga to‘g‘ri javob berganda. Test savollari joriy semestrdagi mavzularning barchasini qamrab olgan bo‘lishi lozim va talaba test savollarga javob berishga qiynaladi, material mohiyatini tushunmaydi, tasavvuri sayoz bo‘ladi, nazariy bilimlarni amaldagi ahamiyatni anglab yetmaydi, test savollarni kattf qismiga javob bera olmaydi deb topilganda

Yakuniy nazorat turini o‘tkazish va mazkur nazorat turi bo‘yicha talabning bilimini baholash o‘quv mashg‘ulotlarini olib bormagan professor-o‘qituvchi tomonidan amalga oshiriladi.

Tegishli fan bo‘yicha o‘quv mashg‘ulotlarini olib borgan professor-o‘qituvchi yakuniy nazorat turini o‘tkazishda ishtirok etishi taqiqlanadi.

Yakuniy nazorat turini o‘tkazishda kelishuv asosida boshqa oliy ta‘lim muassasalarining tegishli fan bo‘yicha professor-o‘qituvchilari jalb qilinishi mumkin.

Oliy ta‘lim muassasasida nazorat turlarini o‘tkazilishi tegishli oliy ta‘lim muassasasining ta‘lim sifatini nazorat qilish bo‘limi tomonidan doimiy ravishda o‘rganib boriladi. Bunda nazorat turlarini o‘tkazilish tartibi buzilganligi aniqlangan hollarda, o‘tkazilgan nazorat turlarining natijalari bekor qilinishi hamda tegishli nazorat turi qaytadan o‘tkazilishi mumkin.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUUA

Yakuniy nazorat turiga kirmagan yoki kiritilmagan, shuningdek ushbu nazorat turi bo‘yicha “2” (qoniqarsiz) baho bilan baholangan talaba akademik qarzdor hisoblanadi.

Nizomning 23 – bandiga asosan, talaba uzrli sabablarga ko‘ra yakuniy nazorat turiga kirmagan taqdirda ushbu talabaga yakuniy nazorat turini qayta topshirishga fakultet dekanining farmoyishi asosida ruxsat beriladi.

Bir kunda 1 tadan ortiq fan bo‘yicha yakuniy nazorat turi o‘tkazilishiga yo‘l qo‘yilmaydi. Yakuniy nazorat turlarini o‘tkazish kamida 2 kun oralig‘ida belgilanishi lozim.

Yuqorida keltirib o‘tganimizdek, bitiruvchi kurs bo‘lmagan talabalar kuzgi semestr natijalari bo‘yicha 3 tagacha fandan (fanlardan) akademik qarzdorligi bo‘lgan hollarda talabaga bir oygacha, bahorgi semestr natijalari bo‘yicha 3 tacha fandan (fanlardan) akademik qarzdorligi bo‘lgan talabaga tegishli fan (fanlar) bo‘yicha oralik va (yoki) yakuniy nazorat turlarini yangi o‘quv yili boshidan qayta topshirish uchun 1 oy muddat beriladi.

Bitiruvchi kurs talabalariga bahorgi semestr natijalari bo‘yicha o‘zlashtirmagan fandan (fanlardan) qayta topshirish uchun yakuniy davlat attestatsiyasi boshlangunga qadar ruxsat beriladi.

Fanlardan akademik qarzdorligi 4 ta va undan ko‘p bo‘lgan talabalarga qayta topshirishga ruxsat berilmaydi va ular oliy ta‘lim muassasasi rektorining buyrug‘i bilan kursdan qoldiriladi.

Talabaga yakuniy nazorat turini qayta topshirish uchun berilgan muddat davomida talaba tomonidan qayta topshirishlar soni 2 martadan ko‘p bo‘lmasligi kerak.

Talaba yakuniy nazorat turini birinchi marta qayta topshirishdan o‘ta olmagan taqdirda, fakultet dekani tomonidan komissiya tuziladi. Komissiya tarkibi tegishli fan bo‘yicha professor-o‘qituvchi va soha mutaxassisleri orasidan shakllantiriladi.

Ikkinchi marta yakuniy nazorat turini o‘tkazish va talabani baholash mazkur komissiya tomonidan amalga oshiriladi.

Berilgan muddat davomida mavjud bo‘lgan qarzdorlikni topshira olmagan talaba bo‘yicha fakultet dekani bildirgi bilan oliy ta‘lim muassasasi rektorini xabardor qiladi va ushbu talaba rektor buyrug‘i asosida kursdan qoldiriladi.

Baholash natijasidan norozi bo‘lan talabalar fakultet dekani tomonidan tashkil etiladigan Apellyatsiya komissiyasiga apellyatsiya berish huquqiga ega.

Apellyatsiya komissiyasi tarkibiga talabani baholashda ishtirok etmagan tegishli fan professor-o‘qituvchilari orasidan komissiya raisi va kamida to‘rt nafar a‘zo kiritiladi.

Talaba baxolash natijasidan norozi bo‘lgan taqdirda, baholash natijasi e‘lon kilingan vaqtdan boshlab 24 soat davomida apellyatsiya berishi mumkin. Talaba tomonidan berilgan apellyatsiya Apellyatsiya komissiyasi tomonidan 2 kun ichida kurib chiqilishi lozim.

Talabaning apellyatsiyasini ko‘rib chiqishda talaba ishtirok etish huquqiga ega.

Apellyatsiya komissiyasi talabaning apellyatsiyasini ko‘rib chiqib, uning natijasi bo‘yicha tegishli qaror qabul qiladi. Qarorda talabaning tegishli fanni o‘zlashtirgani yoki o‘zlashtira olmaganini ko‘rsatiladi.

Apellyatsiya komissiyasi tegishli qarorni fakultet dekani va talabaga yetkazilishini ta‘minlaydi.

III. Baxolash natijalarini qayd qilish

Nizomning 35-bandida keltirilganidek, talabalar bilimini baholash tegishli fan bo‘yicha professor-o‘qituvchi tomonidan Talabalarning fanlarni o‘zlashtirishini hisobga olish jurnalida (bundan buyon matnda Jurnal deb yuritiladi) qayd etib boriladi. Professor-o‘qituvchi qo‘shimcha ravishda talabalar bilimini baholashni elektron tizimda ham yuritishi mumkin.

Professor-o‘qituvchi Jurnalda talabaga qo‘yilgan baholarni shu kunning o‘zida qayd etib boradi. Agar talabaning bilimini baholash yozma ish shaklida o‘tkazilgan bo‘lsa, bunda professor-o‘qituvchi talabalarning natijalarini 3 kundan ko‘p bo‘lmagan muddatda Jurnalga qayd etishi lozim.

Nazorat turi bo‘yicha talabaning bilimi “3” (qoniqarli) yoki “4” (yaxshi) yoxud “5”

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUA

(a’lo) baho bilan baholanganda, nazorat turini qayta topshirishga yo‘l qo‘yilmaydi.

Talaba nazorat turi o‘tkazilgan vaqtda uzrli sabablarsiz qatnashmagan hollarda Jurnalga “0” belgisi yozib qo‘yiladi.

Jurnal tegishli fan bo‘yicha o‘quv mashg‘ulotlarini olib borgan professor-o‘qituvchi, kafedra mudiri va fakultet tomonidan imzolanadi hamda fakultet dekanatida saqlanadi. Jurnalning saqlanishi uchun fakultet dekani mas’ul hisoblanadi.

Talabalarning yakuniy nazorat turi bo‘yicha baholari Jurnalga kayd etilganda, shu kunning o‘zida talabaning Baholash daftariga ham yozib qo‘yilishi kerak.

Yakuniy nazorat turi bo‘yicha talabaning bilimi “2” (qoniqarsiz) baho bilan baholangan yoki Jurnalga “0” belgisi yozib qo‘yilgan hollarda ushbu baho yoki belgi talabaning Baholash daftariga yozilmaydi.

Jurnalning o‘z vaqtida, to‘g‘ri va to‘liq yuritilishi, shuningdek undagi baho va boshqa ma’lumotlarga asossiz o‘zgartirishlar kiritilmasligi uchun fakultet dekani va tegishli fan bo‘yicha professor-o‘qituvchi mas’ul hisoblanadi.

Tegishli o‘quv yili yakuni bo‘yicha ishchi o‘quv rejagi fanlar bo‘yicha “3” (qoniqarli) yoki “4” (yaxshi) yoxud “5” (a’lo) baho bilan baholangan talaba oliy ta’lim muassasasi rektorining buyrug‘iga asosan keyingi kursga o‘tkazadi.

Baholash natijalari kafedra yig‘ilishlari, fakultet va oliy ta’lim muassasasi Kengashlarida muntazam ravishda muhokama etib boriladi va tegishli qarorlar qabul qilinadi.

“ALGORITMLAR” FANIDAN O‘QUV USLUBIY MAJMUVA

VI. FOYDALANILGAN ADABIYOTLAR

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. Introduction to algorithms. Massachusetts Institute of Technology. London 2009.
2. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Сер: Классические учебники: COMPUTER SCIENCE. М.: МЦНМО, –960с., 2004.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2010 y.
4. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
5. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyo”, 2011y.
6. T. X. Holmatov, N. I. Toyloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta`minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
7. М. Ашуров, М. Мирзахмедов, Ш. Сапаев. Замоновий дастурлаш тиллари фанидан лаборатория ишлари. Т. : ТДПУ, 2008 й.
8. M.J.Yunusova, A.V.Rahimov. Dasturlash bo‘yicha praktikum. T.: “Ilm ziyu”, 2006 y.
9. А.В.Файсман. Профессиональное программирование на Турбо – Паскале. Информ Экс - Корпорейшн, 1992 г.
10. B.Boltayev, M.Mahkamov, A.Azamatov. Paskal dasturlash tili. T.; 2007 y.
11. А.Н.Марченко. Программирование в среде *Turbo Pascal 7.0*. К.,Век+, М., «ДЕСС», 1999 y.
12. Karimov R. va boshqalar. Dasturlash. T. “O‘zbekiston”. 2003 y. 206 b.
13. Матрос Д.Ш., Поднебесова Г.Б. Теория алгоритм. Учебник для педагогического образования. М.: Бином. Лаборатория знаний, - 2008. -202с.
14. Макконелл Дж. Основы современных алгоритмов. 2-доп.изд., М.: ТЕХНОСФЕРА, 366с., 2004.
15. Вирт Н. Алгоритмы и структуры данных. С примерами на Паскале. Санкт-Петербург, 352с., 2005.
16. Рейест Р. и др. Алгоритмы: построение и анализ. М., Мир, 1994.
17. Малышко В.В. Алгоритмы и алгоритмические языки. Конспект лекций для студентов Ташкентского филиала МГУ, 68с., 2006.
18. Пильщиков В.Н., Абрамов В.Г., Вылиток А.А., Горячая И.В. Машина Тьюринга и алгоритмы Маркова. Решение задач. (Учебно-методическое пособие) Московский государственный университет им. М.В.Ломоносова, Факультет вычислительной математики и кибернетики, Москва, 47с., 2006.
19. Yunusov A.S. Matematik mantiq va algoritmlar nazariyasi elementlari. Samarqand davlat universiteti nashriyoti, 182 b, 2012y.
20. Слинкин Д.А. Основы программирования на Турбо-Паскале: Учебно-методическое пособие для студентов вузов. Шадринск: Изд-во Шадринского пединститута, 2003. – 134-136 сс.
21. Фаронов В. В. Turbo Pascal. — СПб.: ВХВ- Санкт-Петербург, 2004. – 1056 с. (301-320сс)
22. М.У.Ашуров, Н.Д.Мирзахмедова. Turbo Pascal дастурлаш тили.(услубий кўлланма), Тошкент ТДПУ – 2011 (81-87)
23. A.U.Ashurov, N.D.Mirzahmedova, N.S.Haytullayeva. Algoritmash va dasturlash tillari. Informatika o‘qitish metodikasi ta’lim yo‘nalishi uchun uslubiy qo‘llanma. Toshkent – 2015 (113-115 B)
24. www.ziyonet.uz.
25. www.de.uspu.ru/Informatics/metodes/DPP/F/08/1/Index.htm.