

O'ZBEKISTON ALOQA VA AXBOROTLASHTIRISH
AGENTLIGI

TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI SAMARQAND FILIALI

INFORMATIKA VA AXBOROT TEXNOLOGIYALARI
KAFEDRASI

5521900 – “Informatika va axborot texnologiyalari” va
5140900 – “Kasb ta’limi (Informatika va axborot
texnologiyalari)” bakalavriatura yo’nalishlari I kurs
talabalari uchun

Paskal tilida dasturlash
bo’yicha amaliy va
laboratoriya mashg’ulotlari
uchun uslubiy qo’llanma

Qarshiyev A.B., Yalg’ashev O.R.

Samarqand – 2006

Ushbu uslubiy qo'llanma 5521900 – “Informatika va axborot texnologiyalari” va 5149900 – “Kasb ta'lim (Informatika va axborot texnologiyalari)” bakalavriatura yo'nalishlarining I kursida o'qitiladigan “Informatika” fanining amaliy va laboratoriya mashg'ulotlari rejasi asosida yaratilgan. Unga Paskal tilida programmashtirish bo'yicha amaliy va laboratoriya mashg'ulotlariga oid mavzular kiritilgan. Har bir mavzuda nazariy material, topshiriqlarni bajarish uchun uslubiy ko'rsatmalar, amaliy mashg'ulot mashqlari va laboratoriya ish topshiriqlari to'plami berilgan.

Uslubiy qo'llanma lotin yozuvida sodda va tushunarli til bilan yozilgan. U I kurs talabalari uchun foydali qo'llanma bo'lishi mumkin.

Qarshiyev A.B., Yalg'oshev O.R. Paskal tilida dasturlash bo'yicha amaliy va laboratoriya mashg'ulotlari uchun uslubiy qo'llanma. – Samarqand, TATU Samarqand filiali nashri, 2006. – 120 s.

Ilmiy muharrir:

Taqrizchilar: Dotsent Aminov I.

Katta o'qituvchi Normurodov T.B.

© TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI SAMARQAND FILIALI

Mundarija

1-mavzu. Ma'lumotlarning standart turlari

<i>1.1. Dastur tarkibi.....</i>	<i>5</i>
<i>1.2. Ma'lumotlarning standart turlari tavsifi.....</i>	<i>5</i>
<i>1.3. Ifodalar va ta'minlash operatori.....</i>	<i>9</i>
<i>1.4. Kiritish va chiqarishni tashkillashtirish.....</i>	<i>10</i>
<i>1.5. Savollar va mashqlar.....</i>	<i>15</i>
<i>1.6. 1-topshiriq. Kiritish - chiqarishni tashkillashtirish. Ma'lumotlarning standart turlarini o'rganish.....</i>	<i>17</i>
<i>1.7. 2-topshiriq. Ifodalarni hisoblash. Standart funksiyalardan foydalanish.....</i>	<i>21</i>

2-mavzu. Murakkab operatorlar.

<i>2.1. Shartli operator. Tarkibli operator.....</i>	<i>28</i>
<i>2.2. Takrorlash operatorlari.....</i>	<i>32</i>
<i>2.3. Savollar va mashqlar.....</i>	<i>36</i>
<i>2.4. 1-topshiriq. Tenglama va tengsizliklarni yechish.....</i>	<i>37</i>
<i>2.5. 2-topshiriq. Funksiya qiymatlari javdalini tuzish.....</i>	<i>39</i>
<i>2.6. 3-topshiriq. Dasturda takrorlanishni tashkillashtirish.....</i>	<i>40</i>
<i>2.7. 4-topshiriq. Belgili ma'lumotlarni qayta ishlashda murakkab operatorlardan foydalanish.....</i>	<i>44</i>

3-mavzu. Sanoqli va skalyar turlar

<i>3.1. Sanoqli skalyar turlar.....</i>	<i>46</i>
<i>3.2. Cheklangan turlar.....</i>	<i>48</i>
<i>3.3. Tanlov operatori.....</i>	<i>48</i>
<i>3.4. Savollar va mashqlar.....</i>	<i>51</i>
<i>3.5. 1-topshiriq. Sanoqli turlar bilan ishlash.....</i>	<i>54</i>

4-mavzu. Massivlar turi

<i>4.1. Massivlarni tavsiflash.....</i>	<i>59</i>
<i>4.2. Yig'ilgan massivlar.....</i>	<i>61</i>
<i>4.3. Massiv elementlarini kiritish va chiqarish.....</i>	<i>62</i>
<i>4.4. Savollar va mashqlar.....</i>	<i>64</i>
<i>4.5. 1- mashg'ulot. Sonli massivlar bilan ishlash.....</i>	<i>67</i>

4.6. 2–mashg'ulot. Yig'ilgan belgili massivlar bilan ishlash....	71
5-bo'lim. Aralash tur. Birlashtiruvchi operator.	
5.1. Yozuvlar.....	74
5.2. Birlashtirish operatori.....	75
5.3. Savollar va mashqlar.....	76
5.4. 1-Topshiriq: Aralash tur bilan ishlash.....	78
6-mavzu. Protsedura va funksiyalar.	
6.1. Protsedura tavsifi.....	82
6.2. Funksiya tavsifi.....	86
6.3. Savollar va mashqlar.....	87
6.4. 1-topshiriq. Funksiyalardan foydalanib dastur tuzish.....	89
6.5. 2-topshiriq. Protseduralar qatnashgan dasturlar tuzish..	92
7-bo'lim. To'plamlar turi	
7.1. To'plamlar turi o'zgaruvchilari.....	100
7.2. To'plamlar ustida amallar.....	101
7.3. Savollar va mashqlar.....	105
7.4. 1-Topshiriq. To'plamlar turidagi qiymatlar bilan ishlash.....	107
8-bo'lim. Fayllar turi	
8.1. Fayllar bilan ishlash protseduralari va funksiyalari.....	109
8.2. Savollar va mashqlar.....	116
8.3. 1–topshiriq. Vaqtinchalik fayllar bilan ishlash.....	116
8.4. 2-topshiriq. Tashqi fayllar bilan ishlash.....	118

1-MAVZU. MA'LUMOTLARNING STANDART TURLARI

Bu bo'limda ma'lumotlarning butun, haqiqiy, belgili va mantiqiy turlari qaraladi. Bu turlar o'zgaruvchilari uchun aniqlangan amallar va standart funksiyalar keltiriladi.

1.1. Dastur tarkibi

Paskal tilida dastur sarlavha va blok deb nomlanuvchi asosiy qismlardan iborat. Blok bir necha qismlardan tuziladi. Bo'limlar ko'pi bilan oltita bo'lishi mumkin. Bo'limlar qat'iy ravishda quyidagi tartibda joylashtiriladi.

1. nishonlar bo'limi;
2. o'zgarmaslar bo'limi;
3. turlar bo'limi;
4. o'zgaruvchilar bo'limi;
5. protsedura va funksiyalar bo'limi;
6. operatorlar bo'limi.

Operatorlar bo'limi operatorli qavs, ya'ni *begin .. end* maxsus so'zlari orasida beriladi. Unda EHM bajarishi kerak bo'lgan amallar ketma-ketligi ko'rsatiladi. Qolgan hamma bo'limlar tavsiflovchi xarakterga ega.

Oxirgi bo'limdan tashqari hamma bo'limlar dasturda qatnashmasligi mumkin. Bo'limlar va operatorlar nuqtali vergul (;) bilan ajratiladi. Dastur oxirida esa nuqta (.) turishi shart.

Dasturning ixtiyoriy qismida izohlar berilishi mumkin, bu hol dastur buzilishiga olib kelmaydi. Izohlar figurali qavslar {} orasida beriladi. Agar tizimda figurali qavslar bo'lmasa ular o'rniga (* va *) belgilari qo'llaniladi.

1.2. Ma'lumotlarning standart turlari tavsifi.

Paskal tilida yozilgan dastur ma'lumotlar deb nomlanuvchi ob'yektlar ustida amallar bajaradi.

Ma'lumotlarning har bir elementi dasturda yoki o'zgarmas yoki o'zgaruvchi bo'lishi mumkin. Har bir o'zgaruvchi uchun u qabul qilishi mumkin bo'lgan qiymatlar, uning ustida bajarilishi mumkin bo'lgan amallarni aniqlovchi tur ko'rsatiladi. O'zgaruvchi turi o'zgaruvchilar bo'limida ko'rsatiladi. O'zgarmas o'zgaruvchidan dastur bajarilishi vaqtida qiymatlarining o'zgarماسligi va o'zgarماسlar bo'limida tavsiflanishi bilan farqlanadi. O'zgarماسning turi uning qiymati yordamida aniqlanadi.

Paskal tili turlar bo'limida dasturda ko'rsatiladigan ma'lumotlarning murakkab turlarini yaratish imkonini beradi. Murakkab turlar quyidagi elementar turlar yordamida yaratiladi:

- to'rtta standart – butun (*integer*), haqiqiy (*real*), belgili (*char*) va mantiqiy (*Boolean*);
- sanoqli.

Butun tur. Paskal tilida butun tur butun sonlarning berilishini ta'minlaydi. Har qanday hisoblash mashinasida butun sonlar oralig'ining chegarasi mavjud. Butun sonlar butun sonlar oralig'ida joylashishi kerak. Aks holda xatolik kelib chiqadi. Butun sonlarga misol, 73; -3; 0; +64.

Butun turga tegishli o'zgaruvchi ustida quyidagi arifmetik amallar aniqlangan: + (qo'shish), - (ayirish), * (ko'paytirish), / (bo'lish), *div* (butun bo'lish), *mod* (butun bo'lishdan qolgan qoldiq). Bo'lish amalidan tashqari hamma amallar natijalari butun turga tegishli, bo'lish amali natijasi esa har doim haqiqiy turga tegishli bo'ladi.

Bundan tashqari, natijasi mantiqiy turga tegishli bo'ladigan = (teng), <> (teng emas), <, >, <=, >= munosabat amallari ham aniqlangan.

Butun tur argumentlari bilan quyidagi standart funksiyalar qo'llanilishi mumkin:

Funksiya nomi	Matematik ko'rinishi	Natija turi
$Sin(x)$	$Sin\ x$	<i>Real</i>
$Cos(x)$	$Cos\ x$	<i>Real</i>
$Ln(x)$	$Ln\ x$	<i>Real</i>
$Sqrt(x)$	$\sqrt{x}$	<i>Real</i>
$Arctan(x)$	$Arctan\ x$	<i>Real</i>
$Exp(x)$	e^x	<i>Real</i>
$Sqr(x)$	x^2	<i>Integer</i>
$Abs(x)$	$ x $	<i>Integer</i>

Bundan tashqari, quyidagi funksiyalar aniqlangan:

$Odd(x)$ – funksiya natijasi mantiqiy qiymatdan iborat bo'lib toq argument uchun – *True*, juft argument uchun – *False*

$Succ(x) - x$ bir birlikka oshiriladi;

$Pred(x) - x$ bir birlikka kamaytiriladi;

Haqiqiy tur. Paskal–dasturda haqiqiy turga tegishli o'zgaruvchilar va o'zgarmaslar matematikadagi haqiqiy son kabi qo'llaniladi. Paskal tilida haqiqiy son fiksirlangan nuqtali son ko'rinishida yoziladi. Masalan, 151.3487 ; 0.001 ; 21.3 . Juda katta yoki juda kichik sonlarni ekponensial shaklda tasvirlash qulayroq: $2.108E-10$; $1.1E7$; $3.4E+5$, bu yerda, masalan, $2.108E-10 = 2,108 \cdot 10^{-10}$.

Haqiqiy turga tegishli o'zgaruvchi ustida quyidagi arifmetik amallar aniqlangan: $+$ (qo'shish), $-$ (ayirish), $*$ (ko'paytirish), $/$ (bo'lish). Bundan tashqari, munosabat amallari ham o'rinli.

Haqiqiy turga terishli argument bilan quyidagi standart funksiyalar qo'llanilishi mumkin: $sin(x)$, $cos(x)$, $ln(x)$, $sqrt(x)$, $arctan(x)$, $exp(x)$, $sqr(x)$, $abs(x)$; Bu funksiyalarning barchasining natijasi haqiqiy turga tegishli.

Haqiqiy turga tegishli qiymatni butun turga tegishli qiymatga akslantiruvchi quyidagi standart funksiyalar aniqlangan:

◆ $Trunc(x)$ – argumentning kasr qismini tashlab yuborish yo'li bilan butun turga tegishli qiymat hosil qiladi;

◆ $Round(x)$ – eng yaqin butun sonni olish yo'li bilan butun turga tegishli qiymat hosil qiladi;

Masalan,

$x=21.53$, $trunc(x)=21$; $round(x)=22$;

$x=-2.7$, $trunc(x)=-2$; $round(x)=-3$;

Belgilar turi. Kompyuterda ishlatiladigan barcha belgilar (harflar, raqamlar, amal belgilari va h.k.) belgi turidagi qiymat bo'lib hisoblanadi.

Apostrof ichiga olingan simvol belgilar turiga tegishli o'zgarmas hisoblanadi. Masalan, ' X ', ' 5 ', ' $=$ '.

Bir-biriga teskari ikki standart funksiyalar mavjud:

$Ord(S)$ – tartiblangan belgilar to'plamidagi S belgining tartib nomerini beradi; Masalan, $ord(':')=58$; $ord('5')=53$;

$Chr(i)$ – tartiblangan belgilar to'plamida i nomerda turgan simvolni beradi. Masalan, $Chr(66)='B'$; $Chr(57)='9'$;

Ko'rinib turibdiki, $Chr(Ord(S))=S$, $Ord(Chr(i))=i$. Belgilar turiga tegishli o'zgaruvchilar ustida munosabat amallari aniqlangan. $C1$ va $C2$ – belgilar turiga tegishli belgilar bo'lsin. $Ord(C1)<Ord(C2)$ shart bajarilgandagina $C1<C2$ shart bajariladi.

Belgilar turiga tegishli argumentga $Pred(S)$ va $Succ(S)$ funksiyalari qo'llaniladi. $Pred(S)$ funksiya S belgidan bitta oldingi qiymatni beradi. $Succ(S)$ funksiya S belgidan bitta keyingi qiymatni beradi. Masalan, $Pred('B')=A$, $Succ('9')=':'$; Bundan tashqari quyidagi tengliklar o'rinli:

$Pred(S)=Chr(Ord(S)-1)$ va $Succ(S)=Chr(Ord(S)+1)$

Mantiqiy tur. Mantiqiy turga tegishli o'zgaruvchi *true* (rost) yoki *false* (yolg'on) qiymatlardan birini qabul qilishi

mumkin. Ular ustida *and* (kon'yunksiya), *or* (diz'yunksiya) va *not* (bekor qilish) mantiqiy operatsiyalar aniqlangan. Mantiqiy turda $false < true$ munosabat aniqlangan. Bu munosabat amallari nafaqat butun, haqiqiy, belgilar turlarida, balki mantiqiy turda ham qabul qilinganligini bildiradi.

False va *True* qiymatlarni ikki elementdan tashkil topgan tartiblangan to'plam sifatida qarash mumkin. Quyidagi standart funksiyalar aniqlangan.

$$\begin{aligned} Ord(False) &= 0; Ord(True) = 1; \\ Succ(False) &= True; Succ(True) = False; \end{aligned}$$

1.3. Ifodalar va ta'minlash operatori.

Bu mavzuda faqat standart turga tegishli bo'lgan o'zgarmas va o'zgaruvchilarni o'z ichiga oluvchi ifodalar va ta'minlash operatorlari qaraladi.

Ifodalar. Ifodalarning qiymati mavjud qavslar va amallarning tartibiga mos holda hisoblanadi.

Quyida amallar ustunlik darajasining kamayishi tartibida yozilgan:

$$\begin{aligned} ¬ \\ &*, /, mod, div, and \\ &+, -, or \\ &<, <=, >, >=, <>, = \end{aligned}$$

Bu ro'yxatda bir satrdagi amallar bir xil ustunlik darajasiga ega. Bir xil ustunlik darajasidagi amallar ifodada qatnashish tartibi bo'yicha chapdan o'ngga qarab bajariladi. Qavslar ichidagi ifodalar birinchi navbatda hisoblanadi.

Har ikkalasi haqiqiy, shuningdek biri haqiqiy va biri butun turga tegishli kattaliklar ustida bajariladigan amallar natijasida haqiqiy turga tegishli son hosil bo'ladi. Ifodaning turi barcha amallar natijalari turlarini ketma-ket aniqlash yordamida topiladi. Masalan, $(Ter - \ln(x) > 2.5 * kl)$ and

($Ord(S) < Ni$) ifodaning qiymati va turi quyidagi tartibda aniqlanadi (Ni – integer; ter, x – real; s – Char):

Amal	Natija turi
$A1 = Ln(x)$	Real
$A2 = 2.5 * kl$	Real
$A3 = ter - a1$	Real
$A4 = A3 > A2$	Boolean
$A5 = Ord(S)$	Integer
$A6 = A5 < ni$	Boolean
$A4 \text{ and } A6$	Boolean

Ta'minlash operatori. Bu operator $:=$ belgisi bilan ifodalanadi. Ta'minlash operatori bajarilishi natijasida belgining o'ng tomonidagi ifoda hisoblanadi va uning qiymati chap qismidagi o'zgaruvchiga ta'minlanadi. Bu holda ifoda turi o'zgaruvchi turiga mos kelishi shart. Bundan tashqari, haqiqiy turga tegishli o'zgaruvchiga butun turdagi ifoda ta'minlanishi mumkin, aksincha bo'lgan esa hol man etiladi. Ta'minlash operatoriga misollar: (x, u o'zgaruvchilar – haqiqiy, a, k, m lar esa butun turga tegishli)

$$x := y * a - 2/k; \quad m := k * k * k;$$

$$y := a * k + m - 7 * k; \quad k := \text{round}(\exp(16 * \ln(x)));$$

1.4. Kiritish va chiqarishni tashkillashtirish.

Kiritish va chiqarish tezkor (operativ) xotira va tashqi qurilmalar orasida axborot almashish bilan bog'langan. Bu bo'limda quyidagi kiritish–chiqarish protseduralarini qaraymiz: *Read, Readln, Write, Writeln*. Bu protseduralar kiritish–chiqarishning standart fayllari deb ataluvchi *Input* va *Output* nomli fayllarni ishlatadi.

Boshlang'ich ma'lumotlarning berilishi. Dastur bajarilishi uchun kerak bo'lgan boshlang'ich ma'lumotlarni berish har xil usullar bilan tashkillashtirilishi mumkin. Bu usullarni misollarda qaraymiz.

1. Kiritiladigan qiymatlar dasturning o'zgarmaslar bo'limida beriladi. Aytaylik, $a=2.5$; $x=7.3$; $k=-17.5$; $t=548$; ekanligidan foydalanib $y=ax^2+kx-t$ ni hisoblash talab qilingan bo'lsin. Bu masala uchun dasturning o'zgarmaslar bo'limi va jismi quyidagi ko'rinishga ega bo'ladi:

Const $a:=2.5$; $x:=7.3$; $k:=-17.5$; $t:=548$;

Begin

$y=a*x*x + k*x-t$;

End.

Bu dasturda o'zgarmaslarning turi oshkor holda ko'rsatilmagan. Tenglik belgisidan o'ngdagi son turi o'zgarmas turi sifatida olinadi, ya'ni, a , x , k o'zgarmaslarga haqiqiy tur, t nomli o'zgarmasga esa butun tur o'zlashtiriladi.

Biroq boshlang'ich ma'lumotlarni berishning bu usuli hamisha ham qulay emas, chunki dastur jismida o'zgarmaslar uchun boshqa qiymatlar berish mumkin emas. Dastur jismida o'zgaruvchi qiymatini o'zgartirish mumkin emas.

2. Dasturning o'zgaruvchilar bo'limida a , x , k , t o'zgaruvchilar tavsiflanadi, operatorlar bo'limida esa ularga mos qiymatlar ta'minlanadi.

Var $a,x,k,t,y:real$;

Begin

$a:=2.5$; $x:=7.3$; $k:=-17.5$; $t:=548$;

$y=a*x*x + k*x-t$;

.....

End.

Bu holda parametrlarning qiymatlarini variantlash imkoniyati kengayadi, dasturda ularning qiymatini o'zgartirish mumkin.

3. Dasturning operatorlar bo'limida kiritish operatorlari *Read* yoki *Readln* qo'llaniladi. Bu operatorlar dasturning har xil boshlang'ich ma'lumotlar to'plami bilan bajarilishini ta'minlaydi.

Masalan, 2.5; 7.3; -17.5; 548 sonlarini kiritish talab qilinsin. Kiritishni bir necha usullarda bajarish mumkin:

a) *Read* operatorini bir marta qo'llash:

Var a, b, d, t:real;

Begin

Read(a, b, d, t);

.....

End.

Ma'lumotlar bir satrda 2.5 7.3 -17.5 548 ko'rinishda yoki bir necha satrlarga yoziladi va "Enter" tugmasi bosiladi. Shundan so'ng *Read* operatori bajarilishi natijasida *a* o'zgaruvchiga 2.5, *b* o'zgaruvchiga 7.3, *c* o'zgaruvchiga -17.5, *t* o'zgaruvchiga 548 ta'minlanadi.

b) *Read* operatorini bir necha marta qo'llash:

..... *Read(a,b,d); Read(t);*.....

Bunda *a, b, d, t* o'zgaruvchilarga mos sonlar **a)** banddagi kabi kiritiladi.

d) *Read* va *Readln* operatorlarini qo'llash:

Readln(a,b,d); Read(t);

Bu operatorlar orasidagi farq shundaki, *Readln* joriy satrdagi sonlarni kiritib bo'lgach, satr ko'chirishni bajaradi, ya'ni keyingi kiritish yoki chiqarish yangi satrning boshida amalga oshiriladi. *a, b, d* o'zgaruvchilarga mos sonlar bitta satrda kiritiladi, *t* o'zgaruvchiga mos son esa yangi satr boshidan kiritiladi.

Butun yoki haqiqiy turga tegishli sonlarni kiritish bir xil amalga oshiriladi. Belgili turga tegishli qiymatlarni kiritishning asosiy xususiyatlarini qarab o'tamiz:

- ma'lumot kiritishda belgilar orasida bo'sh joy kerak emas.

- muloqot rejimida ish boshlanishida *Input* kiritish faylida avtomatik ravishda bo'sh joy va "satr oxiri" belgisidan tashkil topgan bo'sh satr hosil qilinadi. Sonli ma'lumot kiritishda bu bo'sh joy belgisi ajratuvchi kabi qaraladi. Belgili ma'lumotni kiritishda bo'sh joy belgisi asosiy belgilardan hisoblanadi.

Masalan, dasturda dastlab belgilar turiga tegishli x , y o'zgaruvchilar kiritilishi kerak bo'lsin. U holda dastur qismi quyidagi ko'rinishga ega bo'ladi.

....*Readln*; *Read*(x,y);.....

Birinchi operator bo'sh joyni o'tkazib yuboradi, ikkinchisi ikki belgini kiritadi va ularning qiymatini mos ravishda x va y o'zgaruvchilarga ta'minlaydi.

Ma'lumotlarni chiqarish. Chiqarish operatorining umumiy ko'rinishi quyidagicha:

Write ($V1[:W1[:D1]]$, . . . , $Vn[:Wn[:Dn]]$);

Writeln ($V1[:W1[:D1]]$, . . . , $Vn[:Wn[:Dn]]$);

Bu yerda operatorning qatnashishi zaruriy bo'lmagan qismlari kvadrat qavslar ichiga olingan; $V1$, $V2$, , Vn - qiymati chiqariladigan ifodalar; $W1$, $W2$, , Wn va $D1$, $D2$, , Dn lar butun turdagi ifodalar bo'lib, W_i chiqariladigan qiymat uchun ajratilgan umumiy pozitsiyalar sonini, D_i esa verguldan keyingi pozitsiyalar sonini bildiradi (D_i - faqat *real* turi uchun ko'rsatilishi mumkin).

Writeln operatori qiymatlar chiqarilayotgan qatorning tugashiga olib keladi. Navbatdagi chiqarish operatori qiymatlarni yangi qatordan boshlab chiqara boshlaydi.

Agar chiqariladigan qiymatda pozitsiyalar W_i dan kam bo'lsa, bunday holda u chap tomonidan probellar bilan to'ldiriladi. Agar W_i ko'rsatilmagan bo'lsa, u holda ma'lum bir

standart uzunlikdagi pozitsiyalar soni ishlatiladi. Bu standart uzunlik har - xil EHMLar uchun har xil bo'lishi mumkin.

Misollar:

1. *Write (a,b); Write (c);*

Bu yerda hamma sonlar bir qatorda chiqariladi. Har bir son uchun standart pozitsiyalar soni ajratiladi. Agar $a=15$, $b=-43$, $c=3$ bo'lsa, u holda $15 \quad -43 \quad 3$ qator chiqariladi. Xuddi shunday qatorni *Write(a, b, c)* operatori bilan ham chiqarish mumkin

2. *Writeln (a,b); Write (c);*

a va b ning qiymatlari bir qatorda, c esa navbatdagi qatorning boshidan boshlab chiqariladi. Natija:

$15 \quad -43$
 3

3. *Write (a: 2, b: 5, c: 4);*

$15 \quad -43 \quad 3$

Agar pozitsiyalar soni kam ko'rsatilgan bo'lsa, u kerakli o'lvogacha avtomatik ravishda oshiriladi.

4. *Write ('a=', a:2, 'b=', b: 3, 'c=', c: 1);*

$a = 15 \quad b = -43 \quad c = 3$

5. Aytaylik a, b, c *real* turdagi o'zgaruvchilar va qiymatlari:

$a = 841,316 \quad b = 1748,5 \quad c = -0,45834$

Writeln(a:8:3, b:9:2, c:10:5); operatori bajarilganda

$841,316 \quad 1748,5 \quad -0,45834$

qator bosmaga chiqariladi.

Agar chiqarilayotgan ifoda *char* (belgi) turida bo'lsa va pozitsiyalar soni ko'rsatilmagan bo'lsa, u holda qiymat uchun bir pozitsiya ajratiladi. Masalan, $S1$ ning qiymati $*$ va $S2$ niki W bo'lsin. U holda

Writeln('S1=', S1:5, 'S2=', S2);

operatori bajarilgach

$S1 = \quad * \quad S2 = W$

qator bosmaga chiqadi.

Agar ifoda *boolean* (mantiqiy) turga tegishli bo'lsa, u holda *True* yoki *False* qiymatlar chiqariladi. Masalan, *Writeln* ($8 > 3 : 7, 10 < 5 : 6$); operator ishining natijasi *True False* bo'ladi.

1.5. Savollar va mashqlar

1. Bo'lish amalini butun tur o'zgaruvchisi ustida bajarish mumkinmi?

2. 15 ni 4 ga bo'lganda natija qaysi turga tegishli bo'ladi?

3. *Odd*(15) va *Odd*(26) nimaga teng?

4. *Succ*(137) va *Succ* (26) nimaga teng?

5. Quyida keltirilgan yozuvlarning qaysilari haqiqiy turga tegishli emas va nima uchun?

a) 7.

d) -6.1

g) 0.0

j) 9

b) .0E-2

e) 0.1E-5

h) -5.3E4

k) 2.3E+3

c) -71

f) 0.31

i) .456

l) 2,1

6. Nimaga teng?

a) *Trunc*(5.61)

d) *Trunc*(-5.61)

b) *Round*(17. 16)

e) *Round*(17. 96)

c) *Round*(-17. 16)

f) *Round*(-17. 96)

7. Quyida keltirilgan yozuvlarning qaysilari xato va nima uchun?

a) *Odd*(17.1)

d) *Cos*(32.1)

b) *Cos*(5)

e) *Sin* 0.2

c) *Succ*(3.2)

f) *Pred*(7)

8. Nimaga teng?

a) *Odd*(*Chr*(49))

d) *Chr*(*Ord*('*'))

b) *Pred*('B')

e) *Succ*('B')

9. Mantiqiy turga tegishli o'zgaruvchi ustida qanday amallar aniqlangan?

10. Paskal tili qoidalari bo'yicha quyida keltirilgan yozuvlarning qaysi birlari to'g'ri?

- 11. Paskal tilida quyidagi ifodalarni yozing:**

12. Quyidagi ifodalarning natijalari turini aniqlang. Bu yerda $i, j, k: integer$; $x, y, z: real$;

13. Agar $i, j, k: integer$; $x, y, z: real$; $a, b: Boolean$; bo'lsa quyidagi keltirilgan ta' minlash operatorlaridan qaysilari to'g'ri?

- 14.** Agar $a=2,5$; $b=7,8$; $c=-17,3$; $m=5$; $x=8,7$; $e=true$; berilgan bo'lsa quyidagi ifodlarning qiymatini toping

- 15.** Quyida aytilganlarni ifoda ko'rinishida yozing:

- d)** (x,y) koordinatali a nuqta markazi koordinatalar boshida bo'lgan va radiusi 2,3 bo'lgan aylana ichida yotadi;

- e) (x,y) koordinatali a nuqta markazi koordinatalar boshida bo'lgan va radiusi l bo'lgan aylana ichida yoki chegarasida yotadi;

16. Mantiqiy tur qiymatini *Read* operatori yordamida kiritish mumkinmi?

17. Quyida berilgan operatorlar bajarilishi natijasida qanday natija olinadi?

- a) *Write('a':3,2:1); Writeln(5*3.2);*
- b) *Writeln(5<6,5<=6:8); Writeln('Tamom');*
- c) *Writeln(5*3,2:1, 'Natija');*

18. Ta'minlash operatorini qo'llamasdan quyida berilgan ifodaning qiymatini toping:

$$\frac{5,23 + 7,6^2 + \sin \frac{\pi}{7}}{\sin \frac{2\pi}{7} + 3,1}$$

*1.6. 1–topshiriq. Kiritish - chiqarishni tashkillashtirish.
Ma'lumotlarning standart turlarini o'rganish.*

Topshiriq maqsadi:

- 1. Oddiy dastur tuzilmasini o'zlashtirish;
- 2. Ma'lumotlarning standart turilarga tegishli qiymatlarni kiritish va chiqarish ko'nikmasini olish;
- 3. Muloqot rejimida ishlash amaliy ko'nikmasini olish;

Masalaning qo'yilishi:

Ma'lumotlarning berilgan turi uchun mos keluvchi sarlavhalar qo'yib kiritish va chiqarishni boshqarish.

Hisobot mazmuni:

- 1. Masalaning qo'yilishi;
- 2. Dastur matni va uning bajarilishi natijasi;
- 3. Xulosa;

Uslubiy ko'rsatmalar:

1. Klaviaturadan sonlar va belgilarni kiritilayotganda hohish bo'yicha bitta yoki har xil satrlarga yozish mumkin. Bu holda, ya'ni keyingi satrdan kiritish oldingi operator *Readln* bo'lganda amalga oshishini yodda tutish kerak.

2. Mantiqiy tur o'zgaruvchisiga qiymat yoki o'zgarmaslar bo'limida yoki operatorlar bo'limida ta'minlash operatori yordamida beriladi. Mantiqiy tur o'zgaruvchisiga qiymat kiritish mumkin emas.

3. Muloqot rejimida ishlashda kiritish operatoridan oldin chiqarish operatorini qo'llab, bajariladigan ish-harakat haqida ma'lumot beriladi. Masalan: *Writeln('k, l, m butun sonlarni kiriting');*

4. Ma'lumotlarni kiritish ikki usul bilan olib boriladi: standart shaklda va chiqarish maydoni kengligini ko'rsatish bilan, bu holda chiqariladigan ma'lumotga sarlavha qo'yish zarur bo'ladi.

5. Dasturga misol.

Boshlang'ich ma'lumotlar: *k=-73; l=12; m=21425; x=-4192,21; y=21,341; s1='*'; s2='v'; s3='f';*

Program Kirit_Chicar;

Const log=True;

Var k,l,m:integer; x,y:real; s1,s2,s3:char;

Begin

Writeln('k, l, m butun sonlarni kiriting');

Read(k,l,m);

Writeln(' ':30, 'butun sonlar');

Writeln(' ':15, 'standart format',' ':12,'maydonning berilgan kengligi');

Writeln(' ',k, l, m, ' ':5, 'k=',k:3, ' l=',l:2, ' m=',m:5);

Writeln;

Writeln('x va y haqiqiy sonlarni kiriting');

Readln;

```

Writeln(' ':25,'haqiqiy sonlar');
Writeln(' ':5,'Standart format',' ':18,' maydonning berilgan
kengligi');
Writeln(' ':5,x, y, ' ':10,'x=',k:8:2,' y=',y:6:3);
Writeln;
Writeln('s1,s2,s3 belgilarni kiriting');
Readln(s1,s2,s3);
Writeln(' ':20,'belgili o'zgaruvchilar:', 's1=', s1, 's2=',s2,
's3=',s3);
Writeln('5 maydon kengligi bilan belgili o'zgaruvchilarni
chiqarish:', s1:5, s2:5, s3:5);
Writeln('Mantiqiy o'zgaruvchi=',log);
End.

```

Dasturning ishlashi:

k, l, m butun sonlarni kiriting

-73 12 21475

Butun sonlar

standart format maydonning berilgan kengligi

-73 12 21475 k=-73 l=12 m=21475

x va y haqiqiy sonlarni kiriting

-4192.21 21.345

Haqiqiy sonlar

standart format maydonning berilgan kengligi

-4.192210E+03 2.134500E+01 x=-4192.21

y=21.345

s1,s2,s3 belgilarni kiriting

**v f*

Belgili o'zgaruvchilar: *s1=* s2=v s3=f*

5 maydon kengligi bilan belgili o'zgaruvchilarni chiqarish:

** v f*

Mantiqiy o'zgaruvchi=True;

Topshiriq variantlari

	Butun sonlar			Haqiqiy sonlar				Belgilar soni	Mantiqiy o'zgaruvchi qiymatini chiqarish
Variant nomeri	Miqdori	Format		Miqdori	Format				
		Standart	Maydon kengligi		Standart	Maydon kengligi	Nuqtadan keyingi belgilar soni		
1	2	+	5	4	+	6	2	4	True
2	3	+	6	2	+	7	3	5	False
3	3	+	6	2	+	6	3	3	True
4	2	+	5	4	+	6	2	6	True
5	4	+	4	3	+	7	4	4	False
6	2	+	3	3	+	5	1	3	True
7	3	+	2	4	+	8	3	2	True
8	3	+	5	4	+	6	2	5	False
9	4	+	7	3	+	5	1	6	False
10	2	+	6	5	+		4	7	True
11	2	+	6	5	+	6	2	6	True
12	3	+	4	6	+	6	2	5	False
13	3	+	3	6	+	5	2	4	False
14	3	+	2	4	+	5	2	3	True
15	4	+	5	5	+	7	3	4	True
16	2	+	7	2	+	7	4	2	False
17	2	+	8	4	+	4	1	4	True
18	3	+	6	4	+	6	2	4	False

Ogohlantirish: "+" belgisi chiqarish standart formatda chiqarilishi zarurligini anglatadi.

1.7. 2-topshiriq. Ifodalarni hisoblash. Standart funksiyalardan foydalanish

Mashg'ulot maqsadi:

1. Ifodalarni hisoblashda amal bajarish tartibini o'rganish;
2. Paskal tilida ifodalar yozilishini o'rganish va standart funksiyalarni qo'llash;

Masalaning qo'yilishi:

x berilgan holda $y(x)$ funksiyaning qiymatini toping. Standart funksiyalarni qo'llagan holda $y11=[y]$ ni hisoblang, bu yerda $[]$ sonning butun qismini anglatadi; $y22=[y\pm0,5]$.

Nuqta koordinatalari $x1$ va $y1$ ga bog'liq holda agar nuqta shtrixlangan sohaga tegishli bo'lsa *True* qiymat qabul qiladigan, aks holda *False* qiymat qabul qiladigan ifoda yozing. Berilgan nuqta uchun bu ifodani hisoblang va natijani chiqaring.

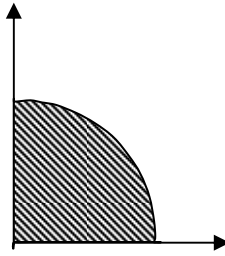
Uslubiy ko'rsatmalar:

1. Paskal tilida mavjud bo'lmagan funksiyalarni mavjud bo'lgan funksiyalar yordamida ifodalab oling.

2. Berilgan nuqtadagi ifodaning qiymatini chiqarish uchun uni *Writeln* operatori ichiga yozing.

3. Dastur namunasi. Quyida berilgan dastur $y(x)=3^{-x+1}\sin(x)$ funksiyani berilgan x ning berilgan qiymati uchun $y11$, $y22$ ni chiqaradi va $(x1,y1)$ koordinatali nuqtaning shtrixlangan sohaga tegishli yoki yo'qligini tekshiradi.(1.1 rasm).

Boshlang'ich ma'lumotlar: $x=-1.5$; $x1=0.5$; $y1=1.2$.



1.1 rasm

```

Var x,y,x1,y1:Real;
Begin
  Writeln('x ni kiriting');
  Readln(x);
  y:=exp((-x+1)*ln(3))*sin(x);
  Writeln('x=',x:8:3,'da y=',y:8:4);
  Writeln('x1, y1 koordinatalarni kiriting');
  Readln(x1,y1);
  Writeln((x1>=0) and (y1>=0) and
    sqr(x)+sqr(y)<=1));
End.

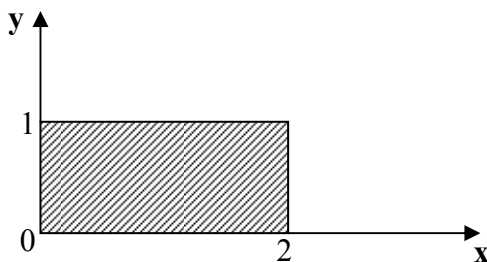
```

Dastur ishi natijasi
 x ni kiriting
 -1.5
 x= -1.500 da y=-15.5494
 y11= -15 y22=-16
 1, y1 koordinatalarni kiriting
 0.5 1.2
 False

Topshiriq variantlari:

1. a) $x=4,741$ da $y = 2^{-x} \sqrt{x + \sqrt[4]{|x|}}$ ni hisoblang

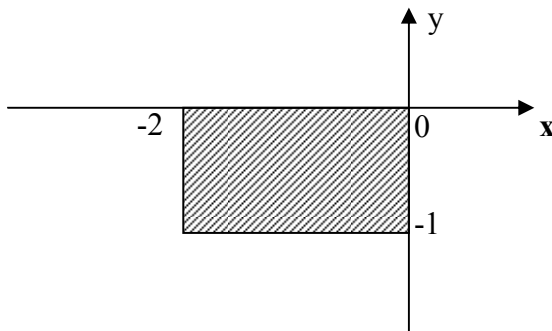
b) $(0,5; 0,5)$ nuqtaning 1.2 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.2 rasm

2. a) $x=2,312$ da $y = \sqrt[3]{e^x - \sin x}$ ni hisoblang

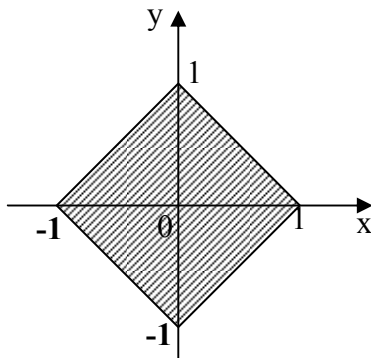
b) $(1,5; 0,5)$ nuqtaning 1.3 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.3 rasm

3. a) $x=12,7409$ da $y = \sqrt{|x-1|} + \sin x$ ni hisoblang

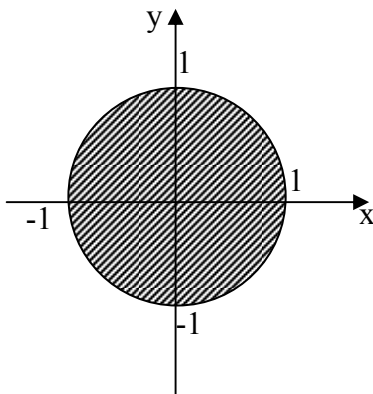
b) $(0,2; 0,9)$ nuqtaning 1.4 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.4 rasm

4. a) $x=32,872$ da $y=x\cos x + \sin^3 x$ ni hisoblang

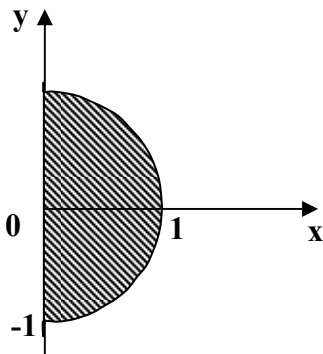
b) $(0,75; -0,3)$ nuqtaning 1.5 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.5 rasm

5. a) $x = -2,6312$ da $y = \operatorname{tg} x + |x|$ ni hisoblang

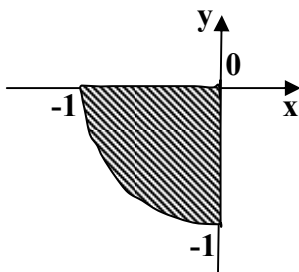
b) $(0,2; 0,45)$ nuqtaning 1.6 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.6 rasm

6. a) $x = -0,387$ da $y = 1 + \frac{1}{x} + \frac{1}{x^2}$ ni hisoblang

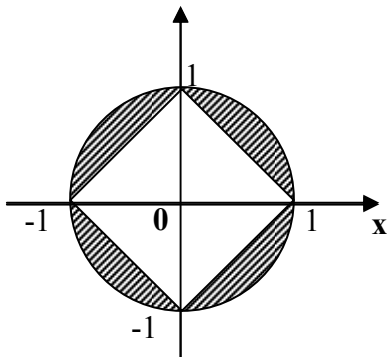
b) $(0,5; -2,5)$ nuqtaning 1.7 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.7 rasm

7. a) $x = 4,352$ da $y = ch|x + 1|$ ni hisoblang

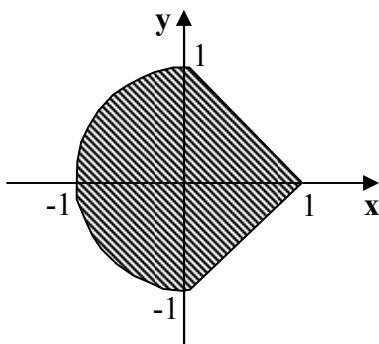
b) $(0,0; 0,0)$ nuqtaning 1.8 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.8 rasm

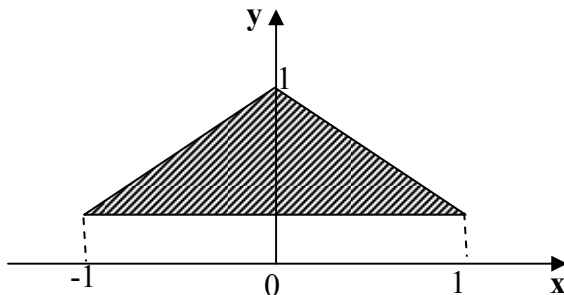
8. a) $x = 0,112$ da $y = \arcsin x + x^2$ ni hisoblang

b) $(1,0; 1,5)$ nuqtaning 1.9 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



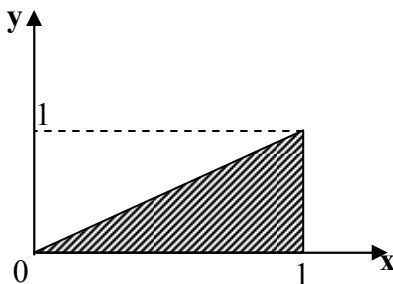
1.9 rasm

9. a) $x = -0,7192$ da $y = \sin \arctg x$ ni hisoblang
 b) $(0,5; -2,5)$ nuqtaning 1.10 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.10 rasm

10. a) $x = -4,4172$ da $y = 5 \arctg x$ ni hisoblang
 b) $(1,5; 0,0)$ nuqtaning 1.11 rasmda tasvirlangan shtrixlangan sohaga tegishli yoki tegishli emasligini aniqlang.



1.11 rasm

2-MAVZU. MURAKKAB OPERATORLAR.

Tarkibli operator, tanlash operatorlari va takrorlash operatorlari murakkab operatorlaridir. Bu operatorlar jismida boshqa operatorlar qatnashishi mumkin.

2.1. Shartli operator. Tarkibli operator.

Shartli operator mumkin bo'lgan ikki ish-harakatdan birini tanlash imkonini beradi, tanlash dastur bajarilishi vaqtida amalga oshiriladi.

Shartli operatorning ikki ko'rinishi mavjud:

1. *if B then S1; (qisqa ko'rinishi);*
2. *if B then S1 else S2; (to'liq ko'rinishi);*

Bu yerda B – mantiqiy turga tegishli ifoda; $S1$, $S2$ – alohida operatorlar yoki *begin* va *end* operatorli qavslar yordamida guruhlangan operatorlardir. Bunday operator tarkibli operator deb nomlanadi.

Birinci ko'rinishdagi shartli operator quyidagi ish-harakatni amalga oshiradi: agar B mantiqiy ifoda *True* (chin) qiymat qabul qilsa *then* maxsus so'zidan keyin turgan $S1$ operatori bajariladi, aks holda, ya'ni B mantiqiy ifoda *False* (yolg'on) qiymat qabul qilsa, $S1$ operatori bajarilmasdan tashlab o'tiladi.

Masalan, *if $x > y$ then $x := 5.1$;* agar dastur ishlash vaqtida x va y o'zgaruvchilarning olgan qiymatlari o'rtasida $x > y$ munosabat o'rinli bo'lsa, u holda $x := 5.1$ ta'minlash operatori bajariladi va x o'zgaruvchi 5.1 qiymat qabul qiladi; agar $x \leq y$, ya'ni $x > y$ mantiqiy ifoda qiymati yolg'on bo'lsa, u holda $x := 5.1$ ta'minlash operatori bajarilmaydi va x ning ilgarigi qiymati o'zgarmaydi.

Shartli operatorning ikkinchi ko'rinishida agar B ifoda *True* qiymat qabul qilsa, u holda $S1$ operator bajarilib, $S2$ operator bajarilmaydi. Agar B ifoda *False* qiymat qabul qilsa, u holda $S2$ operator bajarilib, $S1$ operator bajarilmaydi.

Masalan, *if $x > y$ then $x := 5.1$ else $y := 0$* ; Bu holda, agar x o'zgaruvchi qiymati y o'zgaruvchi qiymatidan katta bo'lsa, ya'ni $x > y$ ifodaning qiymati chin bo'lsa, x ga *5,1* qiymat ta'minlanadi, y esa o'zgarishsiz qoladi. Agar $x \leq y$, ya'ni $x > y$ ifodaning qiymati yolg'on bo'lsa, u holda y ga *0* ta'minlanadi, x esa o'zgarishsiz qoladi.

Shartli operatorlarda *S1* yoki *S2* o'rnida bitta operator kelishini yodda tutish zarur, agar bittadan ortiq operator qatnashishi zarur bo'lsa, tarkibli operatorlardan foydalaniladi. Masalan, $a < 0$ da x va y butun turga tegishli o'zgaruvchilarga mos ravishda *5* va *23* qiymatlar, aks holda nollar ta'minlanishi kerak bo'lsin. Bu hol uchun shartli operator quyidagicha yoziladi:

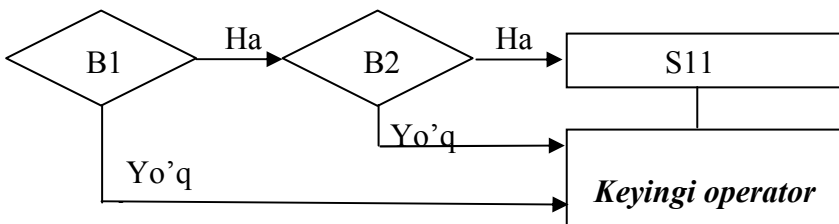
If $a < 0$ then begin $x := 5$; $y := 23$ end
else begin $x := 0$; $y := 0$ end;

Shartli operatorlarda *then* va *else* maxsus so'zlaridan keyin keladigan operatorning turiga chegara qo'yilmagan. Ular ixtiyoriy turdagi operator bo'lishi mumkin. *then* yoki *else* dan keyin yana shartli operator keladigan hollarni qaraymiz. Uchrashi mumkin bo'lgan variantlar ikki xil:

1. Qisqa ko'rinishli *If B1 then S1* ko'rinishdagi shartli operatorlarda *S1* operator yana shartli operator bo'lishi mumkin, u holda quyidagi tuzilmaga ega bo'lamiz:

If B1 then If B2 then S11

Uning blok-sxemasini 2.1 rasmda tasvirlangan.

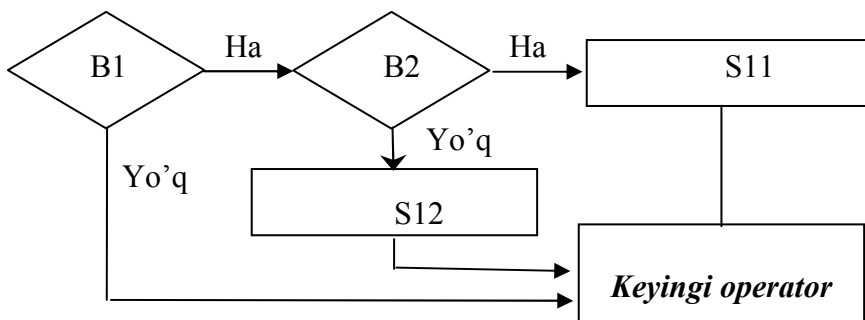


2.1 rasm

Shartli operatorning qisqa ko'rinishidagi *S1* operator to'liq ko'rinishdagi shartli operator bo'lishi mumkin, u holda quyidagi tuzilmaga ega bo'lamiz:

If B1 then if B2 then S11 else S12;

Bu tuzilmaning blok-sxemasi 2.2 rasmda tasvirlangan.



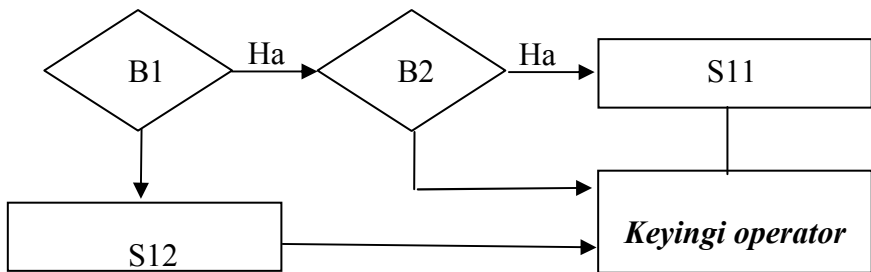
2.2 rasm

Bu holda shunday savol tug'ilishi mumkin: *else* so'zi qaysi *then* ga mos keladi. Paskal tilida bir qiymatlilikni ta'minlash uchun har bir *else* so'ziga o'zidan oldingi *then* qo'yilishi kelishib olingan.

2. To'liq *If B1 then S1 else S2* ko'rinishdagi shartli operatorlarda *S1* va *S2* operatorlar sifatida qisqa ko'rinishdagi shartli operator yoki to'liq ko'rinishdagi shartli operator mos kelishi mumkin. Quyidagi holatni qaraymiz: *S1* – birinchi ko'rinishdagi shartli operator; *S2* – esa shartli operator bo'lmasin. Quyidagi tuzilmaga ega bo'lamiz:

If B1 then begin if B2 then S11 end else S2;

Bu holda *S1* operatorni operatorli qavslarga olish zarurligini osongina anglash mumkin. Operatorli qavslarning tushib qolishi 2.2 rasmda ko'rsatilgan tuzilmaga olib keladi. Yuqoridagi holatning blok-sxemasi 2.3 rasmda tasvirlangan.



2.3. rasm

Operatorli qavslar ta'sir ko'rsatadigan dastur qismidan misol keltiramiz:

.....

if A> then begin

if C<D then x:=2

else if C>D then x:=3

end

else x:=4;

Bu holda $x:=4$ ta'minlash operatori $A>B$ ifoda *false* qiymat qabul qilganida bajariladi. Agar operatorli qavslarni olib tashlasak, ya'ni yuqoridagi konstruksiyadan *begin* va *end* ni olib tashlasak, u holda $x:=4$ ta'minlash operatori $A>B$ ifodaning qiymati *true* bo'lib $C<D$ ifodaning qiymati *false*, $C>D$ ifodaning qiymati ham *false*, ya'ni $C=D$ bo'lgan holdagina bajariladi.

Dasturda operatorlarning birin-ketin bajarilish tartibini buzib, belgilangan operatorga o'tishni amalga oshirish mumkin. Buning uchun *Go To <nishon>* ko'rinishga ega bo'lgan oddiy shartsiz o'tish operatori qo'llaniladi.

Dasturning har qanday operatorini uning oldiga butun son bilan ifodalanuvchi va operatoridan ikki nuqta (:) bilan

ajritiladigan nishon bilan belgilash mumkin. Masalan, quyidagi ta'minlash operatori 32 nishon bilan belgilanadi. 32: $x:=5$;

Paskal tilida dasturda uchrovchi hamma nishonlar nishonlar bo'limida tasvirlangan bo'lishi kerak. Yuqorida keltirilgan misolda nishonlar bo'limi *Label 32*; ko'rinishida bo'ladi.

2.2. Takrorlash operatorlari

Ko'p hollarda masalani yechish algoritmidagi ba'zi qadamlarni ko'p marta takrorlash zaruriyati tug'iladi. Paskal tilida bunday takrorlanishlarni boshqarish uchun algoritmlarni yozishda uch xil ko'rinishli takrorlash operatorlari ishlatiladi: parametrli, old shartli va so'ng shartli takrorlash operatorlari. Agar takrorlanishlar soni ilgaridan ma'lum bo'lsa, parametrli takrorlash operatorini qo'llash qulay.

Parametrli takrorlash operatorining umumiy ko'rinishi quyidagicha:

For $P:=PB$ to PO do A ; yoki

For $P:=PO$ downto PB do A ;

Bu yerda P – takrorlash parametri; PB – takrorlash parametrining boshlang'ich qiymatini, PO esa takrorlash parametrining oxirgi qiymatini beradigan ifodalar; A – oddiy yoki tarkibliy operator.

Takrorlash parametrining turini takrorlanishning oxirgi va boshlang'ich qiymatlarini beradigan ifodalar (PO va PB) turi bilan mos tushishi shart. Ilgari ko'rilgan standart turlar ichida haqiqiy tur takrorlash parametri turi sifatida foydalanilishi mumkin emas.

For $P:=PB$ to PO do A ko'rinishidagi parametrli takrorlash operatorining bajarilishi $P \leq PO$ shartni tekshirishdan boshlanadi (Ikkinchi ko'rinishdagisi uchun $P \geq PO$ shartni tekshirishdan boshlanadi). Agar shart noto'g'ri bo'lsa, A operator bajarilmay, boshqarish navbatidagi

operatorga uzatiladi. Agar shart to'g'ri bo'lsa, u holda A operator bajariladi, so'ngra takrorlash parametriga bitta keyingi qiymat $P := Succ(P)$ (to takrorlanishi uchun) yoki bitta oldingi qiymat $P := Pred(P)$ (downto takrorlanishi uchun) ta'minlanadi. Shu tartibda butun jarayon takrorlanadi. Takrorlash parametrining turi butun bo'lgan holda, takrorlashning har bir qadamida uning qiymati bittaga oshiriladi (downto - sikl uchun esa kamaytiriladi).

Parametrli takrorlash operatorida 1 yoki -1 dan farqli qadam berish mumkin emas.

O'zgaruvchilar bo'limi: *Var s,i:integer; sim: char; l:Boolean;* bo'lgan hol uchun takrorlash operatorlariga misollar keltiramiz:

- 1) $s:=0$; For $i:=5$ to 7 do $s:=s+1$;
- 2) $s:=0$; For $i:=10$ downto 6 do $s:=s+1$;
- 3) $s:=0$; For $sim:='A'$ to $'D'$ do $s:=s+1$;
- 4) $s:=0$; For $l:=False$ to $True$ do $s:=s+1$;

Birinchi misolda parametrli takrorlash operatorining ishi tugallangach $s=3$, ikkinchi misolda $s=5$, uchinchi misolda $s=4$, to'rtinchi misolda $s=2$ qiymat hosil bo'ladi.

Agar takrorlash operatorida bir necha operatorlar bajarilishi kerak bo'lsa, u holda tarkibli operatoridan foydalaniladi. Masalan,

```
For k:=10 to 13 do
  begin
    l:=k*k;
    Writeln('l=',l:3);
  end;
```

Natijada quyidagi sonlar ustuni hosil bo'ladi:

```
l:=100
l:=121
l:=144
l:=169
```

Parametrli takrorlash operatori uchun ba'zi cheklashlar mavjud:

- operator ichida takrorlash parametri qiymatini, takrorlash parametrining boshlang'ich va oxirgi qiymatlarini o'zgartirish mumkin emas;

- operator ichiga faqat uning boshidan kirish mumkin.

Shunday qilib, parametrli takrorlash operatori ma'lum bir turida parametrni bir birlikka o'zgartirib, takrorlanuvchi jarayonni tashkil qilish imkonini beradi. Agar qadam birdan farq qilsa, yoki takrorlanishlar soni ilgaridan ma'lum bo'lmasa, old shartli yoki so'ng shartli takrorlash operatorlaridan birini qo'llash zarur.

Old shartli takrorlash operatorining umumiy ko'rinishi quyidagicha:

While B do A;

Bu yerda B – mantiqiy turga tegishli ifoda, A – oddiy yoki tarkibli operator.

Operatorning bajarilishi B ifodani hisoblash bilan boshlanadi. Agar u *true* qiymat qabul qilsa, A operator bajariladi, so'ngra bu jarayon takrorlanib, B ifodaning qiymati *false* qiymat qabul qilgunga qadar davom etadi. *false* qiymat qabul qilingan zahoti boshqarish navbatdagi operatorga uzatiladi. Agar mantiqiy turga tegishli ifoda takrorlanishga birinchi kirishdayoq yolg'on qiymat qabul qilsa, u holda A operator bir marta ham bajarilmaydi. Ko'rinib turibdiki, takrorlanish ichidagi operatorlardan biri ifodaning qiymatiga ta'sir etishi kerak, aks holda takrorlanishlar jarayoni cheksiz davom etadi. Old shartli takrorlash operatorini qo'llashga misol qaraymiz:

Program Test;

Var k:integer;

Begin

k:=0;

```

While  $k \leq 10$  do
  Begin
     $k := k + 2$ ;
    Write( $k:5$ );
  End;

```

End;

Natijada 2 dan 12 gacha bo'lgan butun sonlar chiqariladi. Ta'kidlaymizki, 12 soni chiqarilgach, $k \leq 10$ shart tekshiriladi va takrorlash jarayoni tugatiladi. Chiqarish natijasi quyidagicha bo'ladi:

2 4 6 8 10 12

So'ng shartli takrorlash operatori old shartli takrorlash operatoriga o'xshash, ammo shart sikl ichidagi operatorlar bajarilgach hisoblanadi va tekshiriladi. So'ng shartli takrorlash operatorining umumiy ko'rinishi quyidagicha:

Repeat A1; A2;...An Until B;

So'ng shartli takrorlash operatorining bajarilishi takrorlanish jismi ichidagi operatorlarning bajarilishi bilan boshlanadi. So'ngra B ifoda hisoblanadi, agar rost qiymat olinsa, u holda takrorlanishdan chiqiladi. Agar ifodaning qiymati yolg'on bo'lsa, u holda $A1; A2;...An$ operatorlarning bajarilishi takrorlanadi, so'ng yana B ifoda tekshiriladi. Old shartli takrorlash operatoridan farqli ravishda, so'ng shartli takrorlanishdan chiqish B ifoda rost qiymat qabul qilganida amalga oshirishini ta'kidlab o'tamiz.

Yuqoridagi misolda (*Test* sarlavhali dasturda) bajarilgan ish-harakatni so'ng shartli takrorlash operatoridan foydalangan holda qayta tuzamiz:

```

Program Test1;
Var k:integer;
Begin
  k:=0;
  Repeat

```

```

k:=k+2;
Write(k:3);
Until k>10;

```

End;

Bu dasturda ham natija sifatida 2 dan 12 gacha bo'lgan sonlar chiqariladi.

2.3. Savollar va mashqlar

1. Quyida keltirilgan operatorlardan qaysilari noto'g'ri va nima uchun?

a) *if* $a < b$ *then* $a := a * a$ *else* $b := b * b$;

b) *if* x *and* y *then* $s := s + 1$; *else* $s := s - 1$;

c) *if* $k < > m$ *then* $k := m$;

d) 12: *if* $(a < b)$ *or* c *then* $c := \text{false}$;

e) *if* $(a = b)$ *and* p *then* $p := p + 10.5$;

2. Agar $a = 0,5$, $b = -1,7$ bo'lsa, *if* $a < b$ *then* $a := b$ *else* $b := a$; shartli operator bajarilishi natijasida a va b o'zgaruvchilar qanday qiymat qabul qiladi?

3. *if* $x < y$ *then* $\text{max} := y$ *else* $\text{max} := x$; operatorning bajarilishi natijasida qanday masala yechiladi?

4. Shartli operator yordamida quyidagi ish-harakatlarni yozing:

a) x va y haqiqiy turga tegishli o'zgaruvchilarning qiymati kichigini nol bilan almashtiring, agar ular teng bo'lsa ikkalasini ham nol bilan almashtiring.

b) Butun turga tegishli x , y , z o'zgaruvchilarning qiymatlari orasidan eng kattasini 0,3 ga kamaytiring.

5. Hisoblang: $y = \begin{cases} x^2 + 1, & \text{agar } x < 0; \\ x - 2,1, & \text{agar } x > \frac{\pi}{2}; \\ \sin x, & \text{agar qolgan hollarda} \end{cases}$

6. a , b , c butun turga tegishli o'zgaruvchilarning qiymatlarini $a \geq b \geq c$ tengsizlik o'rinli bo'ladigan qilib o'rin almashtiring.

7. Agar x va y o'zgaruvchilar teng bo'lsa, B mantiqiy turga tegishli o'zgaruvchiga *true* qiymat ta'minlang, aks holda *false* qiymat ta'minlang.

8. Quyidagi operatorlar bajarilgach f o'zgaruvchi qanday qiymat qabul qiladi?

$i:=; f:=2; \text{ While } i < 6 \text{ do } i:=i+1; f:=f*i;$

$i:=; f:=2; \text{ While } i < 6 \text{ do begin } i:=i+1; f:=f*i; \text{ end;}$

9. Gornersxemasini bo'yicha hisoblang:

a) $y = a^7 + 2a^6 + 3a^5 + 4a^4 + 5a^3 + 6a^2 + 7a + 8;$

b) $y = 8a^7 + 7a^6 + 6a^5 + 5a^4 + 4a^3 + 3a^2 + 2a + 1;$

10. 30 ta belgidan tashkil topgan matn berilgan. a va b belgilar bu matnda necha marta uchrashini hisoblang.

11. $y = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots + c$ ifodaning qiymatini $\zeta = 10^{-4}$ aniqlik bilan hisoblash dasturini tuzing.

12. $y = \sin(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots + c$ ifodaning qiymatini $\zeta = 10^{-4}$ aniqlik bilan hisoblash dasturini tuzing.

2.4. 1-topshiriq. Tenglama va tengsizliklarni yechish.

Topshiriq maqsadi.

1. Dasturda shartli operatorni qo'llashni o'rganish.

2. Shartli operator qo'llaniladigan misollar yechimi bilan tanishish.

Masalaning qo'yilishi.

Tenglama (tenglamalar sistemasi, tengsizlik, tengsizliklar sistemasi) yechimini hisoblaydigan dastur tuzing. Boshlang'ich

ma'lumotlar va natijalarni matematikada ishlatiladigan tarzda chiqaring.

Hisobot mazmuni.

1. Masalaning qo'yilishi.
2. Dastur matni.
3. Dasturni sozlash rejasi, testlar ro'yxati, testlar yordamida sinash natijalari.
4. Yo'l qo'yilgan xatolar tahlili.

Uslubiy ko'rsatmalar.

1. Dastur boshlang'ich ma'lumotlarning hamma qiymatlari uchun to'g'ri ishlashi kerak.

2. Dasturning hamma tarmoqlari testlar yordamida sinovdan o'tkazilishi zarur.

3. Tenglama yoki tengsizlikning yechimi yo'q yoki cheksiz yechimga ega bo'lganda, unga mos ma'lumot beruvchi matn chiqarilishi kerak. Masalan, "tenglama yechimga ega emas" yoki "har qanday x tenglamaga yechim bo'ladi" va h.k.

Topshiriq variantlari.

1. $ax^2+b=0$
2. $ax^2+bx+c=0$
3. $ax^2+b>0$
4. $ax^2+b\leq 0$
5. $ax^2+b+c>0$
6. $ax^2+bx+c\leq 0$
7.
$$\begin{cases} a_1x + b_1y + c_1 = 0 \\ a_2x + b_2y + c_2 = 0 \end{cases}$$
8.
$$\begin{cases} a_1x + b_1 < 0 \\ a_2x + b_2y < 0 \end{cases}$$

$$9. \begin{cases} a_1x + b_1 < 0 \\ a_2x + b_2 < 0 \end{cases}$$

2.5. 2-topshiriq. Funksiya qiymatlari javdalini tuzish.

Topshiriq maqsadi.

1. Parametrli takrorlash operatorini qo'llashni o'rganish.
2. Dasturni optimallashtirish usullari bilan tanishish.

Masalaning qo'yilishi.

$f(x)$ funksiyaning $[a, b]$ oraliqqa tegishli $x_i = a + ih$ nuqtalardagi qiymatlarini hisoblash dasturini tuzing. Bu yerda $h = (b - a)/m$, m – berilgan butun son.

Hisobot mazmuni.

1. Masalaning qo'yilishi.
2. Dastur matni
3. Natijalar jadvali.
4. Yo'l qo'yilgan xatolar tahlili.

Uslubiy ko'rsatmalar.

1. x ning qiymatlari va funksiyaning unga mos keluvchi qiymatlari uchun oddiy o'zgaruvlar qo'llanilishi kerak.
2. h qadamning qiymati bir marta hisoblanishi kerak.
3. x o'zgaruvchining qiymatini o'zgartirishda dastur bajarilish vaqtini uzaytiradigan $x := a + i * h$ ko'paytirish amalini emas, $x := x + h$ ta'minlash operatorini qo'llash kerak.

Topshiriq variantlari.

№	$f(x)$ funksiya	Parametrlar		M
		a	b	
1	$x - \sin(x)$	0	$\pi/2$	10
2	$\sin(x)$	$\pi/4$	$\pi/2$	15

3	$\cos(x)$	$\pi/3$	$2\pi/3$	20
4	$tg(x)$	0	$\pi/4$	10
5	$ctg(x)$	$\pi/4$	$\pi/2$	15
6	$\arcsin(x)$	0	1	20
7	$\arccos(x)$	0,5	1	10
8	$\arctg(x)$	2	7	15
9	$\sin(x)-\cos(x)$	0	$\pi/2$	20
10	$x\sin(x)$	0	3π	10
11	$\sin(1/x)$	$\pi/8$	$2/\pi$	15
12	$\cos(1/x)$	$\pi/4$	$4/\pi$	20
13	$\sin(x^2)$	$\pi/6$	$2\pi/3$	10
14	$\cos(x^2)$	$\pi/3$	$3\pi/2$	15
15	$\sin(x)+tg(x)$	0	$\pi/4$	20
16	$\cos(x)+tg(x)$	$\pi/4$	$\pi/2$	10
17	$tg(x/2)$	0	$2\pi/3$	15
18	$tg(x/2)+\cos(x)$	$\pi/2$	π	20
19	$ctg(x/3)+\sin(x)$	$\pi/4$	$\pi/2$	10
20	$\sin(x/4)/2$	$\pi/2$	π	15

2.6. 3-topshiriq. Dasturda takrorlanishni tashkillashtirish.

Topshiriq maqsadi.

1. takrorlash operatorini qo'llashni o'rganish.
2. iteratsion jarayonlar bilan tanishish

Masalaning qo'yilishi.

Takrorlash operatorini qo'llab variantda ko'rsatilgan elementlar yig'indisini toping. Natijani mos keluvchi sarlavha bilan birgalikda chiqaring.

Hisobot mazmuni.

1. masalaning qo'yilishi.

2. dastur matni.
3. dasturni sozlash rejasi, testlar ro'yxati, testlar asosida sinash natijalari.
4. berilgan variantni yechish natijalari.
5. yo'l qo'yilgan xatolar tahlili.

Uslubiy ko'rsatmalar.

1. qator hadlarining yig'indisini aniqlashda qatorning keyingi hadini topish uchun rekurrent formulani qo'llash kerak.

Masalan, umumiy hadi a_n bo'lgan qator yig'indisini $\xi=10^{-4}$ aniqlik bilan topish talab qilingan bo'lsin. Rekurrent formulani olish uchun qatorning keyingi hadini joriy hadga bo'lamiz:

$$\frac{a_{n+1}}{a_n} = \frac{2((n+1)!)^2 3(2n)!}{3(2n+2)!2(n!)^2} = \frac{n+1}{2(2n+1)}, \text{ bu yerdan}$$

$$a_{n+1} = \frac{a_n (n+1)}{2(2n+1)}$$

$a_n < E$ tengsizlik bajarilgan holda talab qilingan aniqlik erishilgan hisoblanadi.

2. Dasturga misol.

Const e=0.1E-3;

Var n:integer; an,summa: real;

Begin

Summa:=0; n:=1; an:=1/3;

While an>e do

Begin

Summa:=summa+an;

n:=n+1;

an:=an(n+1)/2/(2*n+1)*

End;

Writeln('Summa= ',summa, ' qatorning oxirgi hadi= ', an);

End.

Dastur ishining natijalari:

$Summa=4.727439E-01$ qatorning oxirgi hadi
 $=4.113534E-05$

Topshiriq variantlari:

1. 100 dan kichik va 4 ga karrali bo'lgan musbat butun sonlar yig'indisini toping.

2. 100 dan kichik musbat juft sonlar yig'indisini toping.

3. 200 dan kichik musbat toq sonlar yig'indisini toping.

4. 20 dan katta, 100 dan kichik va 3 ga karrali sonlar yig'indisini toping.

5. Umumiy hadi $a_n = \frac{(-1)^{n-1}}{n^n}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

6. Umumiy hadi $a_n = \frac{1}{2^n} + \frac{1}{3^n}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

7. Umumiy hadi $a_n = \frac{2n-1}{2^n}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

8. Umumiy hadi $a_n = \frac{1}{(3n-2)(3n+1)}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

9. Umumiy hadi $a_n = \frac{10^n}{n!}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

10. Umumiy hadi $a_n = \frac{1}{(3n-2)(3n+1)}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

11. Umumiy hadi $a_n = \frac{n!}{n^n}$ bo'lgan qator yig'indisini $\xi=10^{-2}$ aniqlik bilan hisoblang.

12. Umumiy hadi $a_n = \frac{2^n n!}{(n^n)!}$ bo'lgan qator yig'indisini $\xi=10^{-2}$ aniqlik bilan hisoblang.

13. Umumiy hadi $a_n = \frac{3^n n!}{(3n)!}$ bo'lgan qator yig'indisini $\xi=10^{-3}$ aniqlik bilan hisoblang.

14. Umumiy hadi $a_n = \frac{n!}{(3n^n)}$ bo'lgan qator yig'indisini $\xi=10^{-2}$ aniqlik bilan hisoblang.

15. Umumiy hadi $a_n = \frac{(n!)^2}{(2^{n^2})!}$ bo'lgan qator yig'indisini $\xi=10^{-2}$ aniqlik bilan hisoblang.

16. Umumiy hadi $a_n = \frac{\ln(n!)}{n^2}$ bo'lgan qatorning 10 ta hadi yig'indisini hisoblang.

17. Umumiy hadi $a_n = \frac{n^{\ln n}}{(\ln n)^2}$ bo'lgan qatorning 15 ta hadi yig'indisini hisoblang.

18. Umumiy hadi $a_n = \frac{n!}{n^{\sqrt{n}}}$ bo'lgan qatorning 12 ta hadi yig'indisini hisoblang.

19. Umumiy hadi $a_n = \frac{n!}{n^{\sqrt{n}}}$ bo'lgan qatorning 7 ta hadi yig'indisini hisoblang.

20. Umumiy hadi $a_n = \frac{n!}{n^{\sqrt{n}}}$ bo'lgan qatorning 9 ta hadi yig'indisini hisoblang.

2.7. 4-topshiriq. Belgili ma'lumotlarni qayta ishlashda murakkab operatorlardan foydalanish.

Topshiriq maqsadi.

1. takrorlash operatorlaridan foydalanish ko'nikmasini mustahkamlash.
2. belgili ma'lumotlar bilan ishlash ko'nikmasini olish.
3. ixtiyoriy sanoq sistemasidagi sonni o'nlik sanoq sistemasiga o'tkazish algoritmini o'rganish.

Masalaning qo'yilishi.

P asosli sanoq sistemasidagi son yozuvini bildiruvchi belgilar ketma-ketligini kiriting. Kiritiladigan simvollarning soni N ning qiymati variantda beriladi. Bu sonni o'nlik sanoq sistemasiga o'tkazish va u M ga (M ning qiymati variantda beriladi) karralimi yoki yo'qligini aniqlash talab qilinadi. Boshlang'ich sonni va uning o'nlik sanoq sistemasiga o'tkazilgan ko'rinishini chiqaring. Agar u M ga karrali bo'lsa, *true* so'zini chiqaring, aks holda *false* so'zini chiqaring.

Hisobot mazmuni

1. masalaning qo'yilishi.
2. o'nlik sanoq sistemasiga o'tkazish algoritmi.
3. dastur matni va uning bajarilishi natijalari.
4. yo'l qo'yilgan xatolar tahlili.

Uslubiy ko'rsatmalar.

Topshiriqni bajarish uchun, dastlab, quyida berilgan dasturni tahlil qilib chiqish zarur. Bu dastur berilgan sonning 16 lik sanoq sistemasidagi yozuvini bildiruvchi uchta simvolni kiritadi, uni o'nlik sanoq sistemasiga o'tkazadi va pechatga chiqaradi.

```
Var C:Char;  
    i, m: integer'  
Begin  
    Readln;  
    Write('=>');  
    M:=0;  
    For i:=1 to 3 do  
        Begin  
            Read(S);  
            Write(S);  
            If (S>=0) and (S<=9) then m:=m*16+ord(s)-ord('0')  
                else m:=m*16+ord(s)-ord('A')+10;  
        end;  
    Writeln;  
    Writeln('10 lik sanoq sistemasidagi son',M:5);  
End.
```

Yuqoridagi dastur ishi natijasi:

=> FFF

FFF

10 lik sanoq sistemasidagi son 4095.

Topshiriq variantlari:

№	N	P	M	№	N	P	M
1	3	11	3	6	3	12	10
2	5	8	4	7	3	11	11
3	6	7	5	8	4	9	7
4	8	6	8	9	5	2	2
5	9	4	9	10	3	13	9

3-MAVZU. SANOQLI VA SKALYAR TURLAR

Paskal dasturlash tilida biz ko'rgan standart turlardan farqli yangi turlar kiritish (yaratish) imkoniyatlari mavjud. Yangi turlarning xarakteristikallari dasturchi tomonidan aniqlanib, ular dasturning turlar bo'limida tavsiflanadi. Quyida biz yangi turlar yaratish imkoniyatlari, bu turlarga tegishli qiymatlar ustida amallar bajarish tartiblari bilan tanishamiz.

3.1. Sanoqli skalyar turlar.

Ma'lumotlarning sanoqli skalyar turi unga tegishli o'zgaruvchilar qabul qilishi mumkin bo'lgan qiymatlarning (ob'ektlarning) ro'yxatini berish (qiymatlarni sanab o'tish) orqali aniqlanadi. Har bir ob'yekt o'z ismiga ega bo'ladi. Sonlar, mantiqiy va simvolli o'zgarmaslar sanoqli tur ob'ektlari sifatida ishlatilishi mumkin emas.

Sanoqli turlarni ishlatish programmaning ko'rgazmalilik (наглядность) xususiyatini oshiradi va o'zgaruvchilarning ma'lum bir qiymatlarni qabul qilishi mumkin yoki yo'qligini avtomatik ravishda nazorat qilish imkoniyatlarini yaratadi.

Masalan,

Type day = (mo, tu, we, th, fr, sa, su);

var d1, d2, d3: day;

Bu yerda yangi *day* nomli tur aniqlangan. *D1, D2, D3* o'zgaruvchilar *day* turga mansub bo'lib, programmada ular

yuqorida sanab o'tilgan 7 qiymatdan faqat birini qabul qilishi mumkin.

Bir turda keltirilgan qiymat (ob'yekt) boshqa bir turni tavsiflashda ishlatilishi mumkin emas. Masalan, turlarning *Type t1 = (one, two, three, four)*; *type t2 = (four, five, six)*; kabi berilishi xato, chunki *four* nomli ob'yekt ikki marta uchrayapti.

Bir turdagi o'zgaruvchiga ikkinchi bir turga tegishli qiymatni ta'minlash mumkin emas. Masalan,

Type name = (jon, tom, nick);

Type color = (red, blue, black);

Var x,y,z: name; c1, c2, c3: color;

a) *c1: = red; x: = jon; n: = tom; c3: = blue* deb yozish mumkin.

b) *z: = black; c2: = nick;* kabi yozish mumkin emas.

Sanoqli turlarni tavsiflashda ob'yektlarni sanab o'tish tartibi muhim ahamiyatga ega, chunki ular uchun nisbat amallari o'rinli. Yuqorida aniqlangan turlar uchun

jon < tom < nick

red < blue < black

Sanoqli skalyar turga tegishli qiymatlar uchun *pred, succ, ord* standart funksiyalari aniqlangan va xuddi oddiy turlardagi kabi natijalar hosil qiladi. Masalan, *color* turi uchun *succ(red)* funksiyaning natijasi *blue* qiymatni, *pred(black)* funksiya natijasi ham *blue* qiymatni beradi. *ord(blue)* funksiya qiymati *1* ga teng, chunki ob'yektlarni nomerlash noldan boshlanadi.

Misol. Quyidagi programma *FT* (familiyalar) o'zgaruvchining berilgan qiymatlari bo'yicha mos ismlarni IT o'zgaruvchiga ta'minlaydi:

Program PT;

type ism = (ali, soli, bek, gul);

famil = (otaev, mirzaev, nurov, alimova);

var ft, p1: famil; it, p2: ism;

```

.....
p1: = otaev; p2: = ali;
while p1 < ft do
  begin
    p1: = succ (p1); p2: = succ (p2)
  end
it: = p2;
.....

```

Sanoqli skalyar turga tegishli qiymatlarni tashqi qurilmalardan bevosita kiritish (chiqarish) mumkin emas. (Bu ishni variant operatori orqali bajarish mumkin).

3.2. Cheklangan turlar.

Ma'lumki, standart va sanoqli skalyar turlarning qiymatlar to'plami tartiblangan va cheklidir. Agar dasturchi biron bir tur (asosiy tur) ning qiymatlar diapazonini (to'plamini) yanada toraytirishni hohlasa, cheklangan tur tushunchasidan foydalanish mumkin. Masalan, quyidagi turlar aniqlangan bo'lsin:

```

type day = (mo, tu, we, th, fr, sa, su);
nom = 10..25;
sss = 'c..'x';
wd = fr..su;

```

bu yerda *nom*, *sss*, *wd* lar cheklangan turlar bo'lib, *nom* uchun asosiy tur butun tur, *sss* uchun asosiy tur simvolli tur, *wd* uchun esa sanoqli tur *day* dir. Cheklangan turlar quyi va yuqori chegaralarni ko'rsatish yo'li bilan beriladi. Bu diapazonga kirmaydigan qiymatni shu turdagi o'zgaruvchiga ta'minlash xatolikka olib keladi.

Cheklangan turlar uchun asosiy turga qo'llaniladigan hamma amallar va standart funksiyalar o'rinlidir.

3.3. Tanlov operatori

Shartli operator programma bajarilishining mumkin bo'lgan 2 tarmog'idan birini tanlashga imkon beradi. Agar bir - biriga zid bo'lgan ko'p (2 tadan ortiq) shartlarni tekshirish zarur bo'lsa, u holda variant operatoridan foydalanish qulay.

Tanlov operatorining umumiy ko'rinishi quyidagicha:

Case <selektor> of

<variant nishoni(lari)> : <operator>;

.....

<variant nishoni(lari)> : <operator>;

end;

Bu yerda *<selektor>* - skalyar turdagi (haqiqiy tur bo'lishi mumkin emas) ifoda, *<operator>* esa oddiy yoki murakkab operator. Tanlov operatorining bajarilishi *<selektor>* ning qiymatini topishdan boshlanadi. Hosil bo'lgan qiymat *<variant nishoni(lari)>* orasidan izlanadi. Agar u mavjud bo'lsa, shu nishondan keyin turgan *<operator>* bajariladi va harakat *case* operatoridan keyingi operatorga uzatiladi.

Agar selektorning qiymati hech qanday nishon bilan mos tushmasa, u holda bu holat aniqlanmagan deb hisoblanadi va mashinaning bu holatga javobi dasturlash tizimining amalga oshirilgan variantiga bog'liq bo'ladi.

Tanlov operatori qo'llanilgan dastur qaraymiz:

Program War;

Var sum: 'A'..'G'; i,j:1..9;

Begin

Write('A dan F gacha bo'lgan harf kiriting');

Readln(sum);

Write('1 dan 9 gacha oraliqda butun son kiriting');

Readln(i);

Case sum of

'A','B','C','E': Writeln(sum:3,' harf ham lotin alifbosida, ham kirill

```

        alifbosida mavjud');
        'D', 'G', 'F': Writeln(sum:3, ' harf faqat lotin
alifbosida mavjud');
    Case i of
        2,4,8 Begin
            Write(i:2, ' 2 ga karrali');
            J:=round(i/2);
            Writeln('i/2=',j:2);
        End;
        3,9: Begin
            Write(i:2, ' 3 ga karrali');
            Writeln('i/3=', Round(i/3):2);
        End;
        6: Begin
            Write(i:2, ' 2 va 3 ga karrali');
            Writeln('i/6=',Round(i/2/3):7);
        End;
        1,5,7: Write(i:2, 'Oddiy son');
    end;
end.

```

Sum va *i* o'zgaruvchilar cheklangan tur o'zgaruvchisi sifatida tavsiflangan. Aniq qiymat *Readln* operatori yordamida tanlov operatori bajarilgunga qadar kiritiladi. Hech qanday parametr ga ega bo'lmagan *Readln* operatori esa bitta bo'sh satr tashlab o'tish imkonini beradi.

Dasturning tanlov operatori qatnashgan birinchi qismida *sum* o'zgaruvchi qiymatidan bog'liq holda kiritilgan belgining lotin alifbosiga yoki kirill va lotin alifbolarga tegishliligi haqida izoh chiqariladi.

Dasturning tanlov operatori qatnashgan ikkinchi qismida *i* butun turga tegishli qiymat bo'yicha kiritilgan son uning karraliligi haqidagi izoh bilan chiqariladi. Bu qismda kiritishni boshqarishning har xil usullari ko'rsatilgan.

Round standart funksiyasiga murojaat qilish shart, chunki bo'lish amali bajarilganda har doim haqiqiy turga tegishli qiymat olinadi.

Tanlov operatori sanoqli skalyar turga tegishli qiymatlarni kiritish va chiqarishga qulay. Masalan, quyidagi dasturda *Color* sanoqli turi ob'yektlari ro'yxati bo'yicha ob'jektning tartib nomeri kiritiladi. *Case* operatori *clr* o'zgaruvchiga mos qiymatni ta'minlaydi. *clr* o'zgaruvchining qiymati ham *case* operatori yordamida chiqariladi.

```
Program cas;  
type color=(qizil, yashil, qora);  
var x:integer; clr:color;  
begin  
  write('==> ');  
  readln(x);  
  case x of  
    0: clr:=qizil;  
    1: clr:=yashil;  
    2: clr:=qora;  
  end;  
  write('rang- ');  
  case clr of  
    qizil: writeln('qizil');  
    yashil: writeln('yashil');  
    qora: writeln('qora');  
  end;  
end.
```

Dastur natijasi:

==> 2

Rang - Qora

3.4. Savollar va mashqlar

1. 20 dan 35 gacha butun qiymat qabul qiladigan i o'zgaruvchini cheklangan tur o'zgaruvchisi kabi tavsiflang.

2. [2,5] kesmada haqiqiy qiymat qabul qiladigan o'zgaruvchini cheklangan tur cheklangan tur o'zgaruvchisi sifatida tavsiflash mumkinmi?

3. Qiymati hafta kunlari nomlari bo'lgan cheklangan tur o'zgaruvchisini tavsiflang.

4. Quyidagi tavsifdagi xatolikni aniqlang.

Const a=2..30; b:='b'; pi=3.14159;

Type Kesma=(1..20);

Shahar=(Toshkent; Samarqand; Qarshi; Guliston);

Harflar=('A', 'B', 'C', 'D');

Raqamlar=(0,1,2,3,4,5,6,7,8,9);

5. Turlar bo'limida bir vaqtda quyidagi ikki sanoqli turlarni tavsiflash mumkinmi?

Type Qalamlar=(yashil, qizil, sariq);

Ranglar=(qora, malla, yashil);

6. Quyidagi tavsif berilgan:

Type shakl=(aylana, kvadrat, romb);

Var t,x:shakl; z,y: (korzina, sumka, portfel);

Quyidagi ta'minlash operatorlaridan qaysilari noto'g'ri yozilgan va nega?

a) *x:=aylana;*

b) *z:=kvadrat;*

c) *t:=x;*

d) *y:=ord(z) + 1;*

7. Turlar bo'limida detallar ro'yxati tavsiflangan.

Type det_roy: (shurup, gayka, vint, shpilka);

Ko'rsatilgan nomer bo'yicha detal nomini chiqaruvchi dastur tuzing.

8. Sanoqli tur o'zgaruvchisi *fasl* tavsifi quyidagicha:

Var fasl: (yoz, kuz, qish, bahor);

butun turga tegishli i o'zgaruvchining qiymati bo'yicha kiritilgan o'zgaruvchi tartib nomeri i ga teng qiymatni ta'minlang. Dastur bo'lagini tuzing.

9. Quyidagi tavsiflar yozilishi to'g'rimi?

Var fasl: (yoz, kuz, qish, bahor)

fasl 1: yoz..qish;

fasl 2: bahor..kuz;

fasl 1: qish..qish;

10. Quyidagi dasturdan xatolarni toping.

Program Xato;

Type oy=(yan, vev, mar, apr, may, iyun, iyul, avg, sen, okt, noy, dek);

Kuz: sen...noy;

Var m:kuz; d:'0'..'9'; k:0..9;

Begin

Read(m,d,k);

If m>sen then d:=k else k:=ord(m)-8;

Writeln(k,d+k)

End.

Paskal tilining qaysi qoidalarini buzilganligini tushuntiring.

11. Talabalarning tug'ilgan yillarini xarakterlovchi *tug'yl* o'zgaruvchini tavsiflang. Qaysi turni ishlatish qulay?

12. Quyida tavsiflangan cheklangan tur uchun asosiy (bazaviy) tur qaysi? Bu tur o'zgaruvchisi uchun qaysi amallar va standart funksiyalar aniqlangan?

Type mashinalar=(sm3, sm4, iskra226, ibm, pentium);

Var x,y:sm3..iskra226; i:3..9;

13. Quyidagi operator bajarilishi natijasida nima chiqariladi?

k:=40;

Case k mod 4 of

0: writelb('k=4m');

1: writelb('k=4m+1');

```

2: writelb('k=4m+2');
3: writelb('k=4m+3');
end;

```

14. Quydagi keltirilgan dastur bo'lagidagi xatolikni toping.

```

Var t:Char; ...
...T:='a';
Case ord(T) of
  'a': t:=Succ(T);
  'b': t:=Pred(T);
end;

```

15. If B then <operator1> else <operator2> operatori hamisha

```

Case B of
  True: <operator1>;
  False: <operator2>;

```

End; operatori bilan ekvivalent yoki yo'qligini aniqlang.

16. Agar tanlov operatoridagi selektor birorta ham nishon bilan mos tushmasa nima bo'ladi?

17. Faqat *a*, *b*, *c* harflardan tuzilgan va oxiriga nuqta qo'yilgan matn berilgan. Har bir harfning necha marta uchrashini hisoblang. Tanlov operatoridan foydalanib dastur tuzing.

3.5. 1-topshiriq. Sanoqli turlar bilan ishlash.

Topshiriq maqsadi.

1. sanoqli tur va tanlov operatori bilan tanishish;
2. sanoqli tur o'zgaruvchisi qiymatlarini kiritish-chiqarishni amalga oshirishda ko'nikma hosil qilish.

Masalaning qo'yilishi.

Yil choragining nomeri, bu chorakdan biron sana (son va oy nomi) va bu sanaga mos hafta kuni nomeri berilgan. Bu sanadan keyingi kun sanasi aniqlansin.

Hisobot mazmuni.

1. aniq berilgan masala uchun masalaning qo'yilishi.
2. dastur blok-sxemasi
3. dastur matni.
4. dasturni testlar asosida sinash natijalari.
5. dasturning ish natijalar.
6. yo'l qo'yilgan xatolar tahlili.

Uslubiy ko'rsatmalar.

Topshiriq ikki usul bilan bajarilishi mumkin.

a) yil choragi nomeri, sana va hafta kuni dasturda bevosita o'zgarmas ko'rinishida beriladi.

b) boshlang'ich son va hafta kuni tanlov operatori yordamida kiritiladi.

2. Quyidagi dastur berilgan sana bo'yicha keyingi kun sanasini aniqlaydi.

Program Sana;

Type oy=(yan, fev, mar, apr, may, iyun, iyul, avg, sen, okt, noy, dek);

kun:1..31;

yil:1900..2010;

Var d: kun; o: oy; y: yil; raq: 1..12;

Begin

Write('==>');

Readln(d, raq, y);

Case raq of

1: m:=Yan;

2: m:=Fev;

3: m:=Mar;

```

4: m:=Apr;
5: m:=May;
6: m:=Iyun;
7: m:=Iyul;
8: m:=Avg;
9: m:=Sent;
10: m:=Okt;
11: m:=Noy;
12: m:=Dek;
end;
Case m of
  yan, mar, may, iyul, avg, okt:
    if d=31 then
      begin
        d:=1;
        m:=Succ(m)
      end
      else d:=d+1;
  apr, iyun, sen, noy:
    if d=30 then
      begin
        d:=1;
        m:=Succ(m)
      end
      else d:=d+1;
  dek:
    if d=31 then
      begin
        d:=1;
        m:=Yan;
        y:=y+1;
      end
      else d:=d+1;

```

```

    fev:
    if (d=28) and ((y mod 4<>0) or (y mod 100=0)) and (y mod
400<>0) then
        begin
            d:=1;
            m:=mar;
        end
        else d:=d+1;
    end;
    Write(d:2);
    Case m of
        yan: Write('Yanvar');
        fev: Write('Fevral');
        mar: Write('Mart');
        apr: Write('Aprel');
        may: Write('May');
        iuyn: Write('Iuyn');
        iuyl: Write('Iuyl');
        avg: Write('Avgust');
        sen: Write('Sentabr');
        okt: Write('Oktabr');
        noy: Write('Noyabr');
        dek: Write('Dekabr');
    end;
    Writeln('y:5', 'yil');
End.

```

28 Fevral 2004 yil boshlang'ich ma'lumot uchun dastur ishi natijasi:

==> 28 2 2004

28 Fevral 2004 yil

Topshiriq variantlari.

№	Yil choragi nomeri	Sana va hafta kuni
1	1	27 fevral, juma
2	2	12 aprel, yakshanba
3	3	26 avgust, chorshanba
4	4	30 noyabr, dushanba
5	1	23 mart, dushanba
6	2	9 may, shanba
7	3	1 Sentyabr, seshanba
8	4	9 Dekabr, chorshanba

9	1	12 Yanvar, dushanba
10	2	1 May, juma
11	3	30 Avgust, yakshanba
12	4	7 Oktyabr, chorshanba
13	1	23 Yanvar, juma
14	2	11 May, dushanba
15	3	22 Sentyabr, seshanba
16	4	16 Noyabr, dushanba
17	1	29 Mart, yakshanba
18	2	1 Iyun, dushanba
19	3	13 Avgust, payshanba
20	4	21 Dekabr, dushanba

4-MAVZU. MASSIVLAR TURI

4.1. Massivlarni tavsiflash.

Massiv - bir turga tegishli ma'lumotlarning tartiblangan to'plamidir. Massiv uning komponentlarining va indekslarining turini ko'rsatish yordamida tavsiflanadi. Indekslar turini ($T1$ ni) ko'rsatishda massiv komponentlari soni beriladi, shuning uchun indekslar turi haqiqiy turdan tashqari har qanday skalyar yoki cheklangan tur bo'lishi mumkin. Massiv komponentlari turi har qanday oddiy yoki murakkab tur bo'lishi mumkin.

Massiv tavsifining umumiy ko'rinishi quyidagicha:

Array [T1] of T2;

Turlar va o'zgaruvchilar tavsifiga misollar qaraymiz:

Type color=(red, black, blue);

Mas1=array[-5..15] of real;

Mas2=array[1..10, 1..10] of integer;

Mas3=array['a'..'c'] of array [1..5] of boolean;

Mas4=array[boolean] of char;

Var a:mas1; b: mas2; c:mas3; d:mas4; e: array[color] of real;

Tavsiflangan a o'zgaruvchi elementlari -5 dan 15 gacha nomerlangan haqiqiy sonlar bo'lgan vektordan iborat. b o'zgaruvchi esa o'nta satr va o'nta ustunlik butun sonlardan tashkil topgan kvadrat matritsa. c o'zgaruvchi uchta satr ('a', 'b', 'c') va beshta ustundan tuzilgan mantiqiy kattalik matritsasi kabi tavsiflangan. Bu matritsaning ikkinchi satri va birinchi ustuni kesishgan joydagi elementiga murojaat quyidagicha amalga oshiriladi: $c[b, 1]$. d o'zgaruvchi ikki komponentdan tashkil topgan belgili vektordan iborat, e o'zgaruvchi turi bevosita o'zgaruvchilar bo'limida tavsiflangan komponentli vektor hisoblanadi, chunki indeks turi sifatida ko'rsatilgan color sanoqli tur uch ob'yektdan tashkil topgan va ilgariroq tavsiflangan. e vektorning komponentlari

quyidagicha: *e[red]*, *e[blue]*, *e[black]*, bu komponentlarda har biri ifodalarda oddiy haqiqiy o'zgaruvchi kabi qo'llanilishi mumkin. Yuqorida keltirilgan misollarda ikki o'lchamli massivlarni tavsiflashning ikki shakli ko'rsatilgan. Umuman, Paskal tilida ko'p o'lchamli massivlarni tavsiflashning ikki shakli bor.

Birinchi shaklda *array* yordamchi so'zidan so'ng kvadrat qavs ichida barcha indekslarning turi, so'ngra, *of* yordamchi so'zidan so'ng komponentlar turi ko'rsatiladi. Massiv o'lchovi cheklanmagan.

Ikkinchi ko'rinish umumiyroq bo'lib massiv komponentlari turi sifatida ixtiyoriy tur, shu jumladan, massiv ham olinishi mumkin. *c* nomli o'zgaruvchi uch komponentdan iborat massiv shaklida tavsiflangan. Har bir komponent o'z navbatida besh mantiqiy kattalikdan tashkil topgan massivdan iborat. Shuning uchun *c['b']* yozuv *c* vektorning ikkinchi satrini bildiradi (bu satr beshta mantiqiy elementdan tashkil topgan), *c['b']/[5]* ikkinchi satrining beshinchi elementi hisoblanadi. Ikki o'lchamli massivning bu komponentini *c['b', 5]* kabi tavsiflash mumkin.

Quyida keltirilgan dastur *mm* matritsadagi har bir satrning musbat elementlari yig'indisini topadi va hosil bo'lgan sonlardan *vv* vektorni hosil qiladi.

```
Type mat1=array[0..5, 0..9] of integer;
```

```
    vek1=array[0..5] of integer;
```

```
Var mm: mat1;
```

```
    vv: vek1;
```

```
    s,i,j:integer;
```

```
.....
```

```
For i:=0 to 5 do
```

```
    Begin
```

```
        s:=0;
```

```
        For j:=0 to 9 do
```

```

    If  $mm[i,j] > 0$  then  $s := s + mm[i,j]$ ;
     $vv[i] := s$ ;
end;
.....

```

4.2. Yig'ilgan massivlar.

Paskal tilida massivlar yig'ilgan holatda bo'lishi mumkin, bu holda xotiradan joy tejaladi, ammo massiv komponentlariga murojaat vaqti kattalashadi. Yig'ilgan massiv dasturda xuddi yig'ilmagan massiv kabi foydalaniladi. Masalan,

```

Var  $ap$ : packed array [1..3] of boolean;
     $anp$ : array [1..3] of boolean;

```

kabi tavsiflangan ap va anp massivlar hajm va unda saqlanuvchi axborot turi jihatidan bir xil, ammo ular EHM xotirasida turli yo'llar bilan tavsiflangan. Bu vektorlarning $ap[3]$ va $anp[3]$ komponentlarini qaraylik. Ko'rinib turibdiki, murojaat usuli bir xil, ammo xotiradan $ap[3]$ komponentga murojaat vaqti ancha katta.

Paskal tilida belgili turdagi yig'ilgan massivlarga yoki operatorlar bo'limida, yoki o'zgarmaslar bo'limida beriladigan belgilar satri kabi qaraladi. Ma'lumki, o'zgarmaslar turi uning qiymati bilan aniqlanadi. Masalan, agar o'zgarmaslar bo'limida $st = 'end'$ o'zgarmas aniqlangan bo'lsa, u holda bu o'zgarmas *packed array[1..3] of char* turiga tegishli bo'ladi.

Satrlar va yig'ilgan belgili massivlar uchun munosabat amallari aniqlangan, lekin bu holda operandlarning simvollar soni bir xil bo'lishi talab qilinadi.

Simvollar ketma-ketligiga munosabat amallari qo'llanilganda simvollar chapdan o'ngga qarab ketma-ket solishtiriladi. Masalan, *'string' < 'strong'*, chunki alifbo bo'yicha *'i'* simvol *'o'* simvoldan oldin turadi.

4.3. Massiv elementlarini kiritish va chiqarish.

Massiv elementlarining har biri alohida - alohida kiritiladi yoki chiqariladi. Bu qoidadan satr va yig'ilgan belgili massivlarni kiritish mustasno. Massiv elementlarini kiritish va chiqarishni tushuntirib beruvchi dasturga misol qaraymiz.

```
Program Massiv;  
Const s='satr';  
Var xs: packed array [1..6] of char;  
      xf: array [1..3, 1..2] of real;  
      i,j:integer;  
{1-qism}  
Begin  
      Readln;  
      Write('xs=>');  
      For i:=1 to 6 do Read(xs[i]);  
{2-qism}  
      Readln;  
      Write('xf=>');  
      For i:=1 to 3 do  
          For i:=1 to 2 do Read(xf[i,j]);  
{3-qism}  
      Writeln(s, '=',xs);  
{4-qism}  
      For i:=1 to 3 do  
          begin  
              For j:=1 to 2 do Write(xf[i,j]:6:2);  
              Writeln;  
          end;  
end.
```

Yuqorida keltirilgan dastur muloqot rejimida ishlaydi, shuning uchun ekranda dastur ishi quyidagicha olib boriladi.

```
xs=>ABCDEF  
xf=>1.1 1.2 1.1 2.2 3.1 3.2
```

satr=ABCDEF

1.10 1.20

2.10 2.20

3.10 3.20

Dasturning *birinchi bo'lagi xs* yig'ilgan belgili massivni kiritishni amalga oshiradi, u birinchi satrning hamma elementlaridan keyingi satrning birinchi elementigacha bo'lgan joyni o'tkazib yuboradigan *Readln* operatori bilan boshlanadi. So'ngra *Writeln('xs=>')* operatori kiritishni boshlashni taklif qiluvchi ma'lumotni chiqaradi. Yozilgan belgilar satri bittalab kiritiladi.

Dasturning *ikkinchi bo'lagi* ichma-ich joylashgan ikki sikl operatorlari yordamida haqiqiy matritsani kiritadi.

Dasturning *uchinchi bo'lagi* o'zgarmaslar bo'limida aniqlangan *s* satrni, '=' belgisini va *xs* yig'ilgan belgili massivning hamma komponentalarini chaqiradi. Ko'rinib turibdiki, berilgan satrni va yig'ilgan belgili massivni chiqarish uchun ularning nomini *Writeln* operatorining parametri sifatida berish yetarli. Agar belgili massiv yig'ilmagan bo'lsa, u holda bunday massiv elementlarini chiqarishda takrorlash operatorlaridan foydalanish zarur.

Dasturning *to'rtinchi bo'lagi xf* haqiqiy matritsa elementlarini uch satr va ikki ustunga chiqaradi. Matritsani chiqarishni tashkillashtirish uni kiritishga o'xshash.

Misol. Uzunligi ($l \geq 3$) ixtiyoriy bo'lgan satrda *end* so'zining necha marta uchrashini hisoblovchi dasturni qaraymiz.

Program LL;

Var s: packed array [1..3] of char;

k:integer;

Begin

Readln;

Write('=>');

```

Read(s[1],s[2]);
Repeat
  Read(s[3]);
  If s='end' then k:=k+1;
  s[1]:=s[2];
  s[2]:=s[3];
Until eoln;
Write('end so ''zi ',k,' marta uchraydi');
end.

```

Dastur dastlab s massivga berilgan satrning birinchi uchta belgisini kiritishni so'raydi, so'ngra sikl ichida s massiv 'end' belgisi bilan solishtiriladi va ular mos kelgan hollarda so'zning uchraganligini sanab boruvchi k qiymat bittaga oshiriladi. So'ngra ikkinchi komponent qiymati birinchiga, uchinchisining qiymati ikkinchisiga ta'minlanadi. Shundan so'ng takrorlash oxirida kiritilgan satrda yana belgilar bor-yo'qligi yoki oxirgi belgi o'qilgan-o'qilmaganligi tekshiriladi. Agar o'qilmagan belgilar bor bo'lsa jarayon boshidan qaytariladi.

Dasturda Paskal tilining *eoln* standart funksiyasi qo'llanilgan, bu funksiya kiritilgan satrning oxirgi simvoli o'qilgan holda, *True* qiymat, aks holda *False* qiymat hosil qiladi.

Dastur ishi natijasi:
 => endreonzendwapendy
 end so'zi 3 marta uchraydi

4.4. Savollar va mashqlar

1. Quyida keltirilgan bir o'lchamli massivlar tavsiflarining qaysilari noto'g'ri va nega?

- a) *var a: array[1..20] of real;*
- b) *Type mas=array['a'..'d']*
Var a:mas;

- c) *Var a:array[integer] of char;*
- d) *Var mm:array[false..true] of char;*
- e) *Type ss=packed array [-20..0] of integer;*
Var mas:ss;
- f) *Type shahar=(Samarqand, Toshkent, Buxoro);*
Var a:array[shahar] of real;

2. 1 - bandda to'g'ri tavsiflangan massivlarning nechtdan elementlari bor?

3. *a:array['k'..'z'] of real;* kabi tavsiflangan massivning har bir elementiga nol ta'minlang.

4. *a* massiv quyidagicha tavsiflangan: *a: packed array[-5..5] of char;* Uning elementlarini kiritish va ularni chiqarish dasturini tuzing.

5. Quyida keltirilgan ikki o'lchamli massivlar tavsiflarining qaysilari noto'g'ri va nega?

- a) *var a: array[1..5,1..7] of real;*
- b) *Type mas=array['a'..'d', -2..3] of integer;*
Var arr:mas;
- c) *Type mm= packed array[1..10] of char;*
Var a:array[2..7] of mm;
- d) *Var w:array[5,7] of integer;*
- e) *Type niqob(ayiq, tulki, quyon);*
Var a:array[niqob] of array[1..5] of char;
- f) *Var mass: array[1..5],[1..7] of real;*

6. Quyida tavsiflangan massivlarning nechtdan elementlari bor?

- a) *Var mb: array['a'..'c', 2..5] of real;*
- b) *Var mc:array[false..true] of array[1..5] of char;*
- c) *Type ww=(m1,m2,m3);*
Var zar:array[ww] of array [ww] of boolean;

7. 6-bandda ko'rsatilgan tavsiflardan foydalanib quyidagi massiv elementlariga murojatlardan qaysilari noto'g'rligini aniqlang?

- a) *mb*['b',3]
- b) *mc*[true,2]
- c) *zap*[m2,m1]
- d) *mb*[b][6]
- e) *mc*[3,false]
- f) *zap*[true,m3]

8. *a* massiv quyidagicha tavsiflangan:

Type shakllar=(*romb, uchburchak, kvadrat, trapetsiya*);

Var a:array[1..10] of array[shakllar] of real;

i:integer; j:shakllar;

a massiv elementlarini kiritishni amalga oshiruvchi dastur bo'lagini tuzing.

9. *a* massiv quyidagicha tavsiflangan bo'lsin:

Type mas=array[1..5,1..3] of real;

mas1=array[1..3] of real;

Var a:mas; b:mas1;

a[i]:=b yozuv to'g'ri bo'ladimi?

10. Paskal tilida massiv o'lchami cheklanganmi?

11. Dasturda

Const a='detal'; b='.'

Var ma: packed array [1..6] of char

s:char; i: integer;

tavsif mavjud deb hisoblab, quyidagi operatorlardan qaysilari to'g'ri yozilganligini aniqlang.

- a) *s:=a;* c) *s:=b;* f) *mas:=b;*
- b) *mas:=a;* d) *ma[6]:=b;* g)

i:=ord(ma[i])

12. 30 ta belgidan iborat matn kiritib, uni teskari tartibda chiqarish uchun dastur tuzing.

13. 25 ta kishining ro'yxatini kiritish va chiqarish dasturini tuzing. Har bir familiya 15 belgidan oshmasin.

4.5. 1- mashg'ulot. Sonli massivlar bilan ishlash.

Topshiriq maqsadi.

1. Massivlar bilan ishlash uchun amaliy ko'nikma hosil qilish
2. Tartiblash algoritmi bilan tanishish.

Masalaning qo'yilishi.

Aniq berilgan variant uchun boshlang'ich ma'lumotlar massivini tuzing. Tartiblash algoritmlarini o'rganib, ulardan birini tanlang. Ixtiyoriy ma'lumotlar to'plami bilan ishlaydigan dastur tuzing. Kiritilgan qiymatlarni va hisob natijalarini tegishli sarlavhalar ostida chiqaring.

Hisobot mazmuni.

1. Tanlangan variant uchun masalaning qo'yilishi.
2. Dastur matni.
3. Testlar va ular asosida sozlash natijalari
4. Yo'l qo'yilgan xatolar tahlili.
5. Hisoblashlar natijasi

Uslubiy ko'rsatmalar

1. Boshlang'ich ma'lumotlar musbat son, manfiy son va nollarni o'z ichiga olishi kerak.

2. Dasturga misol.

5 satr va 4 ustundan iborat har xil haqiqiy sonli ikki o'lchamli massiv berilgan. Massivning eng katta elementi va eng kichik elementining joyini almashtiring.

Dastur:

Program MASS;

Var m:array[1..5, 1..4] of real;

ml:array[1..5] of real;

max, min:real;

```

    maxi, mini, i, j: integer;
Begin
    Writeln('Boshlang''ich massivni kiriting');
    For i:=1 to 5 do
        For j:=1 to 4 do Read(m[i,j]);
        max:=m[1,1]; min:=m[1,1];
        maxi:=1; mini:=1;
        For i:=1 to 5 do
            For j:=1 to 4 do
                Begin
                    If max<m[i,j] then
                        begin
                            max:= m[i,j];
                            maxi:=i;
                        end;
                    If min>m[i,j] then
                        begin
                            min:= m[i,j];
                            mini:=i;
                        end;
                end;
            m1:=m[maxi];
            m[maxi]:=m[mini];
            m[mini]:=m1;
        Writeln('Satrlar      almashtirilgach      massivning
ko''rinishi');
        For i:=1 to 5 do
            begin
                For j:=1 to 4 do write(m[i,j]:5:2);
                Writeln;
            end;
        end.

```

Dastur ishi bayonnomasi

Boshlang'ich massivni kiriting

1.1 1.2 1.3 1.4

2.1 2.2 2.3 2.4

3.1 3.2 2.2 3.4

4.1 4.2 4.3 4.4

5.1 5.2 5.3 5.4

Satrlar almashtirilgach massivning ko''rinishi

5.1 5.2 5.3 5.4

2.1 2.2 2.3 2.4

3.1 3.2 2.2 3.4

4.1 4.2 4.3 4.4

1.1 1.2 1.3 1.4

Topshiriq variantlari

1. 20 ta sondan iborat massivda eng katta bo'lgan elementini toping va uni birinchi element bilan o'rin almashtiring.

2. 10 ta butun sonlardan iborat massivda eng kichik bo'lgan elementini toping va uni eng oxirgi element bilan o'rin almashtiring.

3. 15 ta butun sonlardan iborat massivda eng katta bo'lgan elementini toping va uni eng oxirgi element bilan o'rin almashtiring.

4. 25 ta butun sonlardan iborat massivda eng kichik bo'lgan elementini toping va uni birinchi element bilan o'rin almashtiring.

5. 20 ta butun sondan iborat bo'lgan massiv elementlarini o'sish tartibida joylashtiring.

6. 10 ta butun sondan iborat bo'lgan massiv elementlarini kamayish tartibida joylashtiring.

7. 15 ta haqiqiy sondan iborat bo'lgan massiv elementlarini o'sish tartibida joylashtiring.

8. 25 ta haqiqiy sondan iborat bo'lgan massiv elementlarini kamayish tartibida joylashtiring.

9. 20 ta elementdan tashkil topgan butun sonli massiv berilgan, shu massivga elementlarning joylashish tartibini saqlagan holda avval manfiy sonlar va nollarni, so'ngra musbat sonlarni yozing.

10. 10 elementdan tashkil topgan butun sonli massiv berilgan, shu massivga elementlar tartibini saqlagan holda avval hamma musbat elementlarni, so'ngra hamma nol va manfiy sonlarni yozing.

11. 15 elementdan tashkil topgan haqiqiy sonli massiv berilgan, shu massivga elementlar tartibini saqlagan holda avval hamma manfiy elementlarni, so'ngra hamma nol va musbat sonlarni yozing.

12. 3 satr va 5 ustundan iborat ikki o'lchamli massiv berilgan. Massiv elementlari butun sonlar. Massivning birinchi ustuni elementlarini o'sish tartibida joylashtiring.

13. 4 satr va 5 ustundan iborat ikki o'lchamli massiv berilgan. Massiv elementlari butun sonlar. Massivning beshinchi ustuni elementlarini kamayish tartibida joylashtiring.

14. 3 satr va 4 ustundan iborat ikki o'lchamli massiv berilgan. Massiv elementlari haqiqiy sonlar. Massivning birinchi ustuni elementlarini o'sish tartibida joylashtiring.

15. 4 satr va 4 ustundan iborat ikki o'lchamli massiv berilgan. Massiv elementlari haqiqiy sonlar. Massivning ikkinchi ustuni elementlarini kamayish tartibida joylashtiring.

4.6. 2 – mashg'ulot. Yig'ilgan belgili massivlar bilan ishlash.

Topshiriq maqsadi.

1. Belgili massivlar (satrlar) bilan ishlash uchun amaliy ko'nikma hosil qilish
2. Mashina vaqti va EHM xotirasini samarador ishlatish usullari bilan tanishish.

Masalaning qo'yilishi.

N ta nomlangan mahsulotlar ro'yxatini tuzing. Har bir mahsulotning narxi va soni ma'lum. Boshlang'ich ma'lumotlarni kiritish. Alohida olingan ma'sulot narxi va mahsulotlarning jami qiymatini hisoblash dasturini tuzing. Natijani jadval ko'rinishda chiqaring.

Hisobot mazmuni.

1. Masalaning qo'yilishi
2. Dastur matni va uning bajarilishi natijasi.
3. Xulosalar.

Uslubiy ko'rsatmalar

1. Nomlangan mahsulotlar sonini $10 \leq N \leq 15$ oraliqdan oling. Mahsulotlar nomini ikki o'lchamli massiv ko'rinishida tasvirlang. Masalan, *mas:array[1..12] of packed array[1..15] of char;*

Agar nomdagi belgilar 15 belgidan kam bo'lsa, u holda ular kiritilayotganda bo'sh joylar bilan to'ldiriladi. Natijani chiqari ko'rinarli bo'lishi kerak va quyidagi ko'rinishda bo'lishi kerak:

Tartib nomeri	Mahsulot nomi	Mahsulot narxi	Mahsulot soni	Umumiy narx
------------------	------------------	-------------------	------------------	----------------

Jadval ramkalarini biror belgi bilan, masalan, '!', '=', yoki '*' belgilari bilan to'ldirish mumkin.

2. *Dasturga misol.* Beshta nomlangan mahsulot narxi ma'lum. Mahsulot nomini va uning narxini chiqarish talab qilinsin. Ma'lumotlar jadval ko'rinishida chiqarilsin. Har bir nom uzunligi o'nta belgidan oshmasin.

Program TAB;

Const n=5;

Var mastow: array[1..5] of packed array[1..10] of char;

narx: array[1..5] of real;

i,j: integer;

Begin

Writeln('10 belgidan oshmagan mahsulot nomini kiriting');

Readln;

For i:=1 to n do

For j:=1 to 10 do Read(mastow[i,j]);

Writeln('Mahsulot narxini kiriting');

For i:=1 to n do Read(narx[i]);

For i:=1 to 26 do Write('');*

Writeln;

*Writeln('*n* mahsulot *narx*');*

*Writeln('*P/P* nomi *sum *');*

For i:=1 to 26 do Write('');*

Writeln;

For i:=1 to n do

Writeln('',i:2, '* ',mastow[i], '* ',narx[i]:6:2, '* ');*

For i:=1 to 26 do Write('');*

End.

Dastur ishi bayonnomasi:

10 belgidan oshmagan mahsulot nomini kiriting

Arra qirg'i tesha bolta ombur

Mahsulot narxini kiriting

1.45 3.7 10.2 5.35 2.1

<i>n</i>	<i>mahsulot</i>	<i>narx</i>
<i>P/P</i>	<i>nomi</i>	<i>sum</i>

<i>1</i>	<i>Arra</i>	<i>1.45</i>
<i>2</i>	<i>qirg'i</i>	<i>3.7</i>
<i>3</i>	<i>tesha</i>	<i>10.2</i>
<i>4</i>	<i>bolta</i>	<i>5.35</i>
<i>5</i>	<i>ombur</i>	<i>2.1</i>

5-BO'LIM. ARALASH TUR. BIRLASHTIRUVCHI OPERATOR.

Aralash turlar tarkibli ma'lumotlarni tashkillashtirishning eng yaxshi vositasi hisoblanadi. Aralash tur turli qismlari har xil xarakteristikaga ega bo'lgan ob'ektlarni tavsiflash imkonini beradi.

5.1. Yozuvlar.

Aralash turlar yozuvlar deb ataladigan ob'ektlarni tavsiflaydi. Yozuv – bu bir necha tarkibiy qismlarga (komponentalarga) ega bo'lgan murakkab o'zgaruvchi. Massivlardan farqli ravishda yozuvning komponentalari (maydonlari) har xil turlarga tegishli bo'lishi mumkin va ularga murojaat indeks bo'yicha emas, balki maydon nomi bo'yicha amalga oshiriladi. Aralash turni tavsiflashda uning nomi va har bir maydon turi beriladi. Aralash tur tavsifi *Record* yordamchi so'zi bilan boshlanadi va *end* so'zi bilan tugatiladi.

Misol sifatida talabalarning anketali ma'lumotini tavsiflovchi yozuv qaraymiz:

Type

student: record

fio: packed array [1..15] of char;

tugkun: record

sana: 1..31;

oy:(yan, vev, mar, apr, may, iyun, iyul, avg, sen, okt, noy, dek);

yil: integer;

end;

jins:(erk, ayol);

end;

Var cc: student; group: array [1..30] of student;

Yozuvning har bir komponentiga murojaat qilish uchun yozuvning nomi va kerakli komponent nomi nuqta bilan ajratilib yoziladi. Masalan, *cc.fio:= 'Valiyev A.A.'* operatori *cc*

yozuvning *fio* maydoniga 12 belgidan tashkil topgan satrni ta'minlaydi. Ma'lumotlarning *Student* turiga tegishli *tugkun* nomli ikkinchi maydoni o'z navbatida yozuv turiga ega va uchta maydondan tashkil topgan. Bu maydonlarga quyidagicha murojaat qilish mumkin:

cc.tugkun.sana:=25;

cc.tugkun.oy:=fev;

cc.tugkun.yil:=1986;

5.2. Birlashtirish operatori.

Yuqorida keltirilgan ta'minlash operatorlarini birlashtirish operatoridan foydalanib yozilganda ancha qisqaradi. Birlashtirish operatorining umumiy ko'rinishi quyidagicha:

*With <o'zgaruvchi-yozuvlar, maydonlar ro'yxati> do
<operator>;*

Birlashtirish operatori sarlavhasida ko'rsatilgan o'zgaruvchi-yozuv va maydon nomlarini *With* operatorining ta'sir sohasida yozuv komponentalariga murojaatda tushirib qoldirish mumkin. Translyator avtomatik ravishda to'liq ismni shakllantiradi. Masalan,

With cc.tugkun do

Begin

Sana:=25;

Oy:=fv;

Yil:=1986

End;

Yozuvlar boshqa o'zgaruvchilarning tarkibiy qismi sifatida ishlatilishi ham mumkin. *Group 30* ta yozuvdan tashkil topgan massiv bo'lsin. Massivning beshinchi yozuvida *oy* nomli maydonga yangi qiymat ta'minlash quyidagi ko'rinishga ega bo'ladi:

Group[5].tugkun.oy:=dk;

Har bir yozuvga (massiv elementiga) murojaat indekslar yordamida amalga oshishini osongina ko'rish mumkin.

Quyida keltirilgan dastur bo'lagi 1984 yilda tug'ilgan o'g'il bolalarning sonini hisoblaydi:

```
.....  
k:=0;  
for i:=1 to 30 do  
  with group[i].tugkun do  
    if (yil=1984) and (jins=erk) then k:=k+1;  
  writeln('O''g''il bolalar sonu', k:2);  
.....
```

Birlashtirish operatori takrorlash ichida ekanligiga e'tibor berish kerak. Bu hol birlashtirish operatorining ta'sir doirasida sarlavhada ko'rsatilgan o'zgaruvchilar ro'yxati elementlarini, ya'ni yozuvlar va maydonlar nomini o'zgartirish mumkin emasligini bildiradi.

5.3. Savollar va mashqlar

1. Qaysi tur yozuvlar deb ataluvchi ob'yektlarni xarakterlaydi?

2. Yozuv komponentlari har xil turga tegishli bo'lishi mumkinmi?

3. Quyidagi tavsif qaysi turga tegishli?

```
kitob=record  
  nomi=packed array [1..20] of char;  
  chiq_yil: 1000..2004;  
  narxi: real;  
end;
```

Yozuvning komponentlarini va ularning turlarini ko'rsating.

4. Quyidagi tavsif qaysi berilgan bo'lsin:

Type oila: record

Oila_boshl: packed array [1..15] of char;

yoshi: 10..100;

farz: array [1..8] of record

nom: packed array [1..10] of char;

tugyil: 1900..2000

end

end;

Var a,b: oila;

Birlashtirish operatoridan foydalanmasdan boshlang'ich ma'lumotni *a* o'zgaruvchiga kiriting va uni *b* o'zgaruvchiga ko'chiring.

5. Quyida keltirilgan tur tavsifidan xatoni ko'rsating:

Type gepm_j=record

nomi: packed array [1..10];

s, sp: real;

hajm: real;

6. Quyida tavsiflar berilgan:

Type vaqt=record

soat: 0..23;

min, sek: 0..59;

end;

Var t1, t2: vaqt;

t1 va *t2* yozuvlarning maydonlariga qiymat kiritadigan va *t1* qiymatni *t2* ga nisbatan oldingi vaqtni ko'rsatsa, *true* so'zini, aks holda *false* so'zini pechatga chiqaradigan dastur tuzing.

7. Mahsulot haqida quyidagi ma'lumotlarni saqlovchi yozuv tavsiflang.

- ◆ tayyorlovchi zavod nomi;
- ◆ xom ashyo beruvchi zavod nomi;
- ◆ talabgor do'kon nomi;

- ◆ nomeri;
- ◆ ishlab chiqariladigan 5 xil mahsulot nomlari;
- ◆ har bir mahsulotning so'm va tiyinlarda narxini;
- ◆ har bir mahsulotning soni;

8. Yozuv maydonlariga murojaat qanday amalga oshiriladi?

9. Yozuvlar boshqa o'zgaruvchilarning komponentalari sifatida kelishi mumkinmi?

5.4. 1-Topshiriq: Aralash tur bilan ishlash.

Topshiriq maqsadi:

1. aralash tur qiymatlarini kiritish–chiqarishni tashkillashtirish ko'nikmasini olish;
2. yozuvlar qo'llaniladigan masalalarni dasturlash amaliy ko'nikmasini olish;

Masalaning qo'yilishi:

25 o'quvchidan iborat guruh ro'yxatini tuzing. Har bir o'quvchi uchun tug'ilgan sanasi, o'qishga kirgan yili, kursi, guruhi, har bir o'quv yilidagi baholarini ko'rsating. O'quvchilar haqidagi axborotni dasturda yozuv ko'rinishida tasvirlang. Yozuvlar majmuini massivga birlashtiring.

O'quvchilar haqidagi ma'lumotni kiritish, ularni jadval ko'rinishda chop qilish, shuningdek topshiriqning berilgan variantida ko'rsatilgan ma'lumotlarni chiqarish uchun dastur tuzing.

Hisobot mundarijasi:

1. masalaning qo'yilishi.
2. boshlang'ich ma'lumotlar.
3. dastur matni va uning bajarilish natijasi.
4. yo'l qo'yilgan xatolar tahlili.

Uslubiy ko'rsatmalar.

Dasturda belgili axborotni yig'ilgan massiv ko'rinishida tasvirlash kerak.

Dasturga misol: Zap nomli dastur besh talaba haqida axborot kiritadi: familiya, tug'ilgan yili, o'qishga kirgan yili va oxirgi sessiyadagi baholari. Dastur ishi natijasi a'lochilar ro'yxatini chiqarishdan iborat. Dastur birlashtirish operatori qo'llanilgan yozuvlar bilan ishlashda birlashtirish operatoridan foydalanish imkoniyatlarini namoyish etadi.

Program Zap;

```
    Type student=record
      fio: packed array [1..10] of char;
      tug_y: 1980..1986;
      uk_y: 2000..2010;
      baho: record
        alg, tar, inf: 1..5
      end;
    end;
```

end;

Var group: array [1..5] of student;

s1: student;

i,j : integer;

begin

readln;

with s1. baho do

begin

for i:=1 to 5 do

begin

for j:=1 to 5 do read(fio[j]);

readln(tug_y, uk_y, alg, tar, inf);

group[i]:=s1

end

end;

writeln('A 'lochilar ro'yxati:');

```

for i:=1 to 5 do
with group[i]. baho do
if (alg=5) and (tar=5) and (inf=5) then
  writeln(fio, ' tug'ilgan yili-', tug_y:4, ' o'qishga kirgan
yili-',uk_y:4);
end.

```

Dastur ishi bayonnomasi:

Voronov	1985	2004	3	4	3
Rashidov	1985	2004	5	5	5
Zaripov	1986	2004	4	5	5
Husanov	1986	2004	5	5	5
Davronov	1985	2004	3	4	5

A'lochilar ro'yxati:

Rashidov tug'ilgan yili-1985 o'qishga kirgan yili-2004

Husanov tug'ilgan yili-1986 o'qishga kirgan yili-2004

Topshiriq variantlari:

1. A'lochi talabalarining anketali ma'lumotlarini chop etuvchi dastur tuzing.

2. Baholari 4 va 5 bo'lgan talabalarining anketali ma'lumotlarini chop etuvchi dastur tuzing.

3. Baholari oxirgi sessiyada faqat 2 bo'lgan talabalarining anketali ma'lumotlarini chop etuvchi dastur tuzing.

4. Baholari barcha sessiyalarda faqat 5 bo'lgan talabalarining anketali ma'lumotlarini chop etuvchi dastur tuzing.

5. Butun o'quv yilida baholari faqat bitta 4 va qolganlari faqat 5 bo'lgan talabalarining anketali ma'lumotlarini chop etuvchi dastur tuzing.

6. Familiyasi *A* harfi bilan boshlanuvchi talabalar ro'yxati va shu talabalarining o'quv yili davomidagi baholarini chop etuvchi dastur tuzing.

7. Familiyasi *B* harfi bilan boshlanuvchi talabalar ro'yxati va shu talabalarining tug'ilgan sanasini chop etuvchi dastur tuzing.

8. Familiyasi *D* va *G* harfi bilan boshlanuvchi talabalarining oxirgi sessiyadagi baholarini chop etuvchi dastur tuzing.

9. Butun o'qish davrida bitta ham 4 baho olmagan talabalarining familiyasi va tug'ilgan yillarini chop etuvchi dastur tuzing.

10. Talabalar ro'yxatini o'rtacha ball bo'yicha tartiblovchi va chop etuvchi dastur tuzing.

11. Talabalar ro'yxatini oxirgi sessiyadagi o'rtacha ball bo'yicha tartiblovchi va chop etuvchi dastur tuzing.

12. Guruhning o'rtacha balini hisoblang va o'rtacha baldan yuqori ball olgan talabalarining ro'yxatini chop etuvchi dastur tuzing.

13. Guruhning o'rtacha balini hisoblang va o'rtacha baldan past ball olgan talabalar ro'yxatini chop etuvchi dastur tuzing.

14. Guruhning oxirgi sessiyadagi o'rtacha balini hisoblang va o'rtacha ball olgan talabalar ro'yxatini chop etuvchi dastur tuzing.

15. Talabalar ro'yxatini tug'ilgan yili bo'yicha tartiblovchi va chop etuvchi dastur tuzing.

16. Talabalar ro'yxatini alifbo bo'yicha tartiblovchi va chop etuvchi dastur tuzing.

17. Talabalar ro'yxatini tug'ilgan oyi bo'yicha tartiblovchi va chop etuvchi dastur tuzing.

18. A'lochilarlar ro'yxatini tug'ilgan yili bo'yicha tartiblovchi va chop etuvchi dastur tuzing.

6-MAVZU. PROTSEDURA VA FUNKSIYALAR.

Protsedura va funksiyalar dasturga o'xshash tuzilmaga ega bo'lib, ular qisman dastur degan umumiy nom bilan nomlanadilar.

Qisman dasturni qo'llash operatorlarning takrorlanib keluvchi guruhlarini bir marta yozib, ko'p marta ishlatish imkonini. Bushdan tashqari bu tushunchalarni qo'llab dasturning bir necha qisman dasturlar majmuasi shaklida loyihalash, tarkibiy qismlarni alohida sozlash va ularni birlashtirib asosiy dasturni tarkib toptirish mumkin. Qisman dasturlar dasturning protsedura va funksiyalar bo'limida tavsiflanadilar. Dasturning operatorlar bo'limida ularga murojaat qilinadi (yoki chaqiriladi) va ular ijro etiladi. Oddiy dastur kabi qisman dasturlar ham o'zining nishonlar, o'zgarmaslar, turlar, o'zgaruvchilar, protsedura va funksiyalar va operatorlar bo'limlariga ega bo'lishi mumkin.

6.1. Protsedura tavsifi.

Har bir protsedura sarlavha bilan boshlanadi. Sarlavhada uning ismi va formal parametrlar ro'yxati ko'rsatiladi. Protsedura parametrlarsiz bo'lishi ham mumkin, bu holda sarlavhada faqatgina uning ismi keltiriladi. Parametrlar boshlang'ich ma'lumotlarni protseduraga uzatish, hamda qayta ishlangan ma'lumotlarni chaqiruvchi dasturga qaytarish uchun xizmat qiladi.

Protsedura sarlavhasining umumiy ko'rinishi quyidagicha:

Procedure <protsedura nomi>(<formal parametrlar ro'yxati>)

Formal parametrlar ro'yxati o'z ichiga parametr qiymatlari, parametr-o'zgaruvchilarni (ulardan oldin *Var* maxsus so'zi turishi kerak), parametr-protseduralarni (ulardan oldin *procedure* maxsus so'zi turishi kerak), parametr-

funksiyalarni (ulardan oldin *function* maxsus so'zi turishi kerak) oladi. Protsedura sarlavhasidan so'ng oddiy dasturdagi kabi hamma bo'limlar keladi.

Protsedurani chaqirish va ijro etish protsedura operatori yordamida amalga oshiriladi:

<Protsedura nomi>(<aniq parametrlar ro'yxati >);

Formal va aniq parametrlar orasida to'liq moslik bo'lishi kerak, ya'ni ularning soni va yozilish ketma-ketligi bir xil bo'lishi kerak. Har bir aniq parametrning turi unga mos keladigan formal parametr turi bilan bir xil bo'lishi shart.

Protseduraga murojaat qilishda uning ichida qiymati o'zgaraydigan o'zgaruvchilar parametr-qiymat sifatida olinadi. Qiymati protsedura ichida yangitdan hisoblanadigan (yoki o'zgaradigan) va asosiy dasturga qaytariladigan o'zgaruvchilar parametr-o'zgaruvchi shaklida tavsiflanadilar.

Masalan, quyida keltirilgan dastur klaviaturadan kiritiladigan satrdagi nuqtalar sonini hisoblaydi, ikkinchisi esa 'a' harflar sonini hisoblaydi. Belgilarni hisoblash hisoblash protsedurasida amalga oshirilgan.

Program Hisoblash;

const nuqta='.';

Var s:integer;

procedure hisoblash(sym: char; var yig: integer);

Var c:char;

begin

Yig:=0;

Writeln('Satrni kiriting');

Repeat

Read(c);

If c=sym then yig:=yig+1;

Until eoln;

end;

begin

```

Readln;
Hisoblash(nuqta,s);
Writeln('Nuqtalar soni', s:3);
Hisoblash('a',s);
Writeln("'a" harflar soni', s:3);
end.

```

Hisoblash protsedurasini ikki formal parametrga ega: *sym* – kirish parametri (parametr–qiymat) satrda hisoblanadigan belgini aniqlaydi. *Yig* – chiqish parametri (parametr–o'zgaruvchi) hisoblangan belgilar sonini asosiy dasturga uzatadi.

sym formal parametrga mos keluvchi aniq parametr protseduraning birinchi chaqirilishida asosiy dasturda aniqlangan o'zgaruvchi nomi bilan berilgan, ikkinchi murojaatda esa oshkora ravishda (bevosita qiymat) ko'rsatiladi.

Protsedura natijalarini asosiy dasturga uzatish nafaqat parametr-o'zgaruvchilar, balki bevosita *global o'zgaruvchilar* yordamida ham bajarilishi mumkin. Asosiy dasturda tavsiflangan o'zgaruvchilar ichki protsedura va funksiyalarga nisbatan global o'zgaruvchi hisoblanadi. Protsedura va funksiyalar ichida tavsiflangan o'zgaruvchilar lokal o'zgaruvchilardir. Protseduraga murojaat bo'lganda ularga xotiradan joy ajratiladi va protseduradan chiqishda ular o'chiriladi, ya'ni lokal o'zgaruvchilar faqat protsedura bajarilishidagina mavjud bo'ladi va asosiy dasturda ularga murojaat qilib bo'lmaydi. Masalan, yuqoridagi *hisoblash* protsedurasida tavsiflangan belgilar turiga tegishli *sym* o'zgaruvchi lokal o'zgaruvchi, asosiy dasturda tavsiflangan butun o'zgaruvchi *s* esa global o'zgaruvchi deb hisoblanadi.

Hisoblash protsedurasini ish natijasini *s* global o'zgaruvchi orqali uzatadigan qilib o'zgartiramiz:

```

Program a_nuqta;
const nuqta='.';

```

```

Var s:integer;
procedure hisoblash(sym: char);
  Var c:char;
  begin
    s:=0;
    Writeln('Satrni kiriting');
    Repeat
      Read(c);
      If c=sym then s:=s+1;
    Until eoln;
  end;
begin
  Readln;
  Hisoblash(nuqta);
  Writeln('Nuqtalar soni', s:3);
  Hisoblash('a');
  Writeln('''a'' harflar soni', s:3);
end.

```

Yuqorida keltirilgan ikki dastur ham bir xil ishlaydi. Qo'yilgan har bir masala uchun protsedura ishi natijalarini uzatishning u yoki bu usulidan foydalanish mumkin, ammo murakkab dasturiy majmualarda protsedura jismida global o'zgaruvchilardan foydalanish tavsiya etilmaydi, bu hol dastur tuzilmasini tushunishning qiyinlashuviga olib keladi.

Paskal tili dasturlarida protsedura va funksiyalar ichma-ich joylashishi mumkin, ya'ni asosiy dasturda tavsiflangan protseduraning tarkibida boshqa bir ichki protsedura, u ham o'z navbatida yana boshqa bir protseduraga va hokazo protseduraga murojaat qilishi mumkin. bunday murakkab tuzilmali dasturlarda ismlarning ta'sir doirasini aniqlash uchun quyidagi lokallashtirish qoidalari ishlatiladi:

1. har qanday ism (o'zgarmaslar, turlar, o'zgaruvchilar, protsedura va funksiyalar) o'zi tavsiflangan protsedura va funksiya ichida amal qiladi.

2. bitta va aynan o'sha nom har bir alohida protsedura (funksiya) yoki dasturda aniqlangan bo'lishi mumkin. Bu holda bu nomning ish–harakat sohasi shu protsedura (funksiya) yoki dastur hisoblanadi.

6.2. *Funksiya tavsifi.*

Funksiya tavsifi protsedura tavsifiga deyarli o'xshash, ammo bir necha farqlar bor. Protseduraning ish natijasida bir yoki bir necha qiymat asosiy dasturga uzatilishi mumkin, funksiya ishi natijasida esa faqat bitta skalyar qiymat yoki bitta ko'rsatkich turiga tegishli qiymat hosil bo'ladi. Natija turi funksiya sarlavhasida ko'rsatiladi. Funksiya tavsifining umumiy ko'rinishi quyidagicha:

Function <funksiya nomi>(<formal parametrlar ro'yxati>):<natija turi>

<Formal parametrlar ro'yxati> funksiya ishlashi uchun boshlang'ich qiymatlar uzatishga xizmat qiladi.

Funksiya jismidagi operatorlar orasida hech bo'lmaganda bitta ta'minlash operatori bo'lishi kerak, bu ta'minlash operatorining chap qismida shu funksiya nomi turadi. Bu operator funksiya qayta ishlagan qiymatni aniqlaydi.

Funksiyani chaqirish standart funksiyalarni chaqirish kabi biron bir ifoda tarkibida uning nomi va aniq parametrlar ro'yxatini ko'rsatish orqali amalga oshiriladi. Funksiya bajarilgach, uning natijasi funksiya nomi qatnashgan ifodalarning qiymatini hisoblash uchun foydalaniladi. Funksiyani chaqirishda aniq parametrlarni uzatish xuddi protseduraning aniq parametrlarini chaqirishdagidek amalga oshiriladi.

6.3. Savollar va mashqlar.

1. Protseduralar nimaga mo'ljallangan?
2. Protsedura sarlavhasi qanday parametrlarni o'z ichiga oladi?
3. Formal va aniq parametrlar bir-biridan qanday farqlanadi?
4. Lokal va global o'zgaruvchilar bir-biridan qanday farqlanadi?
5. Funksiya nimaga mo'ljallangan?
6. Funksiya sarlavhasi qanday parametrlarni o'z ichiga oladi?
7. Funksiya va protseduralar bir-biridan nimasi bilan farqlanadi?
8. Elementlari n ta haqiqiy sondan iborat bir o'lchamli massivning eng kichik elementini topish funksiyasini yarating.
9. Elementlari n ta haqiqiy sondan iborat massivning elementlarini kamayish tartibida joylashtirish protsedurasini yarating.
10. Turlar va o'zgaruvchilar bo'limida quyidagilar berilgan bo'lsin:
type haf_kun=(du, se, cho, pa, ju, sha, ya);
num=1..7;
Var n:num;
Kiritilgan n ($1 \leq n \leq 7$) nomer bo'yicha unga mos hafta kuni nomini chiqarish funksiyasini yozing.
11. Berilgan ikki sonlarning kattasini yoki kichigini topish funksiyasini yozing.
12. Quyidagi dastur xatolikni toping. Paskal tilining qaysi qoidalari buzilganligini aniqlang.

Program XATO;

Var a: real;

Function (x:real):real;

begin

```

    f:=x;
    if x<0 then then f:=f+x
end;
procedure wi (var x:real; y:real);
begin
    x:=siny
end;
begin
    wi(a, f(5));
    writeln(a:6)
end.

```

13. Quyida keltirilgan dastur bo'lagi bajarilishi natijasida nima chop etilishini aniqlang.

```

Program P13;
Var c:char;
procedure p(x,y: char);
begin
    y:=succ(x);
end;
procedure q(x: char; var y:char);
begin
    y:=succ(x);
end;
begin
    c:='a'; p('b', c); writeln('c=',c);
    c:='a'; q('b', c); writeln('c=',c);
end.

```

14. Quyidagi ifodani hisoblashni ta'minlovchi funksiyani yozing:

$$\frac{n!}{m!(n-m)!}, \text{ bu yerda } 0 \leq n \leq m$$

6.4. 1-topshiriq. Funksiyalardan foydalanib dastur tuzish.

Topshiriq maqsadi.

1. funksiya qo'llaniladigan dastur yozish ko'nikmasini olish;
2. parametrlar qiymati bo'yicha uzatish mexanizmini o'rganish.

Masalaning qo'yilishi:

$f(x,y)$ funksiyaning variantda ko'rsatilgan bitta har xil nuqtadagi qiymatlari yig'indisini hisoblash dasturini tuzing. $f(x,y)$ ni hisoblashni funksiya ko'rinishida tasvirlang. $f(x,y)$ funksiyaning ko'rinishi topshiriq variantlarida berilgan.

Hisobot mazmuni.

1. masalaning qo'yilishi;
2. dastur matni va uning bajarilish natijalari;
3. xulosalar.

Uslubiy ko'rsatmalar.

Topshiriqni bajarish uchun x va y argumentlar uchun $f(x,y)=(x+y)^{-5}-(x-2y)^3$ funksiya qiymatini hisoblovchi va natijani chiqaruvchi dastur bilan tanishish zarur. Bu yerda haqiqiy a sonni butun b darajaga ko'tarish uchun *step* nomli funksiya ishlatilgan.

Program FF;

var x, y, fxy: real;

function step(a:real; b: integer); real;

var rab: real; absb: integer;

begin

rab:=1;

absb:=abs(b);

while absb>0 do

```

begin
  rab:=rab*a;
  absb:=absb-1
end;
if b<0 then rab:=1/rab;
step:=rab
end;
begin
  writeln('Ma''lumotlarni kiriting:');
  readln(x,y);
  fxy:=step(x+y, -5)-step(x-2*y,3)
  writeln('x=', x:6:3, ' va y=', y:6:3, ' qiymatlar uchun
f(x,y)=', fxy:6:3);
end.

```

Dastur ishi bayonnomasi

Ma''lumotlarni kiriting:

2.0 -1.0

x=2.000 va y=-1.000 qiymatlar uchun f(x,y)=-63.000

*rab:=rab*a* ta'minlash operatorida *rab* lokal o'zgaruvchidan foydalanilganligiga e'tibor berish lozim. Bu o'zgaruvchini bevosita funksiya nomi o'rniga foydalanish mumkin emas, ta'minlash operatorining o'ng qismida funksiya nomi xuddi bu funksiyaga murojaat kabi amalga oshirilayapti.

Topshiriq variantlari.

$$1. \text{ Agar } f(u,t) = \begin{cases} u^2 + t^2, & \text{agar } u > 0, t > 0; \\ u + t^2, & \text{agar } u \leq 0, t \leq 0; \\ u - t, & \text{agar } u > 0, t \leq 0; \\ u + t, & \text{agar } u \leq 0, t > 0; \end{cases}$$

funksiya berilgan bo'lsa, $z=f(a,b)+f(a^2,b^2)+f(a^2-1,b)+f(a-b,b)+f(a^2+b^2,b^2-1)$ ni hisoblang

- a) $a=2,5; b=-7,3$
- b) $a=-0,5; b=4,2$
- c) $a=-0,2; b=-0,42$
- d) $a=23,7; b=41,2$

2. Agar $f(u,t) = \begin{cases} u + \sin(t), & \text{agar } u > 0; \\ u + t, & \text{agar } u \leq 0; \end{cases}$

funksiya berilgan bo'lsa, $z=f(\sin \alpha, a)+f(\cos \alpha, a)+f(\sin^2 \alpha, a-1)+f(\sin \alpha \cdot \cos \alpha, a^2-1)+f(\sin^2 \alpha-1, \cos \alpha+a)$ ni hisoblang

- a) $\alpha=\pi/18; a=-2,1$
- b) $\alpha=2,3; a=21,3$
- c) $\alpha=-\pi/14; a=-0,2$
- d) $\alpha=-\pi/10; a=31,2$

3. Agar $f(u,t) = \begin{cases} u + 2t, & \text{agar } u \geq 0; \\ u + t, & \text{agar } u \leq 1; \\ u^2 - 2t + 1, & \text{agar } 1 < u < 0 \end{cases}$

funksiya berilgan bo'lsa, $z=f(\sqrt{|x|}, y)+f(a,b)+f(\sqrt{|x|}+1, -y)+f(|x|-|y|, x)+f(x+y, a+b)$ ni hisoblang

- a) $x=2,31; y=4,2; a=2,5; b=-7,3;$
- b) $x=-4,21; y=-31,2; a=1,2; b=-3,2;$
- c) $x=0,34; y=17,2; a=-4,6; b=-0,44;$
- d) $x=-14,2; y=0,32; a=7,2; b=4,7;$

4. Agar $f(u,t) = \begin{cases} u + t, & \text{agar } u > 1; \\ u - t, & \text{agar } 0 \leq u \leq 1; \\ t - u, & \text{agar } u < 0. \end{cases}$

funksiya berilgan bo'lsa, $z=f(\sin(x)+\cos(y), x+y) + f(\sin(x), \cos(y)) + f(x-y, x)+f(\sin^2(x)-2, a)+f(a+3, b+1)$ ni hisoblang

- | | | | | |
|----|-------------|--------------|------------|-----------|
| a) | $x=\pi/4;$ | $y=0,41;$ | $a=0,1;$ | $b=-2,1;$ |
| b) | $x=0,32;$ | $y=-\pi/10;$ | $a=-0,21;$ | $b=4,2;$ |
| c) | $x=19,2;$ | $y=0,48;$ | $a=-4,3;$ | $b=-6,1;$ |
| d) | $x=0,2\pi;$ | $y=2/3;$ | $a=17,1;$ | $b=0,2;$ |

6.5. 2-topshiriq. Protseduralar qatnashgan dasturlar tuzish

Topshiriq maqsadi:

1. protseduralardan foydalaniladigan dastur tuzish ko'nikmasini olish
2. Protseduraning ma'lumot uzatish mexanizmini o'rganish

Masalaning qo'yilishi.

Bir necha sonlar massivlari berilgan. Massivlar uzunligi topshiriq variantida berilgan. Har bir massivning eng katta va eng kichik elementini topish va chiqarish, so'ngra har bir massivning hamma elementlarini kvadratga ko'tarish va yana eng kichik va eng katta elementlarni topish talab qilinadi. Eng katta va eng kichik elementlarni topishni protsedura ko'rinishida tasvirlang. Protsedurada global parametrlardan foydalaning.

Hisobot mazmuni:

1. masalaning qo'yilishi;
2. dastur matni va natijalar;
3. xulosa.

Uslubiy ko'rsatmalar

1. Massivlarni boshlang'ich ma'lumot sifatida protseduraga uzatish uchun ularni parametr-o'zgaruvchi deb kiritish zarur, chunki bu holda protsedurani chaqirishda lokal massiv tashkil etilmaydi va xotira tejaladi.

2. Qayta ishlanayotgan massivlarning uzunligi har xil bo'lishiga qaramasdan, ular dasturda bir xil turga tegishli qilib tavsiflanadi, shuning uchun protseduraga murojaatda formal va aniq parametrlar turi bir-biriga mos kelishi kerak.

3. Quyida keltirilgan dastur har xil (10, 5, 8) o'lchovli massivlarning elementlarini o'sish tartibida joylashtiradi. Tartiblangan massivlar (5) minimal uzunlikdagi yangi massiv elementlarini hisoblash uchun foydalaniladi. Bu massivning har bir elementi boshqa massivlarning mos elementlarini ko'paytirish yo'li bilan topiladi.

Program UU3;

type a=array[1..20] of real;

var x,y,z,res: a; i,n: integer;

procedure kirit(k: integer; var m: a);

var j: integer;

begin

writeln('Massiv elementlarini kiriting');

for j:=1 to k do read(m[j]);

readln

end;

procedure up(k: integer; var m: a);

var i, j: integer; c: real;

begin

for j:=k downto 2 do

for i:=1 to j-1 do

if m[i]>m[i+1] then

begin

c:=m[i+1];

m[i+1]:=m[i];

m[i]:=c

end;

procedure chiqar(k: integer; var m: a);

var j: integer;

```

begin
  for j:=1 to k do write(m[j]:6:2);
  writeln
end;
begin
  kirit(10,x); kirit(5,y); kirit(8,z);
  up(10,x); up(5,y); up(8,z);
  writeln('Tarriblankach massivlarning ko''rinishi:');
  chiqar(10,x); chiqar(5,y); chiqar(8,z);
  writeln('Natija');
  for i:=1 to 5 do res[i]:=x[i]*y[i]*z[i];
end.

```

Dastur ishi bayonnomasi:

Massiv elementlarini kiriting

1.2 -9.0 5.6 7.8 4.5 4.4 3.5 -5.6 3.2 4.0

Massiv elementlarini kiriting

1.2 4.3 1.0 3.3 3.5

Massiv elementlarini kiriting

-5.6 4.5 -7.8 2.3 4.5 1.2 8.9 2.3

Tarriblankach massivlarning ko''rinishi:

-9.00 -5.60 1.20 3.20 3.50 4.00 4.40 4.50 5.60 7.80

1.00 1.20 3.30 3.50 4.30

-7.80 -5.60 1.20 2.30 2.30 4.50 4.50 8.90

Natija

70.20 37.63 4.75 25.76 34.62

Topshiriq variantlari

1. Mos holda 5 ta va 7 ta haqiqiy komponentalarga ega ikki massivni kiriting va qayta ishlang.

2. Mos holda 3 ta, 6 ta va 8 ta butun komponentaga ega uchta massiv kiriting va qayta ishlang.

3. Mos holda 4 ta, 6 ta, 3 ta va 10 ta butun komponentaga ega to'rt massiv kiriting va qayta ishlang.

4. Mos holda 4 ta va 6 ta haqiqiy komponentaga ega ikki massiv kiriting va qayta ishlang.

5. Mos holda 5 ta, 10 ta va 4 ta butun komponentaga ega uchta massiv kiriting va qayta ishlang.

6. Mos holda 3 ta, 5 ta, 8 ta va 6 ta haqiqiy komponentaga ega to'rt massiv kiriting va qayta ishlang.

7. Mos holda 7 ta va 10 ta haqiqiy komponentaga ega ikki massiv kiriting va qayta ishlang.

8. Mos holda 5 ta, 4 ta va 7 ta butun komponentaga ega uchta massiv kiriting va qayta ishlang.

9. Mos holda 7 ta, 3 ta, 5 ta va 4 ta butun komponentaga ega to'rt massiv kiriting va qayta ishlang.

10. Mos holda 6 ta va 8 ta haqiqiy komponentaga ega ikki massiv kiriting va qayta ishlang.

6.5. 3-topshiriq. Aniq integralni to'rtburchaklar usuli yordamida hisoblash

Topshiriq maqsadi:

1. Sarlavhasida parametr - funksiyaga ega protseduralar yozish ko'nikmasini olish

2. Integrallashning sonli usullarini dasturlash amaliy ko'nikmasini olish.

Masalaning qo'yilishi.

$f_1(x)$, $f_2(x)$, $f_3(x)$ uchta funksiya berilgan. Har bir funksiyaning berilgan oraliqdagi integralini 10-4 aniqlik bilan toping. Topshiriqni bajarish uchun aniq integralni hisoblashning to'rtburchaklar usulidan foydalaning.

Hisobot mazmuni.

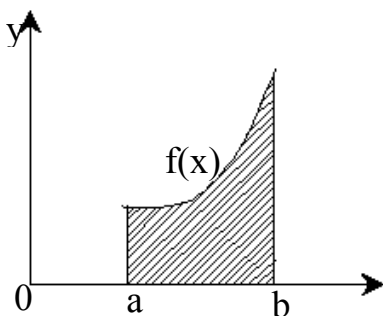
1. masalaning qo'yilishi;
2. integrallashning sonli usuli tavsifi;

3. test va sozlashlar natijalari;
4. dastur matni va natijalar;
5. xulosalar.

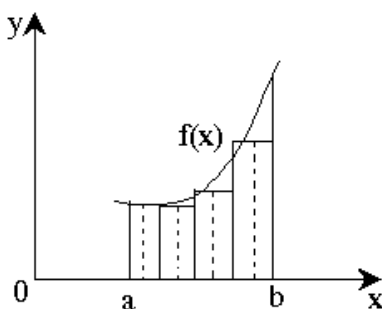
Uslubiy ko'rsatmalar

1. To'rtburchaklar usuli tavsifi. $\int_a^b f(x)dx$ qiymatni

berilgan aniqlikda hisoblash talab qilinsin. $f(x)$ funksiya grafigi 7.1 rasmda tasvirlangan.



7.1 rasm



7.2 расм

Aniq integral qiymatini topish – bu shtrixlangan soha yuzasini topish demakdir. Integralni hisoblashning birinchi yaqinlashishida $[a,b]$ kesmani n ta teng qismlarga bo'lamiz ($n=4$), $f(x_i)$ qiymatlarni hisoblaymiz, bu yerda $x_i=a+h_i-h/2$; $h=(b-a)/n$ (7.2 rasm)

Har bir to'rtburchakning yuzi (s_i ni) hisoblaymiz: $s_i=hf(x_i)$. Bu to'rtburchaklar yuzalarining yig'indisi $s1$ integralning taqribiy qiymatidir:

$$s1 = \sum_{i=1}^n s_i = \sum_{i=1}^n f(x_i)$$

Ammo bitta yaqinlashish hisoblash aniqligini baholashga imkoniyat bermaydi, shuning uchun ikkinchi yaqinlashishni

ham topish kerak. Buning uchun n ni ikki marta oshiramiz, ya'ni $n:=2n$. Xuddi yuqoridagidek $s2$ ni topamiz:

$$s2 = h \sum_{i=1}^n f(x_i)$$

Integral qiymati ξ aniqlikda topish talab qilingan, shuning uchun $|s1-s2|<\xi$ shartni tekshiramiz. Agar bu shart bajarilsa, u holda $s2$ qiymat izlanayotgan integralning qiymati sifatida qabul qilinadi. Aks holda, oxirgi olingan $s2$ qiymat oldingisi deb hisoblanadi, ya'ni $s1:=s2$. So'ngra kesmani bo'lish nuqtalari sonini ikki marta oshiramiz va $s2$ ni yana hisoblaymiz. Bu jarayon $|s1-s2|<\xi$ shart bajarilgunga qadar davom ettiriladi.

2. Integralni to'rtburchaklar yordamida hisoblashni protsedura ko'rinishida ham, funksiya ko'rinishida ham tasvirlash mumkin. Funksiyalar uchun formal parametrlar sifatida integrallash chegaralari, integral ostidagi funksiya, integralni hisoblash aniqligi qabul qilinishi kerak, protsedura uchun esa chaqiriladigan qiymatlar ham qo'shiladi. Masalan, integralni hisoblash funksiyasi sarlavhasi quyidagicha bo'lishi mumkin:

function int(a,b,e: real; function f:real): real;

Integralni hisoblash protsedurasi sarlavhasi quyidagicha bo'lishi mumkin:

procedure w(a,b,e: real; function f:real; var int: real);

yoki parametr-funksiya o'zining parametrlari tavsifini o'z ichiga olgan:

procedure w(a,b,e: real; function f(x: real):real; var int: real);

3. Ushbu torshiriqning har bir variantida uch xil funksiyaning integralini topish talab qilingan. Shuning uchun dasturda ularga mos uchta funksiya tavsiflanishi zarur. Bu funksiyalarning ismlari integral hisrblash protsedurasi (yoki funksiyasining) aniq parametri sifatida ko'rsatiladi.

4. Dastur tuzilishi quyidagi ko'rinishda bo'lishi kerak:

Dastur sarlavhasi;

O'zgaruvchilar tavsifi

Birinchi funksiya tavsifi

Ikkinchi funksiya tavsifi

Uchinchi funksiya tavsifi

Integralni hisoblan protsedurasi (funksiyasi) tavsifi

Operatorlar bo'limi

Topshiriq variantlari

1. $f_1 = \sqrt{e^x - 1}$; $a_1 = 0, 2$; $b_1 = 2, 1$;

2. $f_2 = e^x \sin x$; $a_2 = 0, 0$; $b_2 = \pi/2$;

3. $f_3 = (x^2 - 1)10^{-2x}$; $a_3 = 0, 0$; $b_3 = 0, 5$;

4. $f_1 = x \sqrt[3]{1+x}$; $a_1 = 1, 0$; $b_1 = 8, 0$;

5. $f_2 = \frac{1}{3 + 2 \cos x}$; $a_2 = 0, 0$; $b_2 = \pi/2$;

6. $f_3 = (x^2 - 1)10^{-2x}$; $a_3 = 2, 0$; $b_3 = 2, 7$;

7. $f_1 = \frac{\arcsin \sqrt{x}}{\sqrt{x(1-x)}}$; $a_1 = 0, 2$; $b_1 = 0, 3$;

8. $f_2 = x^3 e^{2x}$; $a_2 = 0, 0$; $b_2 = 0, 8$;

9. $f_3 = \operatorname{tg}^3\left(\frac{x}{2} + \frac{\pi}{4}\right)$; $a_3 = 0, 0$; $b_3 = \pi/8$;

10. $f_1 = x \arctg x$; $a_1 = 0, 0$; $b_1 = 1, 7$;

11. $f_2 = \frac{1}{1 + \sqrt{x}}$; $a_2 = 0, 0$; $b_2 = 2, 0$;

12. $f_3 = \frac{1}{5 - 3 \cos x}$; $a_3 = 0, 0$; $b_3 = 3/2 \pi$;

13. $f_1 = \frac{2}{1 - 4x}$; $a_1 = -2, 0$; $b_1 = -1, 2$;

14. $f_2 = \frac{I}{(x+1)\sqrt{x^2+1}}; \quad a_2=0,0; \quad b_2=0,6;$
15. $f_3 = \sqrt{a^x-1}; \quad a_3=0,2; \quad b_3=1,0.$
16. $f_1 = \sqrt{e^x-1}; \quad a_1=0,1; \quad b_1=1,0;$
17. $f_2 = x^3\sqrt[3]{1+x}; \quad a_2=2,0; \quad b_2=7,0;$
18. $f_3 = \frac{\arcsin \sqrt{x}}{\sqrt{x(1-x)}}; \quad a_3=0,2; \quad b_3=0,3.$
19. $f_1 = x \arcsin x; \quad a_1=0,0; \quad b_1=1,6;$
20. $f_2 = \frac{2^x}{1-4^x}; \quad a_2=-2,0; \quad b_2=-1,3;$
21. $f_3 = \frac{I}{1+\sqrt{2x}}; \quad a_3=0,0; \quad b_3=1,0.$
22. $f_1 = e^x \sin x; \quad a_1=0,0; \quad b_1=\pi/4;$
23. $f_2 = \frac{I}{3+2\cos x}; \quad a_2=0,0; \quad b_2=\pi/6;$
24. $f_3 = x^3 e^{2x}; \quad a_3=0,0; \quad b_3=0,8.$
25. $f_1 = \frac{I}{1+\sqrt{x}}; \quad a_1=0,0; \quad b_1=1,8;$
26. $f_2 = \frac{I}{(x+1)\sqrt{x^2+1}}; \quad a_2=0,3; \quad b_2=0,3;$
27. $f_3 = \sqrt{2^x} I; \quad a_3=0,3; \quad b_3=1,0.$
28. $f_1 = (x^2-1)10^{-2x}; \quad a_3=0,0; \quad b_3=0,4;$
29. $f_2 = \frac{I}{x \ln^2 x}; \quad a_2=2,0; \quad b_2=2,5;$
30. $f_3 = \operatorname{tg}^3\left(\frac{x}{2} + \frac{\pi}{4}\right); \quad a_3=0,0; \quad b_3=\pi/4;$

7-BO'LIM. TO'PLAMLAR TURI

To'plamlar turi biron-bir cheklangan yoki skalyar tur asosida yaratiladi. Asosiy tur sifatida cheklanmagan butun (*integer*) va haqiqiy (*real*) turlarni olish mumkin emas. Umumiy holda ma'lumotlarning to'plamlar turi quyidagicha tavsiflanadi:

type s=set of bs

Bu yerda *bs* asosiy tur *s* to'plamlar turiga tegishli qiymatlarning to'plamini aniqlaydi. Bu to'plamning barcha qisman to'plamlari *s* to'plamlar turiga tegishli o'zgaruvchining qiymatlari bo'la oladilar. Ular asosiy to'plam elementlarining ro'yxati orqali beriladi. Ro'yxatda elementlar orasiga vergul qo'yiladi va butun ro'yxat kvadrat qavs ichiga olinadi. Bo'sh to'plam, ya'ni birorta ham elementga ega bo'lmagan to'plam $\{\}$ ko'rinishda belgilanadi.

7.1. To'plamlar turi o'zgaruvchilari

To'plamlar turi o'zgaruvchilarning tavsiflanishiga namunalar:

Type shahar=(Toshkent, Samarqand, Buxoro);

mg=set of shahar;

Var sh1, sh2: mg;

p2: set of 'A'..'F';

p3, p4: set of 0..9;

sh1 o'zgaruvchi quyidagi qiymatlar qabul qilishi mumkin:

[Toshkent]

[Samarqand]

[Buxoro]

[Toshkent, Samarqand]

[Toshkent, Buxoro]

[Samarqand, Buxoro]

[Toshkent, Samarqand, Buxoro]

//

Umumiy holda, agar asosiy to'plam n elementdan tashkil topgan bo'lsa, hosilaviy to'plamlar turi 2^n ta qismaniy to'plamlarni aniqlaydi. Yuqoridagi misolda *shl* o'zgaruvchi $2^3=8$ ta har xil qiymatlar qabul qilishi mumkin.

Agar to'plam elementlari asosiy to'plamning ketma-ket elementlari bo'lsa, u holda cheklangan tur tavsifidagi kabi faqat birinchi va oxirgi elementlarini ko'rsatish mumkin. Masalan, quyidagi ta'minlash operatorlarida $p2$ o'zgaruvchiga bir xil to'plamli qiymat ta'minlanayapti:

$$p2:=['a'..'c', 'f'];$$

$$p2:=['a', 'b' 'c', 'd'];$$

$['c'..'a']$ to'plam bo'sh to'plam hisoblanadi, chunki elementlarning asosiy turdagi tartibi bo'yicha $'c'$ ning nomeri $'a'$ ning nomeridan katta. To'plam elementlarini asosiy skalyar turdagi ifodalar ko'rinishida ham berish mumkin. Masalan, $k:=7$; $p3:=[1, k+1]$; $p4:=[k-1..8]$ ta'minlash operatorlari bajarilishi natijasida $p3$ to'plam ikki elementdan, $p4$ to'plam esa uchta elementdan ($\{6,7,8\}$) tashkil topadi.

7.2. To'plamlar ustida amallar

To'plamlar ustida to'plamlar birlashmasi (+), kesishmasi (-) amallari aniqlangan. Bu amallar yordamida to'plamlar turiga tegishli har xil ifodalar tuzish mumkin.

Ikki to'plam birlashmasi ularning hech bo'lmaganda biriga tegishli elementlardan iborat to'plamdir. Masalan,

$$['b', 'f'] + ['b'..'d'] = ['b', 'c', 'd', 'f']$$

Ikki to'plamning kesishmasi deb har ikkala to'plamga tegishli elementlardan hosil bo'lgan to'plamga aytiladi. Masalan:

$$[0..4] * [5,6] = []$$

$$[Toshkent, Buxoro] * [Samarqand, Buxoro] = [Buxoro]$$

Ikki to'plamning ayirmasi deb, birinchi to'plamda mavjud bo'lib, ikkinchisiga tegishli bo'lmagan elementlardan hosil bo'lgan to'plamga aytiladi. Masalan,

$$[1,5,9]-[2,4,8,9]=[1,5]$$

To'plamlarni solishtirish uchun munosabat amallaridan foydalaniladi ($=$, $<>$, $<=$, $>=$). Agar a va b to'plamlar elementlari aynan bir xil bo'lsa $a=b$ ifoda rost qiymat qabul qiladi. Agar solishtirilayotgan to'plamlardan birida ikkinchisiga tegishli bo'lmagan bironta element bor bo'lsa, $a<>b$ ifoda rost qiymat qabul qiladi. Agar a to'plamning elementlari bir vaqtda b to'plamning ham elementlari bo'lsa, $a\leq b$ ifoda rost qiymat qabul qiladi. Agar b to'plamning elementlari bir vaqtda a to'plamning ham elementlari bo'lsa, $a\geq b$ ifoda rost qiymat qabul qiladi.

Quyidagi ifodalar ham rost qiymat qabul qiladilar:

$$['c'..'e']=['d','e','c'];$$

$$[7,1,3]<>[2,4,6,8];$$

$$[Samarqand]<=[Toshkent, Samarqand]$$

$$['a'..'e']>=['a','b'];$$

Paskal tilida *in* so'zi bilan ifodalanadigan maxsus amal mavjud bo'lib, u biror elementning biror to'plamga tegishli yoki tegishli emasligiga mantiqiy turdagi (*true* va *false*) qiymat hosil qiladi. Umumiy holda *in* dan chap tomonda asosiy skalyar turga tegishli ifoda yoziladi, o'ng tomonda esa to'plamlar turidagi ifoda ko'rsatiladi.

Masalan, $5 \text{ in } [3..7]$ ifoda *true* qiymat qabul qiladi, chunki $5 \in [3,4,5,6,7]$ to'plamning elementi hisoblanadi. $k+2 \text{ in } [0..7] * [4,6,8]$ ifoda $k=1$ bo'lganda *false* qiymatga ega, chunki o'ng tomonda ko'rsatilgan to'plamlar kesishmasidan hosil bo'lgan $[4,6]$ to'plamda 3 element mavjud emas.

To'plamlar ustida bajariladigan amallarning bajarilish tartibi quyidagicha:

1) * 2) +,- 3) In, =, <>, <=,>=. Bajarilish tartibi bo'yicha bir xil darajaga ega bo'lgan amallar bir guruhga birlashtirilgan. Bu amallar ifodada joylashgan o'rniga bog'liq holda ketma-ket bajariladi. Bajarilish tartibini o'zgartirish uchun qavslardan foydalaniladi.

To'plamlar bilan ishlash vositalari ba'zi hollarda dasturni qisqartirishga va ularni yanada ko'rinarli (наглядность) qilishga imkon beradi. Har xil tekshirishlar soni kamayganligi sababli dastur unumliroq ishlaydi. Masalan, quyidagi dastur berilgan satrdagi raqamlar sonini aniqlaydi va probel, matematik amallar va tinish belgilaridan farqli belgilarni chop etadi:

```
Program MN;  
  var c:char; k:integer;  
  begin  
    k:=0;  
    readln;  
    repeat  
      read(c);  
      if not (c in [' ', ',', '.', ':', ';', '+', '-', '*', '/']) then  
        begin  
          if c in ['0'..'9'] then k:=k+ 1;  
          write(c);  
        end;  
    until eoln;  
    writeln('Satrdagi raqamlar soni', k:2);  
  end.
```

Dasturda *c* belgining qaralayotgan to'plamga tegishlilikini tekshirish paskal tilining to'plam tushunchasidan foydalangan holda bajarilgan. Agar bu tushunchadan foydalanilmasa, har bir tekshirish oshkor holda yozilishi zarur bo'lib, dastur hajmi va ijro vaqti ko'proq bo'lar edi.

MN dasturning ish bayonnomasi:


```
writeln;
writeln('Tinish belgilari');
for j:= '!' to '?' do
  if j in pr then write(j:2);
end.
```

Dastur ishi bayonnomasi:
 => fywa.*12 prllllo.orpa:pr?8;
Lotin harflari
 afloprwy
Tinish belgilari
 . : ; ?

7.3. Savollar va mashqlar:

1. To'plamlar turini yaratishda asosiy tur sifatida qanday turlar ishlatilishi mumkin?

2. To'plamlar turidagi o'zgaruvchi quyidagicha tavsiflangan:

Var pm: set of (yer, oy, mars);

Bu o'zgaruvchining mumkin bo'lgan hamma qiymatlarini yozing.

3. To'plamlar turidagi o'zgaruvchi ustida qanday amallar aniqlangan va ularning bajarilish tartibi qanday?

4. To'plamlar tengmi?

a) [*'a'.. 'd'*] va [*'a', b', 'c', 'd'*];

b) [*yoz, qish*] va [*qish, yoz*].

5. Quyida keltirilgan ifodalardan qaysi birlari paskal tili qoidalarini bo'yicha noto'g'ri va nega?

a) [*'a', b', 'c', 'k'*] / [*'a', 'b'*];

b) [*5,6,1,3*] >= [*1,3,5,6*];

c) [*kub, shar*] + [*prizma*];

d) [*tenglama, sistema*] * [*tengsizlik, sistema*];

e) [*3.5, 7.1, 2.0*] - [*1, 2*].

6. Quyidagi ifodalarni hisoblang

a) [*5*] <= [*1..5*];

- b) $['a'..'d', 'k'..'m'] + ['d'..'k'];$
- c) $[Toshkent, Samarqand, Buxoro] * [Buxoro];$
- d) $[7, 1, 3..6] = [1..7];$
- e) $[',', '(', ')', ':'] - [',', ':'];$
- f) $15 \text{ in } [1..10].$

7. Quyida dastur berilgan. Unga $5a7233111bge2257cde.$ matn boshlang'ich ma'lumot sifatida kiritiladi. '.' belgisi matn oxirini ifodalaydi. Dastur ishini tushuntirib bering.

Program ZAF;

Var z: set of 0..9;

i, k: integer;

si: char;

Begin

readln;

write('=>');

z:=[];

repeat

read(sl);

k:=ord(si)-ord('0');

if k in [0..9] then z:=z+[k];

until k='.';

if z=[] then writeln('Matnda sonlar y''q');

else

begin

write('Matnda quyidagi sonlar bor');

for i:=1 to 9 do

if i in z then write(i:2);

writeln;

end;

end.

8. Quyidagi ifodalarni hisoblang:

a) $['a', 'b', 'c'] = ['b', 'c', 'a'];$

b) $[3, 4, 6] \leq [2..8]$

- c) $[] \geq [1, 2]$
- d) $\text{trunc}(8.3)$ in $[1..9]$
- e) $\text{succ}('c')$ in $['b', 'd']$
- f) $[2] <> [2, 2, 2]$
- g) 16 in $[16]$

9. Ifodaning qiymatini hisoblang:

$$[1..14] * [5, 12..60] + [4..7] - [2..16] * [6]$$

10. n ta belgidan iborat matn berilgan. ('l', 't', 'z', 'x') belgilardan qaysilari berilgan matnda qatnashganligini chop eting. To'plamlar turidan foydalanib dastur tuzing.

11. To'plamlar turidagi ifodalarni soddalashtiring:

- a) $[11..17] * [2] + [7, 17..40] * [2..17] - [2..8]$
- b) $(a-b) * a + (a-b) * b$
- c) $(a+b) * (a-b) * (b-a)$
- d) $a+b-(a-b)-(b-a)$
- e) $a-(a-b)$

7.4. 1-Topshiriq. To'plamlar turidagi qiymatlar bilan ishlash.

Topshiriq maqsadi:

- 1. To'plamlar turidagi o'zgaruvchilarni qo'llash va ular ustida oddiy amallar bajarish ko'nikmasini olish;
- 2. To'plamlar turidagi qiymatlar qo'llaniladigan masalalar bilan tanishish.

Masalaning qo'yilishi

EHM da ishlatiladigan belgilarning cheklangan va tartiblangan to'plami bilan tanishish. Topshiriq variantlardan biri uchun dastur tuzing.

Hisobot mazmuni

- 1. Masalaning qo'yilishi
- 2. Dastur matni va uning bajarilishi natijasi

Uslubiy ko'rsatmalar

Dastur simvollarning ixtiyoriy to'plami uchun to'g'ri ishlashi kerak. Topshiriqni bajarish uchun yuqorida keltirilgan *LAT* dasturi bilan tanishish zarur. Kiritiladigan belgilar satri ekranning satridan uzun bo'lishi ham mumkin. Bu holda dastur *eoln* funksiyasi bilan emas, balki dastur tuzuvchi tomonidan ko'rsatilgan satr oxiri belgisi bilan ishlaydi.

Topshiriq variantlari

Bo'sh bo'lmagan belgilar ketma-ketligi berilgan. Bu ketma-ketlikdan quyida ko'rsatilgan elementlardan tarkib topgan to'plam shakllantirilsin va pechatga chiqarilsin:

1. 0 dan 9 gacha raqamlar;
2. 'a' dan 'f' gacha va 'x' dan 'z' gacha bo'lgan harflar;
3. 'g' dan 'n' gacha harflar va 0 dan 9 gacha bo'lgan raqamlar;
4. Tinish belgilari
5. 'a' dan 'z' gacha harflar va 0 dan 5 gacha bo'lgan raqamlar;
6. 't' dan 'x' gacha harflar va tinish belgilari;
7. 5 dan 9 gacha bo'lgan raqamlar va arifmetik amallar belgilari;
8. Arifmetik amal belgilari va tinish belgilari;
9. Raqamlar va arifmetik amal belgilari;
10. Tinish belgilari va 'e' dan 'n' gacha harflar;
11. Munosabat amali belgilari;
12. 3 dan 9 gacha raqamlar, 'a' dan 'f' gacha harflar va tinish belgilari;
13. Arifmetik amal belgilari va munosabat amali belgilari;
14. 'f' dan 'm' gacha harflar va munosabat amali belgilari;
15. Tinish belgilari va munosabat amali belgilari.

8-BO'LIM. FAYLLAR TURI

8.1. Fayllar bilan ishlash protseduralari va funksiyalari

Dastur bajarilishi natijasida olingan ma'lumotlar hajmi katta bo'lsa va ularni uzoq vaqt saqlab turish kerak bo'lsa, ma'lumotlarni tashqi xotiraga yozish talab qilinadi. Tashqi xotira sifatida qattaq disk va egiluvchan disk (disketalar) ishlatiladi.

Pascal tilida disklar bilan ishlash fayllar ustida amallar bajarishga keltiriladi. Fayl – bu bir xil turdagi ma'lumotlar (komponentlar) ketma-ketligidir. Massivlardan farqli ravishda faylning komponentlari indekslarga ega emas va ularning soni ilgaridan belgilab qo'yilmaydi. Faylning biron-bir komponentini olish uchun oldingi komponentlardan ketma-ket ravishda o'tib borish kerak.

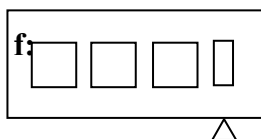
Fayllar turini tavsiflash quyidagicha bo'ladi:

type <ism>=file of <tur>:

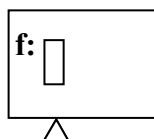
bu yerda <ism> – fayl ismi; <tur> - komponentlar turi. Masalan: *Type V=file of integer;*

Bu yerda *V* fayllar turi tavsiflangan. Bu turga tegishli o'zgaruvchilar komponentlari butun sonlar bo'lgan fayllardan iborat bo'ladi. Masalan, *var f:V;*

Faylning tashqi xotirada joylanishini quyidagi oddiy sxema yordamida tasavvur qilish mumkin:



a



b

Bu yerda *f* – fayl ismi; $\square$ – komponentlar; $\square$ –fayl oxiri belgisi; $\triangle$ – o'qish/yozish qurilmasining boshchasi;

Endi fayllar ustida bajariladigan amallar bilan tanishamiz. Faraz qilamizki, f – fayl ismi, a – turi fayl komponentlari bilan bir xil bo'lgan o'zgaruvchi. Qayd qilamizki, fayl quyidagi uch holatdan birida bo'lishi mumkin:

- yozish uchun ochiq (o'qish uchun yopiq);
- o'qish uchun ochiq (yozish uchun yopiq);
- ham yozish, ham o'qish uchun yopiq.

Faylni yozish uchun ochish. Faylni yozish uchun ochish quyidagicha amalga oshiriladi:

rewrite(<fayl ismi>)

Bu operator bajarilgach fayl bo'sh hisoblanadi va o'qish/yozish qurilmasi boshchasi fayl oxiri belgisi ustida qo'yiladi. (Agar fayl ochilgunga qadar bo'sh bo'lmasa, u holda barcha komponentalar o'chirib tashlanadi).

Faylga yozish. Bu amal dastlab yozish uchun ochilgan va o'qish/yozish qurilmasi boshchasi fayl oxiriga qo'yilgan holda bajarilishi mumkin. Faylga yozish

write(<fayl ismi>, <o'zgaruvchi>);

operatori orqali amalga oshiriladi, bu yerda <o'zgaruvchi> - turi fayl komponentalari turi bilan bir xil bo'lgan o'zgaruvchi.

Bu operatorning bajarilishi natijasida qiymati <o'zgaruvchi> qiymatiga teng bo'lgan yangi komponent fayl oxiriga yozib qo'yiladi. Fayl oxiri belgisi esa yangi komponentdan keyin o'rnatiladi. Faylga yozish amalining bajarilish bosqichlari 2-rasmda tasvirlangan.



Misol. Faraz qilaylik, f - komponentlari integer turiga tegishli bo'lgan fayl bo'lsin. f – faylga 1 dan 100 gacha natural

sonlarning kvadratlarini yozib qo'yadigan dastur bo'lagini keltiramiz.

```
rewrite(f);  
for i:=1 to 100 do  
  begin  
    j:=sqr(i);  
    write(f,j);  
  end;
```

(i va j integer turidagi o'zgaruvchilar)

Faylni o'qish uchun ochish quyidagicha amalga oshiriladi:

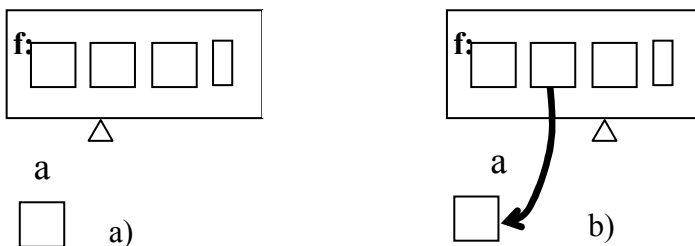
```
reset(<fayl ismi>);
```

bu operator bajarilgach o'qish/yozish qurilmasi boshchasi faylning 1 - komponentasiga o'rnatiladi. Agar fayl bo'sh bo'lsa boshcha fayl oxiri belgisi ustida turadi.

Faylni o'qish. Bu amal fayl dastlab o'qish uchun ochilgan va o'qish/yozish qurilmasi boshchasi biron-bir komponentaning boshlanishga qo'yilgan holatda bajarilishi mumkin. Faylni o'qish

```
read(<fayl ismi>, <o'zgaruvchi>);
```

operatori yordamida amalga oshiriladi, bu yerda <o'zgaruvchi> - turi fayl komponentlari turi bilan bir xil bo'lgan o'zgaruvchi. Bu operatorning bajarilishi natijasida <o'zgaruvchi> va o'qish/yozish qurilmasi boshchasi turgan komponentning qiymati ko'chirib olinadi hamda boshcha navbatdagi komponentga siljiriladi. Agar navbatdagi komponent mavjud bo'lmasa, u holda boshcha fayl oxiri belgisi ustiga qo'yiladi. Faylni o'qish amalining bajarilish bosqichlari 3-rasmda tasvirlangan.



3-rasm

Fayl oxirini topish uchun

EOF(<fayl ismi>);

standart funksiyasi ishlatiladi. Bu funksiya o'qish/yozish qurilmasi boshchasining joylashishiga bog'liq holda True yoki False qiymatga ega bo'ladi:

True – agar boshcha fayl oxiri belgisi ustida joylashgan bo'lsa;

False – boshqa hollarda;

Bu funksiya ko'pincha *if* shartli operator va *while* takrorlash operatorlarida qo'llanilib, o'qish/yozish qurilmasi boshchasi fayl oxiriga yetgan yoki yetmaganligini tekshirish uchun ishlatiladi. Masalan:

if eof(f) then writeln('Fayl oxiri');

Misol. Faraz qilaylik *f* - komponentlari *real* turiga tegishli bo'lgan fayl bo'lsin.

1. Haqiqiy o'zgaruvchilar tavsiflangan.

Faylning komponentlarini ketma-ket o'qiydigan va ularning kvadratlari yig'indisini hisoblaydigan dastur bo'lagini keltiramiz:

reset(f);

s:=0;

while not eof(f) do

begin

```

read(f,t);
s:=sqrt(t)+s
end;

```

Faylni tashqi xotirada saqlab qolish uchun u tashqi ism bilan bog'langan bo'lishi kerak. Bu amal

Assign(<fayl ismi>, <tashqi ism>);

operatori yordamida bajariladi. Bu yerda <tashqi ism> – *string* turidagi o'zgaruvchi yoki apostrof ichiga olingan simvollar ketma-ketligi. Masalan, *assign(f,'f.dat')* deb yozsak, *f* fayl tashqi xotiradagi *f.dat* fayli bilan bog'lanadi. Agar fayl tashqi xotira bilan bog'lanmagan bo'lsa, u vaqtinchalik deb hisoblanadi va dastur ishi tugagach yo'qotiladi.

Faylni yopish amali.

close(<fayl ismi>);

operatori orqali amalga oshiriladi. Agar fayl tashqi ism bilan bog'langan bo'lsa, bu operator bajarilishi natijasida fayl komponentlari tashqi xotirada saqlab qolinadi.

Endi fayllarni ishlatib dasturlar tuzishga doir misollar keltiramiz.

1-misol. Berilgan ixtiyoriy 100 ta simvol tashqi xotiraga *tekst.txt* nom bilan yozib qo'yilsin.

Dastur quyidagicha bo'ladi:

```

Program EZ;
type ft=file of char;
var tekst:ft;
    s:char;
    i:integer;

```

begin

{faylni tashqi ism bilan bog'lash}

assign(tekst,' tekst.txt');

{yozish uchun ochish}

rewrite(tekst);

For i:=1 to 100 do

```

Begin
  {belgini klaviaturadan kiritish}
  read(s);
  {belgini faylga yozish}
  writeln(tekst,s);
end;
close(tekst);
end.

```

2-misol

1-misolda tashkil qilingan tekst.txt fayl komponentlari orasidan kichik lotin harflari ekranga chiqarilsin.

Dastur:

```

Program chiz;
type ft=file of char;
var tekst:ft;
    s:char;
begin
  assign(tekst, ' tekst.txt');
  reset(tekst);
  while not eof(tekst) do
    begin
      {belgini fayldan o'qish}
      read(tekst, s);
      if (s<='z') and (s>='a') then writeln(s);
    end;
  close(tekst);
end.

```

3-misol. Faraz qilaylik 1 dan 200 gacha bo'lgan natural sonlar *son.dat* faylga yozib qo'yilgan. Lekin 99 soni tushirib qoldirilgan. Ana shu sonni *son.dat* faylga o'z joyiga (ya'ni 98 va 100 sonlari o'rtasiga) qo'yish talab qilinadi.

Bu ishni bajarish uchun qo'shimcha *f* fayl olinadi. Dastlab *son.dat* ning 1 dan 98 gacha bo'lgan komponentlari *f* ga ko'chiriladi. So'ngra *f* ga 99 soni va *son.dat* ning qolgan qismi yoziladi. Shundan keyin *f* fayl tulaligicha *son.dat* fayliga ko'chirib o'tkaziladi.

Dastur quyidagicha bo'ladi:

Program P99;

Type t=file of integer;

Var g,f: t;

a, b:integer;

begin

assign(g,'son.dat');

reset(g);

rewrite(f);

{g faylni f ga ko'chirish}

while not eof(g) do

begin

read(g,a);

if a=100 then

begin

b:=99;

write(t,b);

end;

write(f,a);

end;

close(f);

close(g);

{f faylnig ga ko'chirish}

reset(f);

rewrite(g);

while not eof(f) do

begin

read(f,a);

```
write(g,a);  
end;  
close(g);  
end.
```

8.2. Savollar va mashqlar

1. Qanday hollarda fayllardan foydalanish qulay?
2. Fayllar qaerda saqlanadi?
3. Turlar bo'limida fayllar turi qanday beriladi?
4. Komponentlariga murojaat qilish bo'yicha fayllar qanday farqlanadi?

5. Fayllar turi o'zgaruvchisi qanday tavsiflanadi?

6. Agar fayl komponentalari

♦ simvol (belgi)

♦ quyidagi maydonlardan tashkil topgan yozuv

fio: array[1..10] of char;

tugyl: 1960..1980;

ma'lum bo'lsa, fayllar turidagi st o'zgaruvchisini tavsiflang. Mumkin bo'lgan hamma tavsiflash usullarini keltiring.

7. Fayl komponentlariga murojaat qanday amalga oshiriladi?

8. Fayllar ustida qanday amallar aniqlangan?

8.3. 1–topshiriq. Vaqtinchalik fayllar bilan ishlash.

Topshiriq maqsadi:

1. Paskal tilida fayllar turini o'rganish;
2. Dasturda vaqtinchalik fayllarni tashkil etish va ular yordamida axborotni qayta ishlash ko'nikmasini olish.

Masalaning qo'yilishi:

Yaqin qarindoshlaringizning anketali ma'lumotlarini tayyorlang. Har bir qarindoshingiz haqidagi ma'lumotni quyidagi yozuv ko'rinishida rasmiylashtiring:

1. Familiya, ism, otasining ismi;
2. Qarindoshlik darajasi;
3. Yoshi;
4. Tug'ilgan yili;
5. Ishlashi yoki ishlamasligi belgisi;

Ma'lumotlarni klaviatura orqali kiritishni va ularni vaqtinchalik fayllarga yozishni tashkillashtiring. Dasturda qarindoshlarning o'rtacha yoshini hisoblang. Yaratilgan faylni o'qish va urtacha yoshdan kichik bo'lgan qarindoshlar haqida ma'lumot chiqaring.

Hisobot mazmuni:

1. Masalaning qo'yilishi;
2. Qarindoshlarning anketali ma'lumotlari;
3. Dastur matni va uning bajarilishi bayonnomasi;
4. Natijalar;

Ushubiy ko'rsatmalar:

Topshiriqni bajarishda boshlang'ich ma'lumotlarni faylga yozishni quyidagi ikki usuldan biridan foydalanib tashkillashtirish taklif etiladi:

1. muloqot rejimida kiritiladigan yozuvlar soni n ni kiriting va parametrli takrorlash operatori yordamida n ta yozuvni kiritishni tashkillashtiring.

2. So'ng shartli takrorlash operatori yordamida tamom bo'lish belgisi kiritilmagunga qadar yozuvni kiritishni tashkillashtiring.

8.4. 2-topshiriq. Tashqi fayllar bilan ishlash.

1. Tashqi xotirada fayllar tashkil qilish vositalari bilan tanishish;
2. Tashqi fayllar bilan ishlash amaliy ko'nikmasini olish

Masalaning qo'yilishi:

O'qishga kirgan talabalar haqida ma'lumot tayyorlash talab qilinadi. Har bir abituriyent haqidagi ma'lumotni quyidagi maydonlardan tashkil topgan yozuv ko'rinishida rasmiylashtiring:

1. Familiya, ism, otasining ismi
2. Tug'ilgan yili
3. Maktabni tugatgan yili
4. Shahodatnomadagi baholar
5. Yotoqxonaga zarur yoki yo'qligi haqida belgi
6. Kirish imtihonlari ballari

Dasturni boshlang'ich ma'lumotlar yozilgan tashqi fayl bilan ishlashga mo'ljallab tuzing.

1. Tashqi fayldan variantda ko'rsatilgan shartni qanoatlantiruvchi hamma yozuvlarni o'chirib tashlang va faylda qolgan ma'lumotlarni chop eting.

2. Tashqi fayl boshiga (oxiriga) n ta yozuv qo'shing va variantda ko'rsatilgan shart bo'yicha hosil bo'lgan fayl yozuvlarini chop eting.

Hisobot mundarijasi

1. Masalaning qo'yilishi
2. Abituriyentlarning anketali ma'lumotlari
3. Dastur matni va uning bayonnomasi
4. Natijalar

Topshiriq variantlari

I. Boshlang'ich ma'lumotlarni o'z ichiga olgan tashqi fayldan quyidagilarni o'chiring:

1. Hech bo'lmaganda bitta ikki baho olgan abituriyentlarga mos ma'lumotlarni;

2. Hamma baholari uch bo'lgan abituriyentlar haqidagi ma'lumotlarni;

3. Shahodatnomasida hech bo'lmaganda bitta uch baho va o'rtacha bali to'rtidan kichik bo'lgan abituriyentlar haqidagi ma'lumotlarni;

4. O'rtacha bali 4,5 dan yuqori va yotoqxonaga muhtoj bo'lgan abituriyentlar haqidagi ma'lumotlarni;

5. O'rtacha bali to'rtidan kichik bo'lgan abituriyentlar haqidagi ma'lumotlarni;

6. Yoshi 18 dan katta abituriyentlar haqidagi ma'lumotlarni;

7. Bitta ham 5 olmagan abituriyentlar haqidagi ma'lumotlarni;

8. Hamma baholari besh bo'lgan abituriyentlar haqidagi ma'lumotlarni;

9. Hech bo'lmaganda bitta 3 baho va yotoqxonaga muhtoj abituriyentlar haqidagi ma'lumotlarni;

10. Yoshi o'n oltidan katta va a'lo shahodatnoma olgan abituriyentlar haqidagi ma'lumotlarni;

11. Hosil qilingan faylni chop eting.

II. Mavjud tashqi faylga n ta yozuv qo'shing va chop eting:

1. Shahodatnomasidagi hamma baholari besh bo'lgan abituriyentlar ro'yxatini; $n=2$.

2. Shahodatnomasidagi ballaridan faqat bitta 4 va qolganlari 5 bo'lgan abituriyentlar ro'yxatini; $n=3$.

3. O'rtacha bali 4,5 bo'lgan abituriyentlar ro'yxatini; $n=4$.

4. O'rtacha bali 4 bo'lgan abituriyentlar ro'yxatini. $n=1$.
5. Yotoqxonaga zarurati bo'lmagan abituriyentlar ro'yxatini. $n=2$.
6. Yotoqxonaga muhtoj abituriyentlar ro'yxatini. $n=3$.
7. Kirish imtihonlari faqat besh bo'lgan abituriyentlar ro'yxatini. $n=4$.
8. Kirish imtihonlarini besh va to'rt baholarga topshirgan abituriyentlar ro'yxatini. $n=2$.
9. Kirish imtihonlaridan ikkitasini to'rt va qolganlarini 5 bahoga topshirgan abituriyentlar ro'yxatini. $n=2$.