

**O'ZBEKISTON RESPUBLIKASI
OLIV VA O'RTA MAXSUS TA'LIM VAZIRLIGI**

**NIZOMIY NOMIDAGI
TOSHKENT DAVLAT PEDAGOGIKA UNIVERSITETI**

Boqiev R., Mamarajabov M., Ashurov M., Gulyamov J.

**Dasturlash tillari.
(O'quv qo'llanma)**

Toshkent-2008 y.

Ushbu metodik qo'llanma oliy o'quv yurtlarining bakalavr yo'nalishlari talabalari va akademik litsey, kasb-hunar kollejlari o'quvchilari, mustaqil o'rganuvchilar uchun mo'ljallangan.

Qo'llanma Nizomi nomli TDPU ning 2006 yil 30 noyabrdagi ilmiy kengashida ko'rib chiqilgan va nashrga tavsiya etilgan.

Taqrizchilar: Toshkent Temir Yo'l Muxandislari Instituti
dotsenti, Rasulmuxammedov R.

Nizomiy nomidagi TDPU
dotsenti, F-m.f.n Sharipxo'jayeva F.

Mas'ul muharrir: texn.f.d., professor U. Yu. Yuldashev

Muharrir: O. Ergasheva

Kirish

Zamonaviy kompyuterlarda turli dasturlash tillari keng qo'llaniladi. Bu dasturlar iqtisodiyot, boshqarish, xizmat ko'rsatish va ayniqsa sanoat va ishlab chiqarishning turli sohalarida muhim ahamiyatga molik masalalarni hal qilishda ba'zan yagona omilga aylanmoqda. Bu esa o'z navbatida muhandislik va boshqarish sohasi xodimlari uchun kompyuterlardan unumli va oqilona foydalanishni taqozo etadi. ...

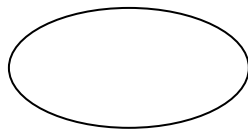
Kompyuterda biror muammoni hal qilish bir necha bosqichlarga bo'linadi. Avvalambor tahlil qilinayotgan jarayon yoki inshootning zarur jihatlarini o'zida mumkin qadar to'la akslantirgan matematik in'ikosi (modeli) tuzib olinadi.

Matematik model formula va tenglamalar tizimi ko'rinishida ifodalanadi. Hosil bo'lgan matematik masalani yechish uchun eng maqbul hisoblash algoritmi tuziladi.

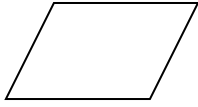
Aniqlangan algoritmgaga xos hisoblash usuli tanlab olinadi va bu usulni kompyuter va foydalanuvchi tushunadigan tilda xotiraga joylashtiriladi.

Biror algoritmik til asosida tuzilgan dastur bo'yicha olingan natijalar tahlil qilinadi va ular asosida o'rganilayotgan jarayon uchun xos bo'lgan umumiy qonuniyatlar aniqlanadi.

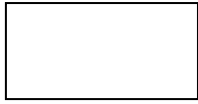
Algoritm — biror masalaning yechilishi uchun zarur bo'lgan buyruqlarning ketma-ketligi bo'lib, odatda blok-sxema shaklida beriladi. Blok-sxemalar quyidagi elementlardan iborat bo'ladi:



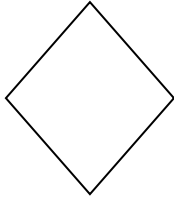
Algoritmning boshlanishi va tugashini bildiradi;



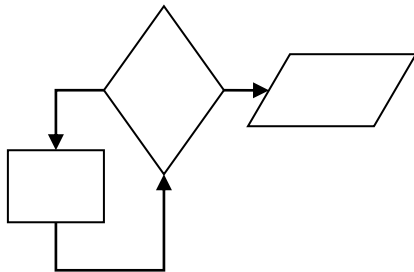
Boshlang'ich ma'lumotlarni kiritish;



Arifmetik ifodalarni hisoblash;



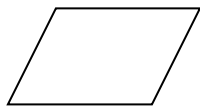
Boshqarishni shart asosida o'zgarishi;



Takrorlanish jarayoni;

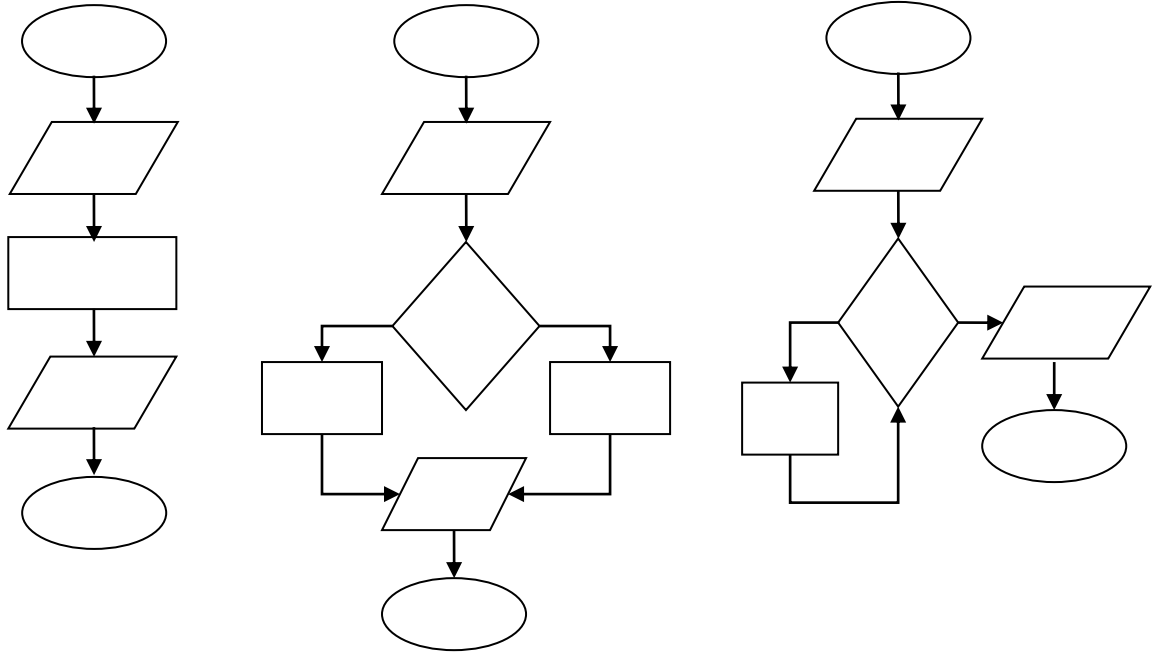


Qism dasturlarga murojaat qilish;



Natijalarni tashqi qurilmalarga chiqarish;

Qayd etilgan blok-sxemalardan foydalanib ixtiyoriy masalani yechish uchun algoritmlar tuzis mumkin, bunda algoritmlarning namunaviy ko'rinishlaridan foydalanish maqsadga muvofiq bo'ladi:



Usbu rasmda chiziqli, tarmoqlanuvchi, takrorlanish jarayonlarining namunaviy blok-sxemalari berilgan.

I Bob. Turbo pascal muhiti bilan tanishish.

1.1. Turbo Pascal muhiti bilan tanishish.

Turbo Pascal dasturi qattiq diskda joylashgan TP\Bin\TPX.exe (Turbo.exe) fayli orqali ishga tushiriladi. Turbo Pascal oynasi ikki xil holatda ishlaydi:

1. Turbo pascal muhiti oynasining umumiy ko'rinishi.

Sarlavha satri

Ishchi soha
(dastur
matnini kiritish
va tahrirlash
oynasi)

Holat satri

The screenshot shows the Turbo Pascal IDE window titled "TPX.EXE". The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The code editor contains the following Pascal code:

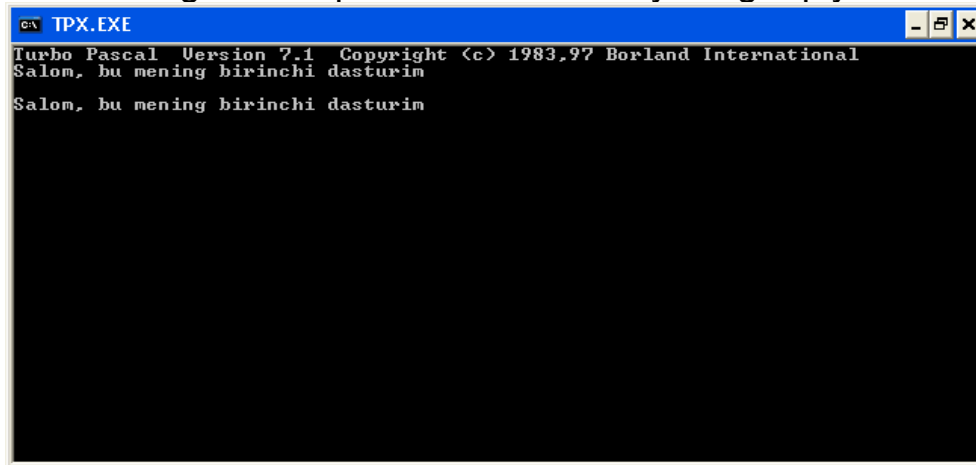
```

Program M_1;
const
  st='Salom, bu mening birinchi dasturim';
begin
  writeln(st);
  readln;
end.
  
```

The status bar at the bottom shows the current cursor position as "7:5" and lists keyboard shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, and Alt+F10 Local menu.

Matn muhariri oynasi dastur matnini kiritish va matnni tekshirish uchun birqancha qulayliklar yaratadi. Turbo.exe fayli ishga tushirilish bilan ekranda Turbo Pascal matn muhariri oynasi paydo bo'ladi.

2. Dastur natijasini ko'rish oynasi. Dastur natijasini olgandan keyin <Enter> tugmasi orqali matn muhariri oynasiga qaytib o'tish mumkin.



1.2. Funksional tugmalar

Turbo Pascal muhitini boshqarish uchun funksional tugmalardan (<F1>, <F2>, ..., <F10>) foydalaniladi. Quyida funksional tugmalar keltirilgan:

<F1> - Turbo Pascal dasturi yordam hizmatidan foydalanish. (Help - Yordam);

<F2> - Dastur matnini faylga saqlash;

<F3> - Faylni ochish;

<F4> - Dasturni ishlashini tekshirish jarayonida qo'llaniladi. Dastur matnidagi kursor o'rnatildan satrgacha bo'lgan satrlarni bajarilishini satrma-satr tekshiradi;

<F5> - Faol oynani butun ekran bo'ylab yoyish;

<F6> - Keyingi tahrirlanuvchi oynaga o'tish;

<F7> - Dasturni ishlashini tekshirish jarayonida qo'llaniladi. Agar dasturda protsedura (funksiya) ga murojaat bo'lsa, protsedura (funksiya) ga kirib, birinchi operatoridan oldin to'xtaydi.

<F8> - Dasturni ishlashini tekshirish jarayonida qo'llaniladi. Dastur matnini sartma-satr bajarilishini tekshiradi. Agar dasturda protsedura (funksiya) ga murojaat bo'lsa, protsedura (funksiya) ga kirmasdan dastur ishlashini tekshiradi;

<F9> - Dasturni kompilatsiya qiladi, lekin ishlashini ta'minlamaydi.

<F10> - Bosh menyu bo'limlari bilan ishlash;

<CTRL>+<F9> - dasturni kompilatsiya qiladi, ishlashini ta'minlaydi va Turbo Pascal muhitiga qaytaradi;

<ALT>+<F5> - dastur matni oynasidan dastur natijalarni ko'rish oynasiga o'tish;

<ALT>+<x> - Turbo Pascal muhitidan chiqish;

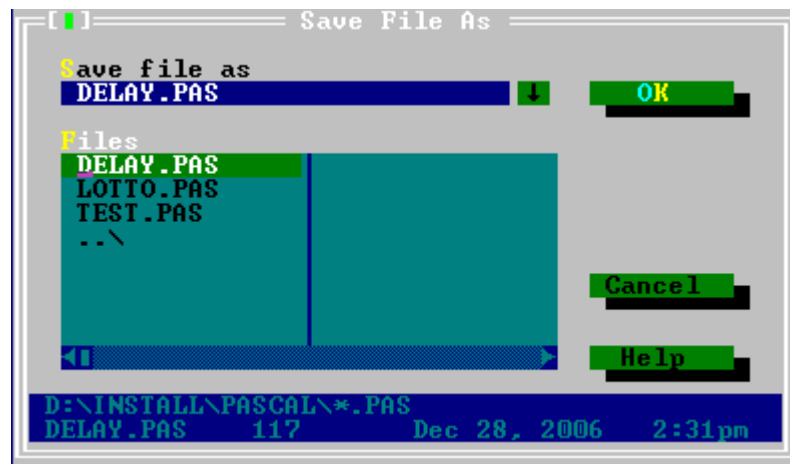
1.3. Dastur matnini kiritishda boshqarish tugmalaridan foydalanish.

Dastur matnini kiritish jarayonida bir qancha boshqarish tugmalaridan foydalaniladi:

- <Enter> - yangi (keyingi) satrga kursorni o'tkazish;
- <PageUp> - bir sahifa yuqoriga;
- <Page Down> - bir sahifa pastga;
- <Home> - satr boshiga kursorni o'tkazish;
- <End> - satr oxiriga kursorni o'tkazish;
- <CTRL>+<PageUp> - matn boshiga o'tish;
- <CTRL>+<Page Down> - matn oxiriga o'tish;
- <Delete> - kursor joylashgan belgini o'chiradi;
- <Backspace> - kursordan chapda joylashdan belgini o'chiradi;

1.4. Fayllar bilan ishlash.

Turbo Pascal muhitida dastur matnini kiritib, natijalar olingandan so'ng, ushbu dasturdan keyinchlik foydalanish uchun dasturni hotirada faylga saqlash zarur bo'ladi. <F2> tugmasi orqali yangi faylni joylashish manzili, nomi ko'rsatiladi va <Enter> tugmasi bosiladi. Diskda saqlangan fayl *.pas kengaytmasini oladi. <F3> tugmasi orqali diskdagi avvaldan mavjud *.pas fayllarni ochish (yuklash) mumkin. Dasturni saqlashda TP katalogidagi tizimli fayllar bilan sizning fayllaringiz aralashib ketmasligi zarur, aks holda siz uchun kerakmas deb hisoblandan fayl Turbo Pascal dasturining tizimli fayli bo'lishi ehtimoli bor. Shuning uchun o'zingiz yaratgan fayllarni TP katalogidan boshqa katalogda saqlash taklif etiladi.



1.5. Turbo Pascal muhiti menyulari.

Fayl menyusi. Fayllar bilan ishlash imkonini beradi:

New – Yangi fayl yaratish;

Open – diskda saqlangan faylni ochish muloqotli oynasini ochadi.

Ushbu amalni <F3> tugmasi orqali bajarish mumkin;

Save – Joriy taxrirlanayotgan dastur matnini faylga saqlash. Ushbu amalni <F2> tugmasi orqali bajarish mumkin;

Save as – Joriy taxrirlanayotgan dastur matnini yangi nom va yangi manzili bilan saqlash muloqotli oynasini chaqiradi;

Print – Dastur matnini chop etish;

Exit – Turbo Pascal muhitidan chiqish;

Fayl menyusidagi boshqa (Save All, Change dir, Print Setup, Dos shell) buyruqlardan juda kam foydalanilishi uchun ushbu buyruqlarni o'tkazib yubordik;

Edit menyusi. Dastur matnini tahrirlash uchun bir qancha buyruqlarni taklif qiladi:

- Undo – Oxirgi amalni bekor qilish (Alt+backspace);
- Redo – Bekor qilingan amalni qayta tiklash;
- Cut – beldilangan dastur qismini “Qirqish” (Shift+Del);
- Copy - beldilangan dastur qismini nusxalash (Ctrl+Ins);
- Paste – Cut yoki Copy buyruqlari yordamida xotiraga olingan dastur qismini kursor turgan joyga o'rnatish; (Shift+Ins);
- Clear – Belgilangan matn qismini o'chirish (Shift+Ins);
- Show clipboard – Bufer xotiraga olingan dastur matni bo'laklarini ko'rish;

Search menyusi. Dastur matnidagi matn bo'laklarini qidirish, hatoliklar mavjud bo'lgan satrga o'tish imkoniyatini beradi:

- Find – Qidirilayotgan matn qismi va qidirish usulini kiritish uchun muloqotli oyna taklif qiladi;
- Replace – Biror matn bo'lagini boshqa matn bo'lagi bilan almashtirish uchun muloqotli oyna taklif qiladi;
- Search again – matn bo'lagini qaytadan qidirish;
- Goto line number – berilgan tartib raqamli satrda o'tish;
- Show last compile error – dastur matnidagi hato satrni va hatolik haqidagi ma'lumotni ko'rsatadi;
- Find error – Dastur matnidagi hato satrni aniqlaydi.
- Find procedure – dastur matnidagi protsedurani qidirish muloqotli oynasini taklif qiladi. Muloqotli oynada dastur matnida ishtirok etuvchi protsedura nomi kitiladi.

Run menyusi. Dasturni ishga tushirishni amalga oshiradi.

- Run – Agar dasturning boshlag'ich matnida o'zgarish kiritilmagan bo'lsa, dasturni ishlashini ta'minlaydi. Agarda dastur matniga o'zgartirish kiritilgan bo'lsa, avval dasturni qayta kompilyatsiya qilish zarurligi haqida so'raladi. Dasturni kompilyatsiya qilinishini bekor qilish uchun <Ctrl+break> tugmalari bosiladi.
- Step over - Qism dasturga kirmasdan dasturning asosiy qismini satrma-satr bajalishini tekshiradi (F8);
- Trace into – Dasturning asosiy qismida qism dasturga murojaat qilinganda qism dasturni satrma-satr bajarilishini tekshiradi (F7);
- Goto cursor – Kursor o'rnatilgan satrgacha dasturni bajarilishini ta'minlaydi;

- Program reset – Yuqoridagi tekshirish jarayolarini bekor qiladi. (Ctrl+F12);

Compile menyusi. Dastur matni kiritilgandan so'ng, uni kompilyatsiyaga yuborish zarur. Ushbu amalni ikki xil usulda amalga oshiriladi:

- Compile menyusidan Compile buyrig'ini tanlash;
- <ALT>+<F9> tugmalarini bosish;

Avval kompilyator dastur matnining sintaktik xatolarini tekshiradi. Agar xatolik aniqlansa, kompilyator xatolikka yo'l qo'yigan satrda o'z ishini to'xtatadi va xatolikka mos keluvchi xabarni beradi. Aniqlangan xatoliklarni dastur tuzuvchi to'g'rilaydi va qaytadan kompilyatsiyaga yuboradi. Dastur matni kompilyatsiyadan muvaffaqiyatli o'tgandan song dastur natijasini olib, natijalarni tahlil qilish imkoniyati paydo bo'ladi.

Debug menyusi. Dastur ishini teksirishda to'xtash nuqtalarini o'rnatish, dasturda ishlatiluvchi o'zgaruvchilarni kiritish va mos qiymatlarini ko'rish imkoniyatlarini beradi;

Tools menyusi: Turbo pascal muhitidan chiqmasdan qo'shimcha uskunalardan foydalanish imkoniyatini beradi;

Options menyusi: Turbo pascal dasturlash muhitini kerali paramtrlarini o'rnatish uchun buyruqlar taklif qiladi.

Window menyusi: Oynalarni ochish, yopish, faollashtirish va ekran maydoniga joylashtirishga yordam beradi.

Help menyusi. Turbo Pascal dasturi muhiti, operatorlari, funksiya-protseduralari, kutubxonalar haqida ma'lumot olish (<ALT>+<H>);

- Index – Yordamchi tizimni tematik ko'rsatkich orqali ko'rsatadi;
- Contents - Yordamchi tizimni mundarijasini ochadi;

II-bob. Turbo-Paskal tilining asosiy tushunchalari.

2.1. Turbo-Paskal tili alifbosi.

Ma'lumki, har qanday tilni o'rganish uning alfavitini o'rgnishdan boshlanadi. Tilni ng alfaviti – shu tilgagina tegishli bo'lgan asosiy belgilari va tushunchalari to'plamidan iborat bo'ladi. Pascal tilining alfavitini taskil etuvchi asosiy belgilar jamlanmasini uch guruhda ajratish mumkin:

- A dan Z gacha bo'lgan lotin harflari va bo'sh joyni aniqlovchi belgi;
- 0 dan 9 gacha bo'lgan arab raqamlari.
- Maxsus belgilar.

Maxsus belgilar o'z navbatida uch turga bo'linadi:

1. +, -, *, / arfmetik amallar: qo'shish, ayirish, ko'paytirish, bo'lish.
2. Solishtirish amallari

Pascal tilida yozilishi	Matematik ko'rinishi	Ma'nosi.
-------------------------	----------------------	----------

=	=	Teng
<>	≠	Teng emas
<	<	Кичик
<=	≤	Kichik yoki teng
>	>	Katta
>=	≥	Katta yoki teng

3. Tinish yoki ajratish belgilari:

. | , | : | ; | (|) | [|] | { | } | ' | ' | :=

Ajratish belgilari vazifasini tilni o'rgnish davomida aniqlab boramiz. Xizmatchi so'zlar guruhi juda keng, shuning uchun bu so'zlarning hammasini birdaniga yodlab, eslab qolish shart emas, balki ulardan foydalanish davomida eslab qolinaveradi. Quyida xizmatchi so'zlar keltirildi:

and, array, begin, case, const, div, do, downto, else, end, for, function, goto, if, in, label, mod va x.zo.

2.2. Turbo-Paskalda nom va identifikatorlar.

Ma'lumki, ma'lumotlarning tahlili jarayonini ifodalovchi algoritm turli xil ob'ektlar (o'zgarmaslar, o'zgaruvchi miqdorlar, funksiyalar va hokazo) ustida ish olib boradi. Bu ob'ektlarga ularning vazifasi va qabul qiladigan qiymatlariga qarab maxsus nomlar beriladi. Shu nomlarni odatda, identifikatorlar deb ataladi. Identifikator deb harf yoki "_" belgisidan boshlanuvchi, harf, raqam va "_" belgisining ixtiyoriy ketma-ketligiga aytiladi.

Xizmatchi so'zlardan identifikator sifatida foydalanish mumkin emas. Odatda identifikator so'zining o'rniga qulayroq va qisqaroq qilib nom deyish mumkin. Dasturda qatnashuvchi ob'ektlarga nomlarni dastur tuzuvchi o'z ixtiyoriga ko'ra tanlab olishi mumkin. Bir xil nom bilan bir necha xil ob'ektlarni nomlash mutlaqo mumkin emas. Turbo Pascal muhitida nomda qatnashuvchi belgilar soni (nom uzunligi) 63 ta belgidan oshmasligi kerak.

Nomlarga misollar:

_Burchak, _A1, Ahmad_Berdiev, C, Summa, Time, A, S1
1m, Kv.yuza identifikator bo'la olmaydi.

2.3. Turbo-Paskal tilida o'zgaruvchilar va ularning turlari.

Odatda, dasturda ishlatiluvchi ma'lumotlar quyidagi tiplarning birortasiga tegishli bo'ladi: butun qiymatli tiplar, haqiqiy qiymatli tiplar, belgili va satrli tiplar, mantiqiy qiymatli va ko'rsatkichli tiplar.

Butun qiymatli tipga tegishli songa misollar:

-1501, 0, 9999.

Butun qiymat qabul qiluvchi o'zgaruvchilarni e'lon qilish uchun Integer, ShortInt, Byte, LongInt va Word xizmatchi so'zlaridan foydalanish mumkin.

Haqiqiy qiymatli tipga tegishli sonlarga misollar:

25.0956, 6.75, -321.936, 1.2E 02, -3.57E-01

Haqiqiy (kasr) qiymatli tipga tegishli o'zgaruvchilarni e'lon qilish uchun Real, Single, Double, Extended va Comp xizmatchi so'zlaridan foydalanish mumkin.

Hamma harflar, belgi va raqamlar, masalan A, b, ", !, \$, S belgili tipga tegishlidir. Belgili tipni qabul qiluvchi o'zgaruvchilarni e'lon qilish uchun Char xizmatchi so'zidan foydalanish mumkin. Belgilarning ixtiyoriy yig'ilmasi (ketma-ketligi) satrlar deb ataladi. Misol 'Axmad', '\$25', '_START'. Satr xatto bo'sh ham bo'lishi mumkin (' '). Bu tipdagi o'zgaruvchilarni e'lon qilish uchun String xizmatchi so'zidan foydalaniladi. Mantiqiy o'zgaruvchilar faqat True (rost) va False (yolg'on) qiymatlarining bittasinigina qabul qilishi mumkin. Bu tip o'zgaruvchilarini e'lon qilish uchun Boolean hizmatchi so'zi ishlatiladi.

Yuqorida sanab o'tilgan tiplar haqida to'liqroq ma'lumotlar keltirib o'tamiz.

Butun qiymatli tiplarning barchasi quyidagi jadvalda keltirilgan:

Tip ko'rinishi	Mazkur tipli o'zgaruvchilarning qabul qiladigan qiymatlar oralig'i	O'zgaruvchilarning kompyuter xotirasida dan egallaydigan joyi
<i>ShortInt</i>	-128..127	8 bit
<i>Integer</i>	-32768..32767	16 bit
<i>LongInt</i>	-2147483648.. 2147483647	32 bit
<i>Byte</i>	0..255	8 bit
<i>Word</i>	0..65535	16 bit

Bu sanab o'tilgan tiplar o'zlarining qiymatlar qabul qilish oralig'i va xotiradan egallagan joyining katta yoki kichikligi bilan farqlanadi. Shuning uchun, o'zgaruvchilarning qabul qiladigan qiymatlarini katta yoki kichikligiga qarab, yuqoridagi tiplardan mosini tanlash maqsadga muvofiqdir.

Endi shu tipdan foydalanishga doir quyidagi misolni ko'rib chiqaylik:

Berilgan m va n butun sonlari ustida quyidagi arifmetik amallar bajarish dasturini tuzing: m n, m-n, m*n. Umuman Paskal tilida dastur tuzish unchalik murakkab emas, hozir shuni amalda ko'rsatamiz. Sistemali qavs { } lar ichiga turli izoh va tushuntirishlar yozib, ular bilan dasturni jixozlaymiz.

Haqiqiy sonlar matematika kursidan ma'lum bo'lgan oddiy o'nlik kasr sonlardir. Agar komp'yuteringiz matematik soprotsessorli bo'lsa quyidagi jadvalda sanab o'tilgan barcha tiplar o'rinli bo'ladi:

Tip ko'rinishi	Mazkur tipli o'zgaruvchilarning qabul qiladigan	O'zgaruvchilarning kompyuter xotirasida dan
----------------	---	---

	qiymatlar oralig'i	egallaydigan joyi
<i>Real</i>	2.9E-39..1.7E38	6 байт
<i>Single</i>	1.5E-45..3.4E38	4 байт
<i>Double</i>	5.0E-324..1.7E308	8 байт
<i>Extended</i>	3.4E-4932..1.1E4932	10 байт
<i>Comp</i>	-9.2E18..9.2E18	8 байт

Agar kompyuterda matematik soprotsessor bo`lmasa, faqat Real tipinigina ishlatish mumkin. Haqiqiy sonlarga doir quyidagi misolni ko`raylik:

Belgili tipli o`zgaruvchilar Char xizmatchi so`zi bilan e'lon qilinib, bu tipning qiymatlari xotiradan 1 bayt joy egallaydi. Paskal tilining barcha belgilari bu tipning qiymatlar sohasiga tegishlidir. Belgili qiymatni (apostrof) belgisi ichiga olib, yoki # belgisidan keyin uning ASCII kodini yozib aniqlash mumkin.

Misol: 'A', yoki #60.

Satr - bu (apostrof) belgisi ichiga olib yozilgan belgilarning oddiy ketma-ketligidir:

'Ab21#9!cd', 'Toshbayev Dilmurod'.

Satr bo`sh yoki bitta belgili bo`lishi ham mumkin. Satrli o`zgaruvchi uzunligi 255 gacha bo`lgan belgili qiymatlarni qabul qilishi mumkin.

Umuman olganda, har bir satrli o`zgaruvchiga xotiradan 256 bayt joy ajratiladi. Xotirani tejash uchun, satrning tipini quyidagicha ko`rsatish maqsadga muvofiqdir: String[N], N - satrdagi belgilar soni. Bu holda belgili o`zgaruvchi uchun N bayt joy ajratiladi.

Belgilar va satrlar ustida bir qancha amallar bajarish mumkin, ya'ni satrdan kerakli bo`lakni kesib olish, satrlarni bir-biriga qo`shish va natijada yangi satrlar hosil qilish. Satrlar haqidagi to`liq ma'lumotni kerakli bo`limdan olish mumkin.

Paskal tilida mantiqiy tip boolean standart nomi bilan aniqlanadi. Mantiqiy tipli o`zgaruvchilar faqat ikki xil qiymat: True(rost) va False (yolg'on) larnigina qabul qilishi mumkin. Mantiqiy tipli qiymatlar ham tartiblangan, ya'ni False<True.

Paskal tilida asosan quyidagi uchta mantiqiy amaldan ko`proq foydalaniladi: not - rad etmoq, and - mantiqiy ko`paytirish, or - mantiqiy qo`shish.

Bu amallarni faqat mantiqiy o`zgarmaslar ustidagina ishlatish mumkin va natijada yana mantiqiy o`zgarmas hosil bo`ladi. Quyida mantiqiy o`zgarmaslar ustida amallar jadvali ko`rsatilgan:

Mantiqiy ko'paytirish	Mantiqiy qo'shish	Mantiqiy inkor etish
True and true = true	true or true = true	not true = false
True and false = false	true or false = true	not false = true

False and true = false	False or true = true	
False and false = false	False or false = false	

2.4. Matematik funksiyalar.

1) ABS(x) x-kattalikning moduli;

2) SQR(x) x-ning kvadrati;

(ABS(x) va SQR(x) funksiyalarning natijalari tipi x-argument tipiga mos)

3) SQRT(x) - x sonning kvadrat ildizi;

4) exp(x) - eksponent e^x , $e \approx 2,71828$;

5) ln(x) - natural logarifm;

6) sin(x) - sin x;

7) cos(x) - cos x;

8) Arctan(x) - arctg x;

(sin(x), cos(x), arctan(x) - funksiyalarning argumentlari radian o'lchovda qabul qilinadi)

Misol.

$\sqrt[3]{x} = y$ funksiya qiymatini hisoblovchi dastur tuzing. Buning uchun $x^n \Rightarrow \exp(n * \ln(abs(x)))$ formulasidan foydalanib, $y = \exp((1/3) * \ln(abs(x)))$ ni topamiz.

III bob. Pascal tilining operatorlari va ularning imkoniyatlari

3.1. Operator tushunchasi va uning turlari

Operator tushunchasi tilning eng asosiy tushunchalaridan biri bo`lib, har bir operator tilning yakunlangan jumlasini hisoblanadi va ma'lumotlar tahlilining tugallangan bosqichini ifodalaydi. Operatorlarni ikki guruhga ajratish mumkin.

1-guruh operatorlarining tarkibida boshqa operatorlar qatnashmaydi va bu operatorlarni asosiy operatorlar deb ataladi. Asosiy operatorlar jumlasiga quyidagi operatorlar kiradi: o`zlashtirish operatori, protsedura operatori, o`tish operatori, bo`sh operator.

2-guruh operatorlarining tarkibida esa boshqa operatorlar ham qatnashib, ular tarkibiy operatorlar deb ataladi. Ular jumlasiga quyidagi operatorlar kiradi:

tashkiliy operator,
tanlov operatori,
takrorlash operatori,
ulash operatori.

Masalani yechish algoritmidagi yuqoridagi ikki guruh operatorlarining ketma-ketligi cheklanmagan miqdorda qatnashishi mumkin. Bu ketma-

ketlikdagi operatorlar o`zaro ";" ajratish belgisi orqali ajratiladi, ya'ni dastur matnining yozuvi alohida operatorlarga bo`linadi. Shunday qilib, S orqali ixtiyoriy yozish mumkin bo`lgan operatorni belgilasak, masala yechilishining algoritmi quyidagi ketma-ketlik bo`yicha ifodalanishi mumkin:

S; S; ...;S.

Operatorlarning bu ketma-ketligi ularning dasturda yozilish tartibi bo`yicha bajariladi. Shunday qilib, operatorning izdoshi undan keyin yozilgan operator hisoblanadi. Operatorlar bajarilishining bu tabiiy ketma-ketligini faqat o`tish operatori yordamida buzish mumkin. Tarkibiy operatorlarda esa operatorlarning bajarilish tartibi o`ziga xos qoidalar bilan aniqlanadi.

Klaviaturadan kiritish operatorlarining umumiy ko`rinishlari quyidagicha:

1.Read (<Kiritiladigan ob'ektlarning ro'yxati>)

Masalan: read(a); read(b); read(c);

2.Readln (<Kiritiladigan ob'ektlarning ro'yxati>)

a) Readln (a,b,c); (Bunda foydalanuvchi a,b,c – larning qiymatini bir satrga bo`shliq belgisi bilan ajratib kiritib, «enter» tugmasini bosadi yoki a,b,c - larning qiymatlarini har birini alohida satrdan kiritadi va « enter» tugmasini bosadi)

b)Readln(a), Readln(b); readln(c); (Bunda foydalanuvchi a,b,c – larning qiymatini bir satrdan kiritib «enter» tugmasini bosadi.).

Dastur natijasini monitorga chiqarish operatorining umumiy ko`rinishi quyidagicha:

1.WRITE (<CHiqariladigan ob'ektlarning ro'yxati>)

Masalan:write (a); write (b); write (c);

write (a,b,c);

2.WRITELN (<CHiqariladigan ob'ektlarning ro'yxati>);

Masalan:

a) WRITELN(a,b,c);

b) WRITELN(a); WRITELN(b); WRITELN(c);

Write (a,b,c) va WRITELN(a,b,c) operatorlarining farqini o`rganish uchun quyidagi ikki dasturni kuzatamiz

Var a,b,c:integer; begin read(a,b,c); write(a,b,c); end.	Var a,b,c:integer; begin read(a,b,c); writeln(a,b,c); end.
Foydalanuvchi a,b,c qiymatlarini quyidagicha kiritishi mumkin) 34 45 56 (oladigan natija ko`rinishi) 344556-(kursor)	(foydalanuvchi a,b,c qiymatlarini quyidagicha kiritishi mumkin) 34 45 56 (oladigan natija ko`rinishi) 344556 -(kursor)

Bu holatlardagi kursor mavqeiga e'tibor bering va natijalarni farqlash uchun dastur tuzuvchi quyidagi ko'rinishda operatorlarni kiritishi lozim:

```
WRITELN(a, ' ', b, ' ', c);
```

yoki

```
WRITELN('a= ', a, 'b= ', b, 'c= ', c);
```

```
Program m_1;  
var x,y: real;  
begin  
  readln(x);  
  Y:= exp((1/3)*ln(abs(x)));  
  writeln(y);  
end.
```

3.2. Chiziqli dasturlarga doir misollar.

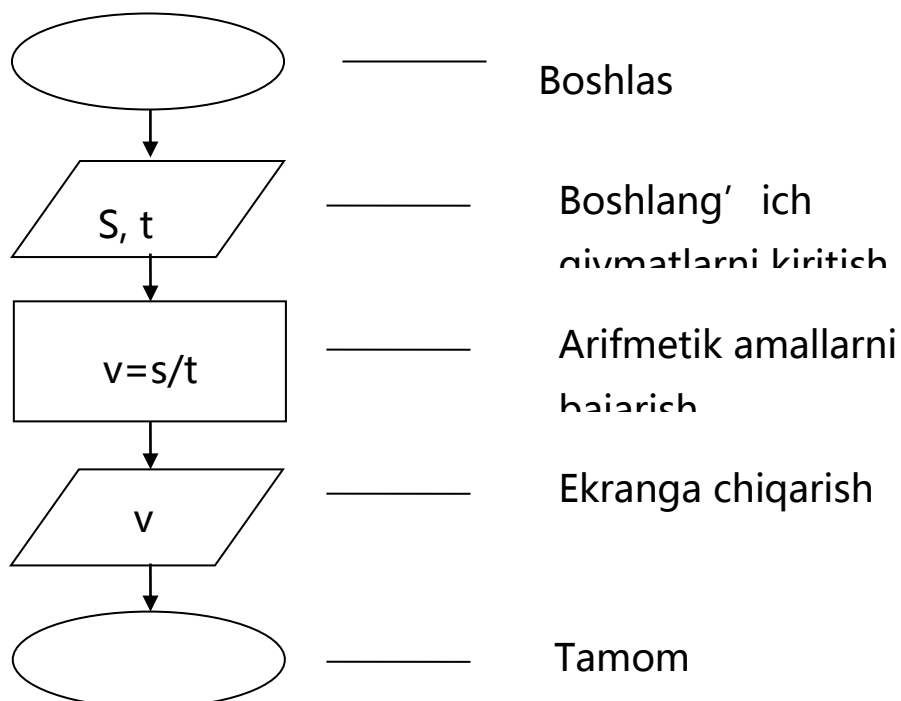
Chiziqli dasturlarga doir masalalar ko'rib chiqamiz:

1-misol. Yo'lovchining bosib o'tgan masofasi va shu masofani bosib o'tish uchun sarflagan vaqtiga ko'ra tezligini topish dasturini tuzing.

Ushbu masalani hal qilish uchun:

1. Matematik model. $v = \frac{s}{t}$

2. Algoritm tuzamiz:



3. Ushbu algoritm asosida dastur matni quyidagicha:

```
Program M_1;  
var  
  s,t:byte;  
  v:real;
```

```

begin
  writeln('Bir satrda probel orqali masofani va vaqtni butun sonlarda
kiriting:');
  read(s,t);
  v:=s/t;
  writeln('Yo''lovchining tezligi:', v:4:2);
  readln;
end.

```

2-misol. Uchburchak tomonlarining koodinatalariga ko'ra yuzini aniqlash dasturi.

Usbu masalani hal qilish uchun ikki nuqta orasidagi masofani topish formulasidan foydalanamiz:

$$a = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

```

program M_2;
var x1,x2,x3,y1,y2,y3,a,b,c,s,p : real;
begin
  write('x1=');readln(x1);
  write('y1=');readln(y1);
  write('x2=');readln(x2);
  write('y2=');readln(y2);
  write('x3=');readln(x3);
  write('y3=');readln(y3);
  a:=sqrt(sqr(x1-x2)+sqr(y1-y2));
  b:=sqrt(sqr(x2-x3)+sqr(y2-y3));
  c:=sqrt(sqr(x3-x1)+sqr(y1-y3));
  p:=(a+b+c)/2;
  s:=sqrt(p*(p-a)*(p-b)*(p-c));
  write('s=',s);
  readln;
end.

```

3-misol. To'g'ri to'rtburchak dioganallari uchlarining koodinatalariga ko'ra uning yuzasini hisoblash dasturi:

```

program M_3;
var x1,x2,y1,y2,s,a,b : real;
begin
  write('x1=');readln(x1);
  write('y1=');readln(y1);
  write('x2=');readln(x2);
  write('y2=');readln(y2);
  a:=abs(x2-x1);

```

```
b:=abs(y2-y1);
s:=a*b;
write('s=',s);
readln;
end.
```

3.3. Tarmoqlanuvchi operatorlar. Tanlash operatori. SHartsiz va shartli o'tish operatorlari.

Kompyuter hisoblash jarayonida ba'zi hollarda boshqarishni dasturning biror qismidan ikkinchi qismiga ko'chirishga to'g'ri keladi. Bunday vazifani bajarish uchun paskal tilida quyidagi operatorlardan foydalanish mumkin:

1. Case k of
2. If B then A else C;
3. If B then goto M;

Ushbu operarotlar bilan birma-bir tanishib chiqaylik.

CASE shartli o'tish operatori.

Bu operatordan dasturda 3 ta va undan ortiq shartlar asosida ishlashga to'g'ri kelganda foydalanish maqsadga muvofiqdir.

Tanlash operatorining umumiy ko'rinishi quyidagicha:

```
case <tanlash uchun parametr> of  
<tanlov holati>  
else  
<operator>  
end;
```

holatga ko'ra else <operator> dasturda qatnashmasligi ham mumkin.

1-Masala. Hafta kuni tartib raqami kiritishi bilan, ushbu kundagi mashg'ulotlar jadvalini ko'rsatuvchi dastur tuzilsin.

```
program m_1;  
var k: word;  
begin  
  Writeln(«Hafta kuni tartib raqamini kiriting»);  
  Readln(k);  
  case k of  
    1:Writeln(`YAkshanba - dam olish kuni`)  
    2:Writeln(`Dushanba - Algebra, Informatika, Ona tili`)  
    3:Writeln(`Seshanba - Geometriya, Fizika, Ximiya`)  
    4:Writeln(`CHorshanba - CHet tili, Jis. tarbiya`)  
    5:Writeln(`Payshanba - Geometriya, Adabiyot, Rus tili`)  
    6:Writeln(`Juma - Algebra, Ximiya, Adabiyot`)  
    7:Writeln(`SHanba - Tarix, Geografiya, Biologiya`)  
    Else Writeln(`Haftada 7 kun bor`)  
  end;  
end.
```

2-Masala. Kiritilgan K1 va K2 sonlar bilan qo'shish, ayirish, bo'lish, ko'paytirish belgilaridan foydalanish holatiga ko'ra beriladigan natijalarni aniqlovchi dastur tuzing.

```
label 10;
var
belgi:char;
k1,k2:real;
begin
  readln(belgi);
  readln(k1,k2) ;
  case belgi of
    '+':writeln('SUMMA ',K1+K2);
    '-':WRITELN('AYIRMA',K1-K2);
    '/':WRITELN('BO''LINMA',K1/K2);
    '*':WRITELN('KO''PAYTMA',K1*K2);
  end; {case operatori tugadi}
end.
```

```
+
24
45
SUMMA 6.9000000000E+01
*
12
10
KO'PAYTMA 1.2000000000E+02
/
450
45
BO'LINMA 1.0000000000E+01
-
345
2000
AYIRMA-1.6550000000E+03
```

SHartsiz o'tish operatori (GOTO).

SHartsiz o'tish operatori bajarilib kelayotgan operatorlar ketma-ketligini o'zgartirib, belgilangan satrga shartsiz o'tishni ta'minlaydi. Belgilangan satr avval aniqlanib, so'ngra shartsiz o'tish operatori GOTO orqali ko'rsatilishi kerak.

Turbo Paskalda satrni belgilash 2 usulda bajariladi:

- 1) 0 dan 9999 gacha bo'lgan butun sonlar orqali.
- 2) harfiy ifoda orqali.

SHartsiz o'tishda foydalanadigan barcha satrlar avval label (shartsiz o'tish satrlarini belgilovchi operator) operatoridan so'ng yozilishi kerak. Dastur jarayonida 2 ta satr 1 ta nomer bilan aniqlanmasligi kerak. Nomer yozilgach ikki nuqta quyiladi.

Misol. Berilgan butun sonning absolyut qiymatini hisoblovchi dastur tuzing.

```
program absolyut;  
    label ST;           {o'tiluvchi satrni aniqlash}  
    var X, Y: integer;  
begin  
    ST:writeln('Biror son kiriting');  
    readln(x);  
    Y:=ABS(x);  
    writeln('Bu sonning absolyut qiymati',Y );  
    goto ST;           {aniqlangan satrga o'tish}  
end.
```

Bu dastur uzluksiz davom etadi. Dasturni bajarilishini to'xtatish uchun Ctrl+Break tugmalari bosilishi kerak.

SHartli o'tish operatori.

SHartli o'tish operatori bilan berilgan shart bajariligi yoki bajarilmasligiga qarab dastur bajarilishini o'zgartiradi.

Operator to'liq va to'liqmas ko'rinishga ega:

1) To'liq bo'lmagan ko'rinishi:

if C then A;

2) to'liq ko'rinishi

if B then A else C;

Bu erda: B – mantiqiy ifoda;

Agar B mantiqiy ifoda rost qiymat qabul qilsa, A ixtiyoriy ifoda bajariladi, aks holda C ixtiyoriy ifoda bajariladi.

Operatorlardan foydalanishga doir misollar.

1) If $K > 0$ then $T := \text{SQRT}(K)$; (to'liqmas ko'rinishi)

Bu misolimizda K ifodani noldan kattaligi tekshiriladi va shu shart bajarilgandagina K ifodadan kvadrat ildizi olinib, T ifodaga o'zlashtiriladi.

2) If $A > B$ then $C := A$

else $C := B$; (to'liq ko'rinishi)

Avvalgi misol kabi avval A ifodani B ifodadan kattaligi tekshiriladi. Agar tekshirilgan shart bajarilsa, C ifodani A ifoda qiymati o'zlashtiriladi. SHart bajarilmagan holda C ifoda B ifodani qiymatini o'zlashtiradi.

3) Berilgan ikkita sonni nisbatini hisoblovchi dastur tuzing. Agar shart tekshirilish jarayonida mantiqiy amallar qo'llansa shartlar qavs ichiga olib yoziladi: If $(A > B)$ AND $(B > C)$ then $D := A$;

1-misol. Foydalanuvchi tomonidan kiritildan ikkta butun sonning bo'linmasini topish dasturini tuzing. Agar bo'linuvchi 0 ga teng bo'lsa, "Bo'luvchi 0 ga teng" ogohlantirishini bersin.

```
program m_1;
var
A,B: integer;
C: real;
begin
  writeln(`Bo`linuvchi va bo`luvchini qiymatini kiriting`);
  readln(A,B);
  if B=0 then writeln (`bo`luvchi nolga teng`)
else
  writeln (`bo`linma`,A/B:4:2);
end.
```

2-misol. Uchta sonning kattasini topish dasturini tuzing.

```
program M_2;
uses crt;
var a,b,c,max : integer;
begin
  clrscr;
  write('a=');readln(a);
  write('b=');readln(b);
  write('c=');readln(c);
  if (a>b) and (a>c) then max:=a;
  if (b>a) and (b>c) then max:=b;
  if (c>a) and (b<c) then max:=c;
  write('max=',max);
  readln;
end.
```

3-misol. Uchburchak uchlari koordinatalariga ko'ra, uchburchakni teng yonli ekanligini aniqlovchi dastur tuzing.

```
program M_3;
uses crt;
var x1,x2,x3,y1,y2,y3,a,b,c : real;
begin
  clrscr;
  write('x1=');readln(x1);
  write('y1=');readln(y1);
  write('x2=');readln(x2);
  write('y2=');readln(y2);
  write('x3=');readln(x3);
  write('y3=');readln(y3);
```

```

a:=sqrt(sqr(x1-x2)+sqr(y1-y2));
b:=sqrt(sqr(x2-x3)+sqr(y2-y3));
c:=sqrt(sqr(x1-x3)+sqr(y1-y3));
if (a=b)or(a=c)or(b=c) then write('teng yonli')
else write('teng yonli emas');
readln;
end.

```

4-misol. $(x_3; y_3)$ nuqtaning berilgan $(x_1; y_1)$ va $(x_2; y_2)$ nuqtalardan o'tuvchi to'g'ri chiziqqa tegishiligi aniqlash dasturi

```

program M_4;
uses crt;
var x1,x2,x3,y1,y2,y3 : real;
begin
  clrscr;
  write('x1=');readln(x1);
  write('y1=');readln(y1);
  write('x2=');readln(x2);
  write('y2=');readln(y2);
  write('x3=');readln(x3);
  write('y3=');readln(y3);
  if (x3-x1)*(y2-y1)-(y3-y1)*(x2-x1)=0
  then write('tegishli')
  else write('tegishi emas');
  readln;
end.

```

3.4.Takrorlanish operatorlar.

Takrorlash jarayonlarini hisoblash uchun uch xil ko'rinishdagi operatorlar mavjud:

1. While B do S; {Sharti oldin beriladigan takrorlanish operatori}
2. Repeat S Until B; {Sharti keyin beriladigan takrorlanish operatori}
3. For i:=a to a1 do {Parametrlil takrorlanish operatori}

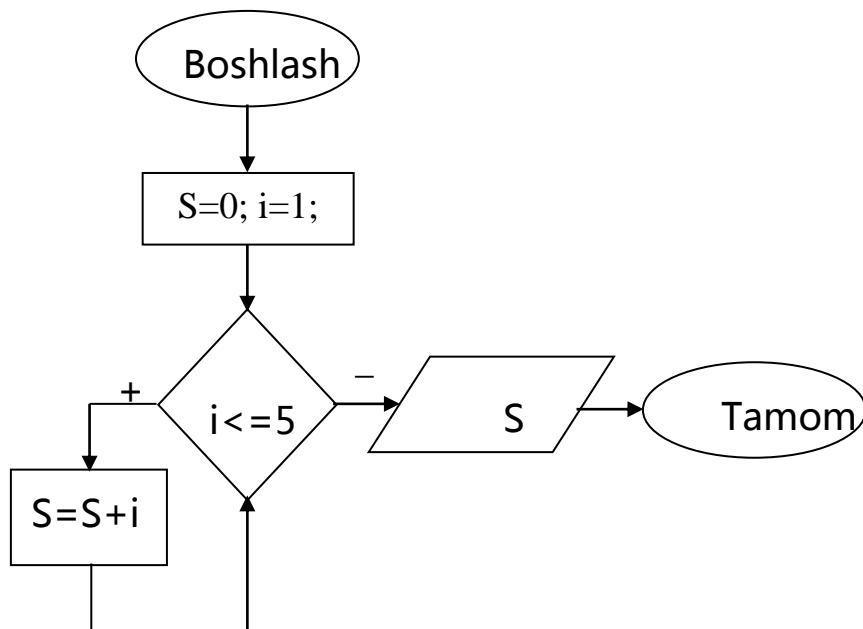
While B do S; { takrorlanuvchi operatori.}

*Bu yerda mantiqiy **B** - mantiqiy ifoda, **S** - ixtiyoriy ifoda yoki operatorlar majmuasi. Bu operator "... bo'lguncha bajar ..." ma'nosini bildiradi, ya'ni **while** operatoridan keyingi mantiqiy ifoda **B** yolg'on qiymat qabul qilgunicha **do** operatoridan keyingi qaralayotgan operatorlar tanasi hisoblanmish **S** qaytarilishi davom etaveradi.*

1- misol. 1 dan N gacha bo'lgan sonlarni yig'indisini topish dasturini tuzing.

Ushbu misolni matematik ko'rinishini (xususiyl holda $N=5$ uchun) yozib olamiz. $S=1+2+3+4+5$;

Keyingi masala uni algoritmini tuzib olamiz;



Endi yuqoridagi algoritmgaga moslab While B do S takrorlanish operatoridan foydalanib dastur tuzamiz.

```

program M_1;
Const n=5;
Var
  s,i:word;
begin
  s:=0; i:=1;
  while i<=5 do
  begin
    s:=s+i;
    i:=i+1;
  end;
  writeln('1 dan ', n, ' gacha bo"lgan sonlarning yig"indisi:', s);
  readln; end.
  
```

2-misol. Ifoda qiymatini hisoblovchi dastur tuzing

$u = x/2 - x^2/4 + x^3/8 - x^4/16 + \dots - x^{10}/1024$;

```

program misol_1;
var
  k,a,x: integer;
  y: real;
begin
  k:=-1;
  y:=0;
  a:=2;
  Readln(x);
  While a<=1024 do
  begin
  
```

```

        y:=(-1)*k*x/a; a:=2*a; k:=(-1)*k;
        end;
        writeln (y); readln;
    end.

```

3-misol. Ifoda qiymatini hisoblovchi dastur tuzing

$$y=1,1+1,01+1,001+\dots+1,000000001$$

```

program misol_3;
    var
        x,y: real;
    begin
        y:=0;
        x:=0,1
        while x>1e-9 do
            begin
                y:=y+(1+x);
                x:=x*0,1;
            end;
        writeln (y);
        readln;
    end.

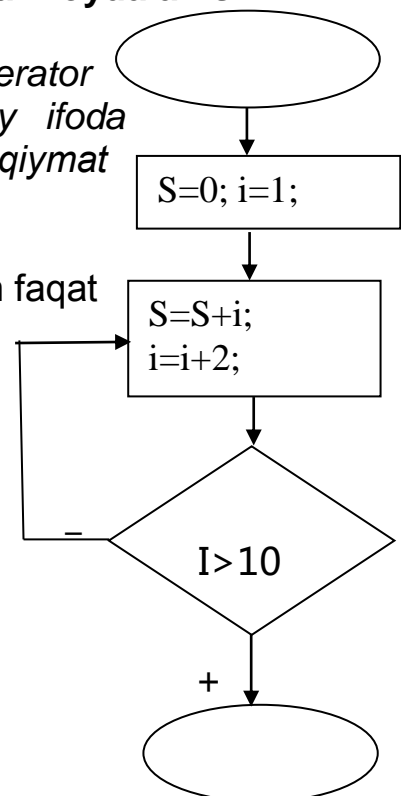
```

REPEAT S UNTIL B takrorlanish operatori va undan foydalanish.

REPEAT S UNTIL B; Bu yerda avval **S** operator tanasi bajariladi, uning tugashini **B** mantiqiy ifoda aniqlab keladi. Agar **B** mantiqiy ifoda yo'lg'on qiymat qabul qilsa, takrorlanish jarayoni to'xtaydi.

1-misol. 1 dan N gacha bo'lgan sonlar ichidan faqat toqlarini yig'indisini hisoblash dasturini tuzing.

Yechimi: Ushbu misolni matematik ko'rinishini (xususiy holda N=10 uchun) yozib olamiz. $S=1+3+5+7+9$;



```

program m_1;
uses crt;
var
    n,s,i:word;
begin

```

```

clrscr;
write('N sonini kiriting:');
readln(n);
s:=0; i:=1;
writeln('-----');
writeln('1 dan ',n,' gacha bo"lgan toq sonlar');
repeat
    s:=s+i;
    write(i, ' ');
    i:=i+2;
until i>n;
writeln('Yig'indi:',s);
readln;end.

```

2-misol.

$$U = \sin 0 + \sin 0,1 + \sin 0,2 + \dots + \sin 1$$

```

program Rsikl;
var x,y: real;
begin
    x:=0; y:=0;
    Repeat
        y=y+x; x=x+0,1;
    until x>1
    writeln ('y= ',y)
end.

```

3-misol.

Quyidagi tenglikni hisoblovchi dastur tuzing: $A = 1/2 + 2/3 + 3/4 + \dots + 9/10$

```

program misol_3;
var
    A: real; B:integer;
begin
    A:=0; B:=0;
    Repeat
        A:=A+B/(B+1);
        B=B+1;
    until B>9
    writeln ('B= ',B)
end.

```

Parametri takrorlanuvchi operatorlar.

Takrorlanishlar soni avvaldan ma'lum bo'lganda parametrli takrorlanish operatorlaridan foydalaniladi. Umumiy ko'rinishi:

```
FOR <takrorlanish parametri>:=<boshlang'ich qiymat> to <oxirgi qiymat> DO
```

Bu yerda:

FOR, DO – Pascal tilidagi maxsus so'zlar;

<takrorlanish parametri> - integer turidagi o'zgaruvchi;

<boshlang'ich qiymat> - takrorlanish parametrining boshlang'ich qiymati;

<oxirgi qiymat> - takrorlanish parametrining oxirgi qiymati;

```
For i:=1 to 10 do  
write(i, ' ');
```

dastur qismi *1 2 3 4 5 6 7 8 9 10* natijasini beradi. i takrorlanish operatori 1 dan 10 gacha 1 qadam bilan o'sib boradi.

```
For i:=10 downto 1 do  
write(i, ' ');
```

dastur qismi *10 9 8 7 6 5 4 3 2 1* natijasini beradi. i takrorlanish operatori 10 dan 1 gacha 1 qadam bilan kamayib boradi.

Agar takrorlanish jarayonida amaldan soni bittadan ortiq bo'lsa, begin ... end; operatorlari ichiga joylashtiriladi.

```
k:=0;  
For i:=1 to 5 do  
begin  
    k:=k+i;  
    write(k, ' ');  
end;
```

Ushbu dastur qismi: *1 3 6 10 15* natijasini beradi.

1-misol.

```
var i: byte;  
y: word;  
begin  
    y:=0;  
    for i:=1 to 10 do  
        begin  
            y:=y+5;  
            writeln (y, ' ');  
        end;  
end.  
(Olingan natija)
```

5 □ 10 □ 15 □ 20 □ 25 □ 30 □ ... 50 □

2-misol

$$P = \prod_{k=2}^{10} (k + 2) \text{ yoki } P=3*4*5*6*7*8*9*10*11*12 \text{ ifoda qiymatini}$$

hisoblovchi dastur tuzing.

```
var
k: byte;
p: longint;
begin
  p:=1;
  for k:=2 to 10 do
  begin
    p:=p*(k+1);
    write (p, ' ');
  end;
end.
```

3-misol. Ifodaning qiymatini topish dasturi.

$$1*1+2*2+\dots+n*n$$

```
program M_3;
var n,s,i : integer;
begin
  write('n = ');
  readln(n);
  s:=0;
  for i:=1 to n do
  s:=s+i*i;
  writeln('s = ',s);
  readln;
end.
```

4-misol. $1+1/2+1/3+\dots+1/n$ ifodani qiymatini hisoblash dasturini tuzing.

```
Program M_4;
var i, n: Byte;
S: real;
begin
  readln(n);
  S:=0; i:=1;
  repeat
    S:= S+1/i;
    i:=i+1;
```

```
until i>n;  
  writeln (S);  
end.
```

3.5. Massivlar (jadval kattaliklar).

Massivlar haqida umumiy ma'lumotlar.

Massiv - bu bir xil tipli, chekli qiymatlarning tartiblangan to'plamidir. Massivlarga misol sifatida matematika kursidan ma'lum bo'lgan vektorlar, matritsalar va tenzorlarni ko'rsatish mumkin.

Dasturda ishlatiluvchi barcha massivlarga o'ziga xos ism berish kerak. Massivning har bir hadiga murojaat esa, uning nomi va o'rta qavs ichiga olib yozilgan tartib hadi orqali amalga oshiriladi:

```
<massiv nomi> [<indeks>]
```

bu yerda <indeks> - massiv hadining joylashgan o'rnini aniqlovchi tartib nomeri.

Massivlarni e'lon qilishda indeks tipi bilan bir satrda massiv hadlarining tipi ham ko'rsatilishi kerak. Bir o'lchamli massivni e'lon qilish quyidagicha amalga oshiriladi:

```
array [<indeks tipi>] of <massiv hadining tipi>;
```

Ko'pincha <indeks tipi> sifatida cheklanma tiplardan foydalaniladi, chunki bu tipga tegishli to'plam tartiblangan va qat'iy nomerlangandir. Misol uchun, 100 ta haqiqiy sonli hadlardan iborat massiv quyidagicha e'lon qilinadi:

```
array [1..100] of real;
```

Massivlarni e'lon qilish haqida to'liqroq ma'lumot berish uchun turli tipdagi indekslarga oid misollarni e'tiboringizga havola qilamiz:

```
a:array [1..50] of integer;
```

Bu yerda 50 ta butun sondan (integer:-32768..32767) a bo'sh massivi e'lon qilindi. Massivning elementiga murojaat [] qavs ichida element turgan o'rnini bo'yicha amalga oshiriladi. Misol uchun bizga 1 dan 10 gacha toq sonlar massivi berilgan. Buni quyidagicha e'lon qilish mumkin:

```
Const
```

```
  a:array[1..4] of byte=(1,3,5,7,9);
```

Ana shu massivning 3-elementiga murojaat z:=b[3] kabi yoziladi. Bu yerda z, a massivning 3-o'rnida (indeksida) joylashgan 5 sonini qabul qiladi.

Bir o'lchovli massivlar va ular ustida amallar.

Endi massivlar ustida tipik amallar bajaruvchi bir nechta dastur bilan tanishib chiqaylik.

1. Bir o'lchamli, n ta hadli (n=30) massiv hadlarini yig'ish.

```
program L1;
  const n=30;
  var
    i: integer;
    x: array [1..n] of real;
    S: real;
  begin
    for i: =1 to n do readln (x[i]); {massiv elementlarini kiritish}
    S: =0;
    for i: =1 to n do S: =S+x[i]; {massiv elementlari yig'indisini
hisoblash}
    writeln ('Natija=', S);
  end.
```

2. Bir o'lchamli, n ta hadli (n=30) massiv hadlarining eng kattasini topish va uning joylashgan joyini aniqlash.

```
Program L2;
const n=30;
var
  x: integer;
  S: real;
  i, k: integer;
begin
  writeln (' x – Massiv hadlarini kiriting:');
  for i: =1 to n do readln (x[i]);
  S: =x[1]; k: =1;
  for i: =2 to n do
    if x[i] > S then
      begin
        S: =x[i]; k: =i
      end;
  writeln ('x massivning eng katta hadi');
  writeln (S);
  writeln ('max(x) ning o''rni', k);
end.
```

4. Massivni N ta butun tasodifiy sonlar bilan to'ldirib, massiv yig'indisi va o'rta arifmetigini hisoblash dasturi.

```
program m_3;
const
  n=10;
  max=20;
var
  a:array[1..n] of integer;
  i,s:integer;
  sr:real;
begin
  randomize;s:=0;
  for i:=1 to n do
    begin
      a[i]:=random(max)+1; {1 dan 20 gacha bo'lgan tasodifiy sonni
massivga kiritish}
      s=s+a[i];
    end;
  sr:=s/n;
  writeln('Massiv yig'indisi:',s);
  writeln('O'rta arifmetigi:',sr:3:2);
  readln; end.
```

Ko'p o'lchovli massivlar va ular ustida amallar.

Bir o'lchamli massivlarning hadlari skalyar miqdorlar bo'lgan edi. Umumiy holda esa, massiv hadlari o'z navbatida yana massivlar bo'lishi mumkin, agar bu massivlar skalyar miqdorlar bo'lsa, natijada ikki o'lchamli massivlarni hosil qilamiz. Ikki o'lchamli massivlarga misol sifatida matematika kursidagi matritsalarini keltirish mumkin. Agar bir o'lchamli massivning hadlari o'z navbatida matritsalar bo'lsa natijada uch o'lchovli massivlar hosil qilinadi va h.k.

Ikki o'lchamli massiv tipini ko'rsatish quyidagicha bajariladi:

```
array [<indeks tipi>] of array [<indeks tipi>] of <skalyar tip>;
```

Ikki o'lchamli massivlarning tiplarini bir necha xil yo'lda aniqlashni quyidagi misol ustida ko'rib chiqaylik (A matritsa 10 ta satr va 20 ta ustundan iborat bo'lib, uning xadlari haqiqiy tipga tegishli bo'lsin):

```
var A: array [1..10, 1..20] of real;
var A: array [1..10] of array [1..20] of real;
```

Ikki o'lchamli massivlar ustidagi bir nechta tugallangan dasturlar bilan tanishib chiqaylik.

1. Matritsalarini qo'shish.

```
Program L2;
const n = 3; m = 4; {n-massiv satrlari soni,m-ustunlar soni}
var i, j: integer;
    A, B, C: array [1..n,1..m] of real;
begin
for i := 1 to n do {A,B massiv hadlarini kiriting}
for j := 1 to m do
readln (A[i,j], B[i,j]);
    for i := 1 to n do
        for j := 1 to m do
            begin
                C[i,j] := A[i,j] + B[i,j];
                writeln (C[i,j]); end; readln; end.
```

2. Matritsani vektorga ko'paytirish.

```
Program L3;
const n=3; m = 4;
var
    i, j: byte;
    A: array [1..n, 1..m] of real;
    B, C: array [1..m] of real;
begin
writeln ('A matrisani kiriting');
for i:=1 to n do
    for j:=1 to m do
        readln (A[i,j]);
writeln ('B vektor hadlarini kiriting');
for i:=1 to n do readln (B[i]);
for i:=1 to n do
    begin
        C[i]:=0;
        for j:=1 to m do
            C[i]:= C[i] + A[i,j] * B[j];
        writeln (C[i]);
    end; end.
```

3. Matritsa hadlarining eng kattasini topish va uning joylashgan joyini aniqlash.

```
program L4;
const n=3; m=4;
var A: array [1..n, 1..m] of real;
```

```

R: real;
i, j: byte; K, L: byte;
begin
for i: =1 to n do {A massiv elementlarini kiritish}
for j: =1 to m do
  readln (A[i,j]);
  R:= A[1,1]; L:= 1; K:= 1;
  for i: =1 to n do
  for j: =1 to m do
    begin
      if R< A[i,j] then
        begin
          R: =A[i,j];
          L: = i; K: =j;
        end;
      end;
    writeln ('max A=', R);writeln ('сатр=',L,'устун =' ,K); end.

```

3.6. Satriy kattaliklar. Satriy kattaliklar va ular bilan bajariladigan amallar.

Satrlar ASCII kodining belgilarining apostroflar (' ') ichiga olingan ketma-ketligidan iborat. Masalan 'mart', '123', 'Informatika', 'Kitob' va h.k. Satr uzunligi deb unda qatnashgan belgilar soniga aytiladi. Bunda, chegaralovchi tashqi apostrof belgilari hisobga olinmaydi. Masalan, ' ' satrning uzunligi 1 ga teng, " yzunligi esa 0. Masalan, 'manba'" satri manba' sifatida chop etiladi. Satr **Satriy** o'zgaruvchi qiymati ham bo'lishi mumkin va u $m+1$ bayt joyni egallaydi.(m-satr uzunligi va, 1 esa xizmatchi baytning nomeri)

Satrn klaviaturadan kiritishda apostroflar kiritilmaydi.

Satriy tip quyidagicha tavsiflanadi:

```
string[<doimiylik>];
```

Bu yerda shart bo'lmagan [<doimiylik>] satr uzunligini chegaralaydi va agarda u ko'rsatilmasa, qiymati 255 ga teng deb oldinadi.

Satriy tipning berilishiga misollar keltiramiz:

```

VAR TT: string[23];
      ww, t1:string;
      var kk,dd:string[10];
begin
kk:='kombinatorica';
dd:=kk;
kk[5]:=chr(174); writeln('chr(174)=' ,chr(174));
kk[6]:=chr(225); writeln('chr(225)=' ,chr(225));

```

```
writeln(dd,' ',kk);
readln;
end.
Olinadigan natija :
Shr(174)=o
Chr(225)=c
```

Kombinator kombocator

Demak, kk:='kombinatorica' buyruqdan var k,dd:string[10]; buyruqda aniqlangani uchun faqat 10 ta belgi qoldirdi va ular bilan ish ko'rildi.

Kokatenasiya (qo'shish) - amali. Bu amal Concat funksiyasi yoki «+» belgisi orqali ifodalanadi.

Masalan,

```
write (concat(`Mega`,`Bayt`))
(olinadigan natija: Megabayt).
```

Copy (<satr>, <tartib raqam>, <miqdor>) funksiyasi satr kattaligidan <tartib raqam> dan boshlab <miqdorda> «qirqib», (nusxa olib) beradi.

Masalan:

```
write (Copy(`Aeroport`,5,4)).
natija: (port).
```

Length(<satr>) funksiyasi berilgan <satr> uzunligini beradi.
Write(length(`Pascal`));
natija: 6

Pos(<satr qismi>, <satr>) funksiyasi <satr> kattaligida, <satr qismi> mavjud bo'lsa, u joylashgan tartib raqamini beradi va agar bu <satr> da shu <satr qismi> mavjud bo'lmasa 0 natijani beradi.

Delete (<satr>, <tartib raqami>, <miqdor>) prosedurasi <satr> dan, <tartib raqami> dan boshlab, <miqdor> belgilarini o'chiradi.

INSERT (<qism satr>, <satr>, <tartib raqami>) prosedurasi <satr> kattaligiga, <tartib raqami> dan boshlab, <qism satr> ni joylashtiradi.

1-misol.

Berilgan so'zni palindromligini tekshiring. (Palindrom so'z – chapdan o'ngga va o'ngdan chapga oqilganda ham bir xil ma'noni bildiradigan so'z);

```
{1-usul}
program L;
var
  s,s1:string;
  a,i:integer;
```

```

begin
  writeln('So''zni kiriting:');
  readln(s);
  a:=length(s); s:= '';
  for i:=1 to a do
    s1:=s1+copy(s,a-i+1,1);
  if s=s1 then writeln('Polindrom so''z') else writeln('Polindrom emas');
  readln;
end.

```

```

{2-usul}
program L_2;
var
  s,s1:string;
  i:integer;
begin
  writeln('So''zni kiriting:');
  readln(s);
  for i:=length(s) downto 1 do
    s1:=s1+s[i];
  if s=s1 then writeln('Polindrom so''z') else writeln('Polindrom emas');
  readln;
end.

```

2-misol. Satrdagi so'zlar sonini aniqlash dasturi.

```

uses crt;
var i,l,k,j:longint;a:string;
begin
  clrscr;
  write('Satrni kiriting: ');readln(a);l:=length(a);
  for i:=1 to l do
    if (a[i]=' ') and (a[i+1]=' ') then inc(j)
    else a[i-j]: = a[i]; l :=l - j;
  if a[1]=' ' then
    begin delete(a,1,1);l:=l-1; end;
  if a[l]<>' ' then begin inc(l); a[l]:=' '; end;
  for i:=1 to l do if a[i]=' ' then inc(k);
  write(' So'zlar soni: ',k);
  readln;
end.

```

3.7. Qism dasturlar.

Funksiyaning berilishi va funksiyaning nomi.

Funksiya bu dasturning bir qismi bo'lib, *function* so'zi bilan boshlanib quyidagi tuzilishga ega:

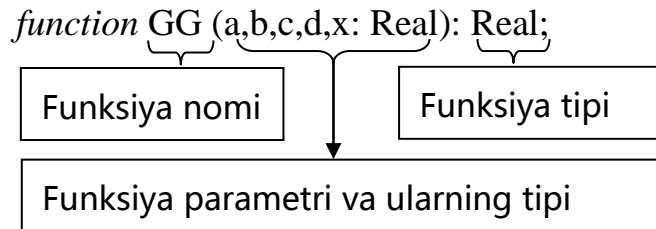
function <funksiyaning nomi> (<formal parametrlar ro'yxati>): <tip>

bu yerda tip funksiya qiymatining tipi.

Bu tip tartibli, haqiqiy, String va Point tipidan bo'lishi mumkin. Formal parametrlar quyidagi tuzilishga ega:

<Bir nechta tipga tegishli parametrlar>:
<tipning nomi>;

Misollar.



2) *function* RR1(x1,y1,x2,y2: bute): Real;

Dasturda funktsiyaga murojaat qilish haqiqiy parametrlar yordamida amalga oshiriladi, masalan

GG (6,5,6,-1,0) yoki RR1(a,b,c,d) va h.

3) *function* Aro(s: String): String;

4) *function* NNC(L: Word; c: Char): String;

Funksiya—dasturning bir qismi bo'lib, unga dasturda bir necha marotaba murojaat qilinishi mumkin. Umuman:

a) agarda parametrlar qiymatlari yordamida yagona natija olinadigan bo'lsa, funktsiyadan foydalanish mumkin;

b) qism dasturning yakuniy natijasi albatta funksiya nomiga o'zlashtirilishi shart.

Misol.

Funksiya yordamida to'rtburchak tomonlariga ko'ra uning perimetri va yuzasini hisoblovchi dastur tuzing.

Program rr;	—dastur nomi
var d,h:integer;	- Haqiqiy o'zgaruvchilarni e'lon qilish
Function Dper (a,b:integer):integer;	—funksiya nomi, formal parametrlar va funktsiyani tipi
Begin	
dper:=2*a+2*b;	—funksiyaning yakuniy natijasi uning nomiga o'zlashtiriladi
End;	
Function ss(a,b:integer):integer;	— funksiya nomi, formal parametrlar va funktsiyani tipi
Begin	
ss:=a*b;	—yakuniy natija ss funksiya nomiga o'zlashtiriladi

End;	
Begin	Dasturing asosiy qismi
Readln (d,h);	
Writeln(`to'g'ri burchak perimetri`);	
Writeln (drep (d,h));	—d,h haqiqiy parametrغا bog'liq dper va ss funksiya qiymatlari hisoblanadi
Writeln (ss(d,h));	
Readln; end.	

1-misol: Funksiya yordamida a, b, c tomonlari berilgan uchburchak yuzasini hisoblovchi dastur tuzing.

```

program ss4;
var p:real;
function geron(a,b,c:real):longint;
begin
    p:=(a+b+c)/2
    geron:=sqrt(p*(p-a)*(p-b)*(p-c));
end;
var x,y,z :read;
begin
    x:=3; y:=4; z:=5;
    writeln(geron(x,y,z));
    readln;
end.

```

2-misol. $f=m!-n!$ ifodani hisoblovchi dastur tuzing.

```

program f;
var f,m,k:longint;
function fact(n:integer):longint;
var p,i:integer;
begin p:=1; for i:=2 to n do
    p:=p*i; fact:=p;
end;
begin
    writeln(`m!-n! hisoblash uchun `);
    write(`m va n qiymatini kiriting:`);
    readln(m,k);
    f:=fact(m)-fact(k);
    writeln(`f=`;f:5);
readln;
end.

```

3.7. Qism dastur proseduralarining berilishi va proseduralarning nomi.

Funksiyaning natijasi uning nomiga o'zlashtirilishi shart, ammo qism dasturning natijalari parametrlar ro'yxatida ham berilishi mumkin. Bu holatda funksiyadan emas, balki prosedura nomli qism dasturdan foydalaniladi. Prosedura sarlavhasi quyidagi tuzilishga ega:

Procedure <prosedura nomi>
(<formal parametrlar ro'yxati >);

Prosedura nomi natijani belgilash uchun ishlatilmaydi. Formal parametrlar ro'yxatida natijalar var xizmatchi so'zidan keyin tiplari bilan birgalikda beriladi.

Prosedura funksiyaga nisbatan umumiyroq qismdastur bo'lib, funksiyani proseduraga aylantirish mumkin va proseduraning natijaviy parametrlari bir nechta bo'lishi mumkin (hatto 0 ga teng bo'lishi ham mumkin).

Proseduraning ko'rinishlariga misol keltiramiz:

- 1) Procedure Uch (a,b,c: Real; var ha, hb, hc, ma, mb, mc: real);
- 2) Procedure Aylana(R: Real; var c,d,s: real);
- 3) Procedure Max(Var A,B,C: Real);
- 4) Procedure Beep(H,T: Word).

Proseduraga murojaat qilish quyidagi ko'rinishga ega:

<Prosedura nomi> (<haqiqiy parametrlar ro'yxati>)

Misol.

- 1) Uch(3,5,7); Uch(x,y,z);
- 2) Aylana (5.1);

Funksiya va prosedurada formal va haqiqiy parametrlari orasida quyidagi mosliklar bo'lishi lozim:

- a) tiplari mos bo'lishi, agar parametr tuzilmalar shaklida bo'lsa unda ham formal va haqiqiy parametrlar tiplari bir xil bo'lishi shart;
- b) ular son jihatdan teng bo'lishi mumkin;
- d) berilish kema-ketligi mos bo'lishi.

(1 formal parametr 1-haqiqiy parametr bilan, 2-formal parametr 2-haqiqiy parametr bilan va h.).

1-masala. Kattasini topish dasturi.

```
program Largest;
```

```
var
```

```
  a,b,c,d,mab,mcd,max:integer;
```

```
procedure max2(x,y:integer; var z:integer); {protsedurani e'lon qilish}
```

```
begin
```

```
  if x>y then z:=x else z:=y;
```

```
end;
```

```

begin    {dasturning asosiy qismi}
writeln('4 ta butun sonni kiriting:');
readln(a,b,c,d);
max2(a,b,mab);
max2(c,d,mcd);
max2(mab,mcd,max);
writeln('max=',max); readln;
end.

```

2-masala. Berilgan massiv elementlarini teskari tartibda chop etuvchi dastur tuzing.

```

uses crt;
const n=5;
var a:array[1..n] of integer; i:integer;
procedure Invert; {protsedurani e'lon qilish}
begin
for i:=n downto 1 do
write(a[i], ' ');
end;

begin                                {dasturning asosiy qismi}
clrscr;
for i:=1 to n do
begin
write(i, ':');
readln(a[i]);
end;
writeln('Teskari massiv:');
Invert;                                {protseduraga murojaat qilish}
readln;
end.

```

IV bob. Pascal tilida ma'lumotlarning qayd qilingan va chegaralangan toifasi.

4.1. Ma'lumotlarning qayd qilingan toifasi. Ularni tavsiflash va ulardan dasturda foydalanish.

Biz shu vaqtgacha dasturlarda ma'lumotlarning standart toifalari bilan ishlab keldik. Bu toifalardagi qiymatlar diapazoni Paskal tili tomonidan belgilanadi. Ma'lumotlarning toifasi shu toifadagi kattaliklarning qabul qilishi mumkin bo'lgan qiymatlar to'plamini belgilaydi. Ma'lumotlarning toifasini belgilash o'z navbatida bu toifadagi kattaliklar ustida bajariladigan amallarni ham belgilaydi. Masalan, haqiqiy toifadagi kattaliklar ustida qo'shish, ayirish, ko'paytirish va bo'lish amallarini bajarish mumkin, lekin DIV, MOD amallarini bajarish mumkin emas.

Lekin Paskal algoritmik tili dastur tuzuvchiga yangi, nostandart toifalarni kiritish, ya'ni ma'lumotlarning o'ziga qulay bo'lgan qiymatlarini ishlatish imkoniyatini beradi. Bularga oddiy o'zgaruvchi toifaga kiruvchi qayd qilingan va chegaralangan toifalar kiradi.

Ma'lumotlarning qayd qilingan toifasidagi qiymatlar cheklangan sonda va qat'iy tartibda qayd etish, keltirish yo'li bilan beriladi. Bu toifadagi o'zgaruvchilar faqat shu toifada keltirilgan qiymatlarning birini qabul qilishi mumkin. Odatda qayd qilingan toifa qiymatlari yechilayotgan masalaning mazmuniga ko'ra belgilanadi.

Qayd qilingan toifalarni har doimgidek, TYPE (toifalarini tavsiflash) va VAR (o'zgaruvchilarni tavsiflash) bo'limida tavsiflash mumkin:

TYPE bo'limida:

TYPE

<qayd qilingan toifa nomi> = <konstantalar ro'yxati>;

VAR <o'zgaruvchi nomi>:<qayd qilingan toifa nomi>;

Bu yerda konstanta sifatida foydalanuvchi tomonidan beriladigan qiymatlar olinadi va ular ro'yxatda vergul orqali keltiriladi.

Masalan:

TYPE BUL = (FALSE, TRUE);

WEEK = (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday);

ANIMAL = (DOG, CAT, GOW);

VAR B1, B2: BUL; D1: WEEK; D2: ANIMAL;

Bu tavsifda BUL, WEEK, ANIMAL qayd qilingan toifalar va ularning qiymatlar ro'yxati keltiriladi. Tavsifga asosan operatorlar bo'limida quyidagi operatorlarni ishlatish mumkin:

D1:=Monday; B1:=TRUE; D2:=CAT;

quyidagi operatorlarni esa ishlatib bo'lmaydi:

B1:=DOG; D1:=FALSE; D2:=Sunday;

chunki ta'kidlaganimizdek, har bir o'zgaruvchi faqat o'zi aniqlangan toifa qiymatlarining birini qabul qilishi mumkin edi.

VAR bo'limida:

VAR <o'zgaruvchi nomi>:<(konstantalar ro'yxati)>;

bu yerda konstantalar ro'yxatida qayd qilingan toifa qiymatlari vergul orqali beriladi.

Masalan:

VAR DAY, D3: (Monday, Tuesday, WEDN, Friday);

SEZ, D4: (SUMMER, AUTUMN);

qayd qilingan toifada konstantalar tartiblangan, ya'ni har bir konstanta o'zining tartib raqamiga ega va u 0 dan boshlanadi. Yuqorida keltirilgan misollardagi WEEK toifasida konstantalar mos ravishda quyidagi tartib raqamiga ega: Sunday – 0; Monday – 1; Tuesday – 2 va hokazo. Konstantalarning tartiblanishi o'z navbatida bu konstantalar ustida ma'lum amallarni bajarish imkonini beradi va ular uchun ba'zi bir standart funktsiyalar belgilangan:

1. PRED(Z) – bu yerda Z qayd qilingan toifadagi o'zgaruvchi yoki shu qayd qilingan toifadagi konstanta. Bu funktsiya Z dan bitta oldingi konstantani aniqlab beradi.

Masalan: (yuqorida keltirilgan misolga asosan)

DAY:=Tuesday; D3:=PRED(DAY);

operatorlari bajarilishi natijasida D3 o'zgaruvchisi Monday qiymatini o'zlashtiradi.

2. SUCC(Z) – bu funktsiya Z dan bitta keyingi konstantani aniqlab beradi.

Masalan:

SEZ:=SUMMY; D4:=SUCC(SEZ);

operatorlarning bajarilishi natijasida D4 o'zgaruvchisi AUTUMN qiymatiga ega bo'ladi.

3. ORD(Z) – bu funktsiya Z ga mos kelgan qiymatning tartib raqamini aniqlab beradi.

Masalan:

SEZ:=SUMMY; D:=ORD(SEZ);

operatorlarning bajarilishi natijasida D ning qiymati 0 ga teng bo'ladi (D:INTEGER).

Bulardan tashqari, qayd qilingan toifa qiymatlari ustida solishtirish amallarini ham bajarish mumkin. Masalan, Monday <Tuesday mantiqiy ifodaning qiymati TRUE ga teng, chunki Monday – 0 tartib raqamiga, Tuesday – 1 tartib raqamiga ega.

Misol:

```
PROGRAM RR;  
TYPE DAY=(MON,TUS,WED,THUR,FRI,SAT,SUN);  
VAR WEEKDAY:DAY; TEL:INTEGER;  
BEGIN WEEKDAY:=WED;  
  WRITELN('1 TIME=',WEEKDAY);  
  WEEKDAY:=SUCC(WEEKDAY);  
  WRITELN('2 TIME=',WEEKDAY);  
  WEEKDAY:=WED;  
  WEEKDAY:=PRED(WEEKDAY);  
  WRITELN('3 TIME=',WEEKDAY);  
  TEL:=ORD(WED);  
  WRITELN('ORD-NUMBER=',TEL);  
END.
```

Dasturning bajarilishi natijasida quyidagi ko'rinishda natija olinadi:

```
1 TIME=WED  
2 TIME=THUR  
3 TIME=TUS  
ORD-NUMBER=2
```

Yuqorida keltirilgan misollardan ko'rinib turibdiki, qayd qilingan toifa qiymatlarini READ (ma'lumotlarni kiritish) operatori yordamida xotiraga kiritib va WRITE (ma'lumotlarni chiqarish) operatori yordamida bosmaga chiqarib bo'lmaydi. Qayd qilingan toifadagi o'zgaruvchilarga qiymatlar o'zlashtirish operatori yordamida o'zlashtiriladi.

4.2. Ma'lumotlarning chegaralangan toifasi

Ba'zi bir hollarda o'zgaruvchi o'zi aniqlangan toifaning hamma qiymatlarini emas, balki ma'lum chegaradagi (diapason) qiymatlarinigina qabul qilishi mumkin. Bu hollarda o'zgaruvchi uchun chegaralangan toifa belgilash mumkin. Chegaralangan toifa oldindan aniqlangan toifaga cheklanish qo'yish bilan beriladi. Oldindan aniqlangan toifa chegaralangan toifaga nisbatan baza toifasi vazifasini bajaradi. Baza toifasi sifatida REAL toifasidan tashqari barcha standart toifalarni ishlatish mumkin.

Chegaralangan toifa quyidagi ko'rinishda tavsiflanadi:

TYPE

<chegaralangan toifa nomi>=1-konstanta..2-konstanta;

VAR <o'zgaruvchi>:<chegaralangan toifa nomi>;

Bu yerda 1-konstanta va 2-konstanta mos ravishda baza toifasining birinchi va oxirgi qiymatlari. Chegaralangan turdagi o'zgaruvchi baza toifasining birinchi qiymati va oxirgi qiymatlari orasidagi ixtiyoriy qiymatlarning birini qabul qilishi mumkin.

Masalan:

TYPE SON=1..100;

SIM='A'..'F';

FASL=(KUZ, QISH, BAHOR, YOZ);

SEZ=KUZ..BAHOR;

BUL=FALSE..TRUE;

VAR S:SON; F:FASL; S:SEZ; B:BUL;

Bu misolda SON toifasi uchun INTEGER toifasi, SIM toifasi uchun CHAR toifasi, SEZ toifasi uchun FASL toifasi, BUL toifasi uchun BOOLEAN toifasi baza toifasi vazifasini bajaradi.

Chegaralangan toifani aniqlashda quyidagi qoidalarga amal qilish kerak:

1. Chegaralangan toifaning birinchi va ikkinchi konstantalari bir xil toifada bo'lishi kerak.
2. Birinchi konstanta qiymati ikkinchi konstanta qiymatidan kichik bo'lishi lozim.

Masala. Guruhdagi 5 ta talaba ichida a'lochi talabani aniqlash dasturi tuzilsin.

Program alochi;

```

Type
Fam=(ahmedov,sobirov,valiyeva, sodiqov,ravilov);
Fan=(mat,him,fiz,ing);
Baho=1..5;
Var stud,i:fam;fan; b:baho; s:real;
Begin
Writeln('talabning bahosini kiriting ');
  for i:=ahmedov to ravilov do
    begin s:=0; stud:=i;
      for j:=mat to ing do
        begin read(b); s:=s+b end;
          readln;
          if s/4>=5 then
            Begin
              writeln('alochi talaba ');
              case stud of
                Ahmedov:writeln('axmedov ');
                Sobirov:writeln('sobirov ');
                Valiyeva:writeln('valiyeva ');
                Sodiqov:writeln('sodiqov ');
                Ravilov:writeln('ravilov ')
              End
            end;
          end;
        end
      end.

```

Bu dasturda FAM – talabalarning familiyalari keltirilgan toifa, FAN – fanlar nomi keltirilgan toifa va BAHO – baholar mezoni ko'rsatilgan toifalar yaratilgan. Dasturni bajarish jarayonida har bir talabning har bir fandan olgan bahosi 5 ballik tizimda alohida satrda kiritiladi. Natijada ekranga

A'LOCHI TALABA

yozuvidan keyin talabning familiyasi chiqariladi.

4.3. Ma'lumotlarning murakkab toifasi. Muntazam toifa

Yuqorida ta'kidlab o'tilganidek, Paskal tilida ishlatiladigan ma'lumotlar 2 turga, oddiy (REAL, INTEGER, BOOLEAN, CHAR, qayd qilingan va chegaralangan) va murakkab turdagi ma'lumotlarga, (muntazam turdagi ma'lumotlar, yozuvlar, faylli turlar, to'plamlar va ko'rsatkichlar) bo'linadi. Oddiy turdagi ma'lumotlarning boshqa turdagi ma'lumotlardan asosiy farqlanuvchi belgisi, ularning tartiblanganligi va yaxlitligidadir, ya'ni masalan INTEGER toifasiga mansub ixtiyoriy kattalik alohida raqamlarga bo'linmaydigan yaxlit kattalikdir (INTEGER toifasidagi kattalik sifatida bitta butun son tushuniladi). Lekin sonli o'qning ixtiyoriy bo'lagini butun sonlar ketma-ketligi deb qarajak, u holda shu ketma-ketlik i-

nchi raqami to'g'risida gapirish, bu elementlar (butun sonlar) to'plamiga «butun son» degan umumlashtiruvchi nom berish mumkin.

Paskal tilida bunday kattaliklarni ma'lumotlarning muntazam toifasi ko'rinishida ifodalash mumkin. Umumiy nomga ega, tartiblangan kattaliklar ketma-ketligi muntazam toifadagi ma'lumotlar yoki massivlar deb yuritiladi. Ularning tashkil etuvchilari massiv elementlari va elementlari toifasi bazaviy toifa deb nomlanadi. Masalan, quyidagi tartiblangan haqiqiy sonlar ketma-ketligi

1.5, 14.7, -5.0, 8.5, 0.45

Haqiqiy sonlar massivi, bu massivning bazaviy toifasi esa haqiqiy (REAL) toifadir.

Massiv tushunchasini A – umumiy nomga ega va bir toifadagi qo'zg'almas kattaliklar ketma-ketligidan iborat bo'lgan sonli vector

$A(5) = (a_1, a_2, a_3, a_4, a_5)$

misolida tushuntirish mumkin. Bu yerda a_1, a_2, \dots - massiv elementlaridir. Ularni ifodalashda ko'rsatkichli (indeksli) o'zgaruvchilardan foydalaniladi.

Matematika kursidan ma'lumki, ko'rsatkich (indeks) o'zgaruvchilarning tartiblangan ketma-ketlikdagi o'rnini bildirib, qavslar ichiga olinib yoki massiv nomidan bir muncha pastda ko'rsatilar edi, masalan A(1) yoki A₁, umumiy holda esa A_l, bu yerda $l=1,2,3,\dots,n$

Paskal tilida o'zgaruvchining ko'rsatkichi to'rtburchak qavslar ichiga olinib yoziladi, ya'ni A[1]=1.5, A[2]=14.7, A[3]=-5.0, A[4]=8.5, A[5]=0.45.

Agar dasturda massiv ishlatilayotgan bo'lsa, u holda uni o'zgaruvchilar bo'limi VAR yoki toifalar bo'limi TYPEda tavsiflash zarur, masalan, VAR bo'limida massiv quyidagi ko'rinishda tavsiflanadi:

VAR <massiv nomi>:

ARRAY [<ko'rsatkich toifasi>] OF <element toifasi>;

Bu yerda <massiv nomi> - ixtiyoriy identifikator, ARRAY (massiv) va OF (-dan) – xizmatchi so'zlar, <ko'rsatkich toifasi> - ko'rsatkichli ifoda, bu toifa qiymatini massiv elementlarining sonini belgilaydi va ko'rsatkichlarni yozish uchun ishlatiladigan belgilarni ko'rsatadi, shuning uchun bu toifa sifatida REAL va cheklanmagan INTEGER toifasidan tashqari barcha oddiy toifalarni ishlatish mumkin, <element toifasi> - massiv elementlarining toifasi bo'lib, bu toifa sifatida fayl va to'plam toifasidan boshqa barcha toifalarni ishlatish mumkin.

Yuqorida ko'rib o'tilgan A vektorni massiv ko'rinishda dasturda quyidagicha tavsiflash mumkin:

VAR A : ARRAY [1..5] OF REAL;

Ineks sifatida faqat aniq son emas, qiymati massiv elementining tartib raqamini belgilovchi ifoda ham ishlatilishi mumkin, masalan, A[1+5], B[l div (j+6)], C[n1 or n2], YEAR[1988] va h.k. Ko'rsatkichli ifoda qiymatining toifasi massiv elementi ko'rsatkichining toifasi deyiladi va bu toifa qiymatlarining to'plami massiv elementlarining sonini va ularning tartiblanganligini aniqlovchi to'plam bo'lishi kerak. Quyida ko'rsatkich toifasi sifatida Paskalda ishlatilishi mumkin bo'lgan toifalar bilan tanishamiz.

Ma'lumki, Paskal tilida INTEGER va REAL toifasidagi ma'lumotlar to'plami cheklanmagan, REAL toifa esa shunindek tartiblanmagandir ham. Shu sababli ko'rsatkich toifa sifatida INTEGER va REAL toifasiga mansub ma'lumotlardan foydalanish mumkin emas, ya'ni quyidagicha yozish mumkin emas:

```
VAR A:ARRAY[5] OF REAL;  
S: ARRAY [INTEGER] OF REAL;  
B: ARRAY [1.20...1.45] OF INTEGER;
```

Ko'p hollarda ko'rsatkich toifasi sifatida chegaralangan toifa, ko'pincha – chegaralangan butun toifa ma'lumotlari ishlatiladi. Masalan, 100 ta haqiqiy elementlardan iborat massiv quyidagi ko'rinishda tavsiflanishi mumkin:

```
VAR A: ARRAY [1..100] OF REAL;
```

Bu yerda 1 dan 100 gacha chegaralangan butun toifa A massiv elementlari soni yuzta ekanligi va ularning tartiblanganligini (birinchidan yuzinchigacha) bildiradi.

To'g'ri tavsiflangan massivlarga misollar:

```
VAR MASSIV:ARRAY [-745..-1] OF REAL;  
YEAR:ARRAY [1477..1988] OF CHAR;  
L:ARRAY [BOOLEAN] OF CHAR;  
A,B,C:ARRAY [1..50] OF REAL;
```

Ma'lumki, qayd qilingan toifa qiymatlarining to'plami ham cheklangan va tartiblangan to'plamni tashkil qiladi, bu esa o'z navbatida qayd qilingan toifa qiymatlarini ham ko'rsatkich toifasi sifatida qo'llash imkoniyatini beradi, masalan:

```
VAR MONTH: ARRAY [mart, apr, may] OF CHAR;  
COLOR: ARRAY [red, blue, yellow, black] OF INTEGER;
```

Ko'rsatkichi va ko'rsatkich toifasi tushunchalari o'rtasida o'zaro farq mavjud bo'lib, ko'rsatkich toifasi massiv elementlari soni va ularning tartiblanganligini bildiradi va u massivni tavsiflash bo'limida ishlatiladi, ko'rsatkich esa massiv elementining tartib raqamini belgilaydi va operatorlar bo'limidagina ishlatiladi. Agar biror bir massivga murojaat qilish uchun uning to'liq nomi, ya'ni muntazam toifaga mansub o'zgaruvchining nomi ishlatilsa, massivning alohida elementiga murojaat qilish uchun ko'rsatkichli o'zgaruvchi ishlatiladi. Masalan, A(5) massiv uchun A o'zgaruvchi to'liq o'zgaruvchi (massiv nomi), A[I] – ko'rsatkichli o'zgaruvchi bo'lib, u A massivning I – elementini ifodalaydi.

Ifodalarda ko'rsatkichli o'zgaruvchilar qiymat berish operatorning chap tarafida ham, o'ng tarafida ham ishtirok etishi mumkin va ular ustida solishtirish amallarini, tartiblash, arifmetik amallar, eng kichik va eng katta qiymatni topish amallarini, ya'ni uning bazaviy toifa ustida bajarilishi mumkin bo'lgan barcha amallarni bajarish mumkin. Masalan, agar bazaviy toifa INTEGER bo'lsa, u holda butun toifa ustida bajarilishi mumkin bo'lgan barcha amallar, hatto standart funktsiyalarni ham qo'llash mumkin.

Massiv elementlarining operatorlar bo'limida ishlatilishiga misollar:

```
B[5]:= B[3] + 1;  
SUM:= SUM – ROUND(C[K]);  
P1:=SQRT(A[2*I+1]);
```

Paskal tilida massivlarni o'zgaruvchilar bo'limida tasvirlashdan tashqari TYPE toifalar bo'limida ham tasvirlash mumkin. Buning uchun TYPE bo'limida massiv toifasi nomi va massiv toifasi beriladi, VAR bo'limida ushbu toifaga mansub o'zgaruvchilar sanab o'tiladi.

TYPE bo'limida toifalarni tavsiflash dasturlashda yaxshi uslub sanaladi va dasturni mantiqiy mukammalligini oshirish imkoniyatini beradi. Massivni TYPE bo'limida tavsiflash quyidagi ko'rinishga ega bo'ladi:

```
TYPE <toifa nomi> =  
ARRAY [<ko'rsatkich toifasi>] OF <element toifasi>;  
VAR <o'zgaruvchi yoki massiv nomi>:<toifa nomi>;  
Masalan:  
TYPE  
MAS = ARRAY [1..5] OF REAL;  
VAR A : MAS;
```

Agar dasturda bir emas, bir nechta massiv MAS toifasiga mansub bo'lsa, u holda har bir massivni alohida tavsiflamasdan, o'zgaruvchilar bo'limida ularni vergul orqali keltirish mumkin, ya'ni

```
VAR A,B,C,D : MAS;
```

Shu vaqtgacha biz elementlari faqat bitta ko'rsatkichli massivlarni, ya'ni bir o'lchamli massivlarni ko'rib chiqayotgan edik. Paskal dasturlash tilining massiv elementlari toifasiga ularning hammasi bir toifaga mansub bo'lishi kerakligidan boshqa hech qanday cheklanishlar qo'yilmasligi massiv elementlari sifatida massivlar ham ishtirok etishi imkoniyatini beradi. Bunday massivlar ko'p o'lchamli massivlarni tashkil qiladi. Agar massiv elementlari o'z navbatida massivdan iborat bo'lsa, u holda ikki o'lchamli massiv (matritsa), agar matritsa elementlari massiv bo'lsa, u holda biz uch o'lchovli massivni olamiz va h.k.

Massivlarning o'lchami ularni EHMda qayta ishlashga hech qanday to'sqinlik ko'rsatmaydi, chunki massiv elementlari ularni o'lchamidan qat'iy nazar EHM xotirasida chiziqli ketma-ketlik sifatida saqlanadi, ya'ni $a_1, a_2, a_3, a_4, a_5, a_6$.

Ko'p o'lchamli massivlar dasturda quyidagicha tavsiflanadi:

```
ARRAY [<ko'rsatkich toifasi>] OF ARRAY [<ko'rsatkich toifasi>] OF  
<elementlar toifasi>;  
yoki  
ARRAY [<ko'rsatkich toifasi, ko'rsatkich toifasi>] OF <elementlar  
toifasi>;
```

Masalan, ikki o'lchovli massivlarga misol qilib matritsalarini keltirish mumkin. Ikki o'lchovli massivni e'lon qilishning bir necha usullari mavjud, masalan massivni elementlari ham massivdan iborat massiv sifatida e'lon qilish mumkin,

```
TYPE
    MAS = ARRAY [1..5] OF REAL;
    MATR = ARRAY [1..3] OF MAS;
VAR
    A:MAS;
    B:MATR;
```

Bu yerda A o'zgaruvchi 5 ta haqiqiy toifaga mansub elementdan iborat bir o'lchovli massiv, B o'zgaruvchi 3 ta satr va 5 ta ustundan iborat (3X5) ikki o'lchamli massiv sifatida tavsiflangan.

B massiv tavsifini MAS toifasini ko'rsatmasdan birmuncha soddalashtirish mumkin,

```
TYPE
    MATR = ARRAY [1..3] OF ARRAY [1..5] OF REAL;
VAR
    B:MATR;
```

yoki

```
TYPE
    MATR = ARRAY [1..3,1..5] OF REAL;
VAR
    B:MATR;
```

Agar ko'rsatilgan toifa dasturda bitta massivni aniqlash uchun ishlatilayotgan bo'lsa, massivni o'zgaruvchilar bo'limida e'lon qilish maqsadga muvofiq bo'ladi:

```
VAR
    B: ARRAY [1..3,1..5] OF REAL;
```

B matritsaning I – satr va J- ustuni kesishmasida turgan elementiga murojat B[I,J] yoki B[I],[J] ko'rinishga ega bo'ladi. Yuqoridagi tavsifga asosan quyidagi operatorlarni ishlatish mumkin:

```
A[I]:=2.5E03; B[I,J+1]:=A[K]*B[K,J]; READLN (B[I,J]);
```

Dasturda massivlardan foydalanish uchun massiv elementlari qiymatlari xotiraga kiritilgan bo'lishi zarur. Massiv elementlariga qiymat berishda ma'lumotlarni kiritish yoki o'zlashtirish operatorlaridan foydalanish mumkin. Quyida keltirilgan dastur lavhasida bir o'lchovli A va ikki o'lchovli B massivlar elementlarini kiritish va chiqarish amalga oshirilgan.

```
BEGIN
```

```

      (*A massiv elementlarini kiritish*)
READ (A[1], A[2], A[3]);
      (*B massiv elementlarini kiritish*)
READLN;
      FOR I:=1 TO 2 DO
      FOR J:=1 TO 2 DO
READ (B[I,J]);
      (*A massiv elementlarini chiqarish*)
WRITELN ('MASSIV A');
WRITELN;
      FOR I:=1 TO 3 DO
WRITE(A[I]:5);
WRITELN;
      (*B massiv elementlarini chiqarish*)
WRITELN('MASSIV B');
      FOR I:=1 TO 2 DO
BEGIN
WRITELN;
      FOR J:=1 TO 3 DO
WRITE (B[I,J]:4)
END
END.

```

A massiv uchta elementdan iborat, shuning uchun kiritish operatorida uning barcha tashkil etuvchilari sanab o'tilgan. Bu massiv elementlarini chiqarish uchun I ko'rsatkich bo'yicha takrorlanish tashkil qilinib, bosmaga «MASSIV A» sarlavhasini chiqarish yo'li bilan amalga oshirilgan. Ikki o'lchovli B massivi elementlarini kiritish (va chiqarish) uchun matritsani satrlab kirituvchi ichma-ich joylashgan takrorlanishlar ishlatilgan.

Kiritish jarayonida massiv elementlari qiymati bir-biridan bo'sh joy (probel) bilan ajratilgan bo'lishi mumkin.

```

1 2 3 <Enter>
4 5 6 7 8 9 <Enter>

```

Chiqariladigan ma'lumotlar mos ravishda quyidagi ko'rinishga ega bo'ladi:

```

MASSIV A
1 _ _ _ _ 2 _ _ _ _ 3
MASSIV B
_ _ _ 4 _ _ _ 5 _ _ _ 6
_ _ _ 7 _ _ _ 8 _ _ _ 9

```

quyida massiv elementlari ustida ko'puncha ishlatiladigan amallar ishtirokida misollarni ko'rib chiqamiz.

1-Misol. Haqiqiy N ta elementlardan iborat A massiv berilgan. Massiv elementlarining o'sib boorish tartibida joylashtiring.

Ushbu masalani echish uchun massiv elementlarini IF operatori yordamida ketma-ket solishtirish va o'zlashtirish operatori yordamida elementlarni o'rnini o'zgartirib, tartiblangan massiv ro'yxatini tuzish kerak bo'ladi. IF operatorida shart tekshirilib, uning bajarilishi X qo'shimcha o'zgaruvchi yordamida elementlarning o'rnini almashtirishga olib kelsa, shartning bajarilmasligi keyingi qadamga o'tish imkonini beradi. Paskal tilidagi dastur quyidagi ko'rinishga ega:

```
PROGRAM MASSIVNI SARALASH;
  VAR
    A = ARRAY [1..100] OF REAL;
    I,K,N : INTEGER; X : REAL;
BEGIN
  READLN(N);
  FOR I:=1 TO N DO
    READ (A[I]);
  FOR I:=1 TO N-1 DO
    FOR K:=N DOWNT0 I DO
      IF A[K] > A[K-1] THEN
        BEGIN
          X:=A[K];
          A[K]:=A[K-1];
          A[K-1]:=X
        END;
  FOR I:=1 TO N DO
    WRITELN (A[I]);
END.
```

Qo'shimcha kiritilgan I,K,N butun o'zgaruvchilarni alohida ta'kidlab o'taylik, I – massivning joriy o'zgaruvchisining ko'rsatkichi, K – ichki takrorlashdagi o'zgaruvchi ko'rsatkichi, N – massiv elementlarining umumiy soni.

2-Misol. A(2x3) va B(3x3) matritsa elementlarini o'zaro ko'paytmasini topish dasturini tuzing.

Natijaviy C matritsa elementlari quyidagi formula yordamida topiladi,

$$C_{ij} = \sum_{k=1}^p a_{ik} b_{kj}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m; \quad k = 1, 2, \dots, p.$$

Bu yerda n – A matritsaning satrlar soni, m – A matritsaning ustunlar soni va B matritsaning satrlar soni, p – B matritsaning ustunlar soni. Umumiy holda natijaviy matritsa C n – ta satr va p-ta ustundan iborat bo'ladi.

Boshlang'ich qiymatlarni kiritish va chiqarish ichma-ich joylashgan takrorlanish yordamida amalga oshiriladi. Natijaviy matritsaning

elementlarini hisoblash ham ichma-ich joylashgan takrorlanishning ichki k parametrli takrorlanishida hisoblab topiladi.

Ko'rib o'tilgan misolni yechish dasturi quyidagi ko'rinishga ega:

```
Program ko'paytma;
(*matritsalar ni ko'paytirish darsturi*)
var
    a,c : array [1..2,1..3] of real;
    b: array [1..3,1..3] of real;
    i,j,k: integer;
begin
    {*a massivni kiritish*}
    for i:=1 to 2 do
        for j:=1 to 3 do
            read(a[i,j]);
        {*b massivni kiritish*}
    readln;
    for i:=1 to 3 do
        for j:=1 to 3 do
            read(b[i,j]);
        {*matritsalar ni ko'paytirish*}
    for i:=1 to 2 do
        for j:=1 to 3 do
            begin
                c[i,j]:=0;
                for k:=1 to 3 do
                    c[i,j]:=c[i,j]+a[i,k]*b[k,i]
                end;
            {*c massivni bosmaga chiqarish*}
        writeln ('massiv c');
        for i:=1 to 2 do
            begin
                writeln;
                for j:=1 to 3 do
                    write(c[i,j]:12:5)
                end;
            end;
        end.
end.
```

4.4. Ma'lumotlarning aralash toifasi. Yozuvlar

Biz avval ma'lumotlarning murakkab toifasi massivlar bilan tanishgan edik. Massivlar yordamida bir toifaga tegishli bo'lgan ma'lumotlarni ifodalash mumkin edi.

Amaliyotda iqtisod va axborotni qayta ishlash masalalarini echishda ma'lum turdagi hujjatlar, kataloglar, ro'yxatlar, vedomostlar ishlatiladi. Masalan, talabalarning anketa ma'lumotlari: familiyasi, ismi, otasining ismi,

turar joyi, tug'ilgan yili, mutaxassisligi, guruh raqami va hokazo. Bu hollarda turli toifadagi ma'lumotlarni bir guruhga birlashtirish zarurati tug'iladi. Bizning visolimizda bu ma'lumotlarni talaba guruhiga birlashtirish mumkin. Ko'rinib turibdiki, bu guruhdagi ma'lumotlarning toifalari turlicha: familiya, ism – satr (so'z), tug'ilgan yili, guruh raqami – butun toifaga tegishli. Paskal tilida bunday ma'lumotlarni aralash toifa yordamida ifodalash imkoniyati berilgan. Aralash toifaning qiymati sifatida yozuvni olish mumkin.

Yozuv deb turli toifadagi ma'lumotlarning cheklangan to'plamiga aytiladi.

Dasturda yozuvlarni tavsiflash

Dasturda yozuv tarkibiga kirgan barcha ma'lumotlar maxsus so'z RECORD va END orasiga olinadi va ular yozuv elementlari deb yuritiladi. Elementlar ramziy nom bilan nomlanadi. Elementlar turli toifaga tegishli bo'lganligi uchun ularning toifasi tavsifi alohida-alohida beriladi. Element nomi va uning tavsifi keltirilgan satr yozuv maydonini tashkil etadi. Demak, yozuvlar bir nechta maydondan tarkib topgan bo'lishi mumkin.

Yozuvlarni tavsiflashda quyidagi qoidalardan foydalaniladi:

<aralash toifa> ::= RECORD <maydonlar ro'yxati> END;

<maydonlar ro'yxati> ::= <yozuvar bo'limi>;

<yozuvar bo'limi> ::= <maydon nomi> {, <maydon nomi> } : <toifa>;

Har bir ob'ektga nom berilgani kabi yozuvlarga va uning elementlariga ham ramziy nom beriladi.

Yozuvlarni ikki usulda tavsiflash mumkin:

1. Toifalarni tavsiflash bo'limida.

Tavsiflashning umumiy ko'rinishi quyidagicha:

TYPE <toifa nomi> = RECORD

1-element nomi : 1-element toifasi;

2-element nomi : 2-element toifasi;

.....

n-element nomi : n-element toifasi

END;

Var <yozuvar nomi, ya'ni aralash toifadagi o'zgaruvchi>: toifa nomi;

Masalan,

type zapis=record

fam: string[15];

ty: integer;

ky: integer;

gtr: integer

end;

var stud:zapis;

Bu yerda STUD nomli yozuv 4 ta elementdan iborat: FAM satr toifadagi, TY, KY, GTR butun toifadagi qiymatlarni qabul qiladi.

2. o'zgaruvchilarni tavsiflash bo'limida.

Tavsiflashning umumiy ko'rinishi quyidagicha:

VAR yozuv nomi: RECORD

1-element nomi : 1-element toifasi;

2-element nomi : 2-element toifasi;

.....

n-element nomi : n-element toifasi

END;

Masalan:

```
var stud : record
```

```
    fam: string[15];
```

```
    gtr: integer;
```

```
    baho : array [1..3] of integer
```

```
end;
```

Yozuv elementi va ular ustida bajariladigan amallar

Bir yozuvga tegishli bo'lgan elementlar turlicha nomlanishi shart. Lekin turli yozuvlarda bir xil nomli elementlar uchrashi mumkin. Chunki, har bir elementga u tegishli bo'lgan yozuv nomi orqali murojat qilinadi.

Dasturda yozuv elementlari quyidagicha ifodalanadi:

<yozuv nomi> <element nomi>

Masalan: STUD.GTR, STUD.BAHO[2], STUD.FAM

Bu yerda STUD – yozuv toifasidagi o'zgaruvchi nomi, GTR, BAHO[2], FAM –element nomlari.

Yuqorida ta'kidlab o'tilganidek, element toifasi turlicha bo'lishi mumkin. Element toifasi to'g'ridan-to'g'ri yozuv ichida yoki toifalarni tavsiflash bo'limida aniqlangan bo'lishi mumkin. O'z navbatida element toifasi yozuvdan iborat bo'lishi mumkin. Bu holatda yozuvlar murakkab tuzilishni tashkil qiladi.

Masalan, quyidagicha tavsif berilgan bo'lsin:

```
type pr = record
```

```
    mat, fiz, him, inf : integer
```

```
end;
```

```
    tyil = record kun:1..31;
```

```
                oy:1..12;
```

```
                yil:integer
```

```
end;
```

```
var uch: record
```

```
    fam : string[15];
```

```
    gd : tyil;
```

```
    baho : pr
```

```
end;
```

Bu tavsifga ko'ra UCH nomli yozuvning GD va BAHO elementlari yozuvdan iborat. Tavsif asosida quyidagi o'zlashtirish operatorlarini yozish mumkin:

```
uch.fam:=valiev'; uch.gd.kun:=23;
```

```
uch.gd.yil:=1967;
```

```
uch.baho.mat:=4;
```

```
uch.gd.oy:=10;
```

Yozuv elementlari ustida u aniqlangan toifadagi ma'lumotlar ustida bajarilishi mumkin bo'lgan amallarni bajarish mumkin.

Masala. Talabalar haqida quyidagi ma'lumotlar berilgan: Talabaniing familiyasi, tug'ilgan yili, dorilfununga kirgan yili, guruh raqami. Shu ma'lumotlar asosida 34-guruhda o'qiydigan talabalar ro'yxati bosmaga chiqarilsin.

Masalaning yechish dasturi quyidagicha:

```
Program guruhni aniqlash;
```

```
Type zapis=record
```

```
    fam:string[15];
```

```
    tugyil: integer;
```

```
    kiriyil: integer;
```

```
    grtar: integer
```

```
end;
```

```
Var stud: zapis; gr:array[1..10] of zapis;
```

```
    i,j,n:integer;
```

```
Begin writeln('talabalar sonini kiriting');
```

```
    readln(n);
```

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

```
        begin
```

```
            readln(stud.fam);
```

```
        Readln(stud.tugyil,stud.kiriyil,stud.grtar);
```

```
            gr[i]:=stud;
```

```
        end;
```

```
        Writeln('34-gurux talabalari');
```

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

```
                if gr[i].grtar=34 then
```

```
                    writeln(gr[i].fam,',',gr[i].grtar)
```

```
End.
```

4.5. Birlashtirish operatori

Yuqorida keltirilgan misollardan ko'rinib turibdiki, agar yozuv tarkibi murakkab tuzilishga ega bo'lsa, ya'ni yozuv ichida yozuv aniqlangan bo'lsa, yozuv elementini ifodalashda u o'zi aniqlangan yozuv nomini qayta-qayta yozish kerak bo'layapti. Buning evaziga dastur hajmi ham oshib ketadi. Shu noqulayliklarni bartaraf qilish maqsadida, Paskal tilida maxsus birlashtirish operatori kiritilgan.

Birlashtirish operatori yordamida yozuv nomi "sarlavhaga" chiqariladi va elementga murojat qilishda faqat elementning nomi ko'rsatiladi xolos.

Operatorning sintaksis tavsifi quyidagicha:

```
<Birlashtirish operatori>::=<sarlavha><operator>;
```

```
<saralash>::=WITH <o'zgaruvchilar ro'yxati> DO
```

<o'zgaruvchilar ro'yxati>::=<yozuv toifasidagi o'zgaruvchi>
{<yozuv toifasidagi o'zgaruvchi>}

Yoki operatorning umumiy ko'rinishi quyidagicha:

with f do o

Bu yerda WITH va DO – xizmatchi so'zlar, F – yozuv toifasidagi o'zgaruvchi yoki o'zgaruvchilar ro'yxati, O – operator (oddiy, murakkab yoki tarkiblashgan operator).

Masalan, yuqorida misol qilib olingan tavsifga ko'ra quyidagi dastur lavhasini yozish mumkin:

```
with uch,gd,baho do
begin readln(fam);
      readln(mat,fiz,him,inf)
end;
```

yoki

```
with uch do
with gd do
with baho do
begin
      readln(fam,mat,fiz,him,inf)
end;
```

Masala: Talabalar haqida quyidagi ma'lumotlar berilgan: talabaning familiyasi va matematika, informatika, fizika fanlaridan olgan baholari. Shu ma'lumotlar asosida qarzdorlar ro'yxati tayyorlansin. Qarzdor talaba deb, hech bo'lmaganda bitta fandan 2 baho olgan talaba hisoblansin.

Masalaning yechish dasturi:

```
program qarzdor;
type st=record
      fam:string[15];
      baho:record
            mat,inf,fiz : integer
            end
      end;
var grup:array[1..5] of st; i,n:integer; s:st;
begin writeln('talabalar sonini kiriting');
      readln (n);
      with s, baho do
      begin for i:=1 to n do
            begin readln(fam,mat,inf,fiz);
                  grup[i]:=s
            end
      end;
writeln('qarzdorlar royxati');
      for i:=1 to n do
            begin with grup[i],baho do
```

```

begin if (mat=2) or (inf=2) or (fiz=2) then writeln(fam)
      end
end
end.

```

Bu dasturda 5 ta yozuvdan iborat GRUP massivi tashkil qilingan. S aralash toifadagi o'zgaruvchi, ya'ni yozuv nomi, FAM, BAHO, MAT, INF, FIZ yozuv elementlari.

4.6. To'plamlar. Paskal tilida to'plam tushunchasi. To'plam ustida bajariladigan amallar.

Matematika kursida biz to'plamlar tushunchasi bilan tanishmiz. To'plam deganda, bir nechta elementlarning majmuasi tushuniladi. Bu elementar bir xil toifali, lekin tartiblanmagan bo'ladi. Masalan, Butun sonlar to'plami, radiodetallar to'plami va hokazo.

Paskal tilida to'plam deb, bir toifadagi cheklangan sandagi ma'lumotlarning betartib majmuasiga aytiladi. Har bir to'plamga nom beriladi. To'plamga kirgan ma'lumotlar to'plam elementlari deb yuritiladi. Elementlar toifasi baza toifasi deyiladi. Baza toifasi sifatida REAL va cheklanmagan INTEGER toifasidan tashqari barcha standart, qayd qilingan va chegaralangan toifalarni olish mumkin.

To'plamlarni tavsiflash

To'plamlar quyidagicha ifodalanadi:

<to'plamning berilishi> ::= SET OF <baza toifasi>

<baza toifasi> ::= <turning berilishi> | <tur nomi>;

bu yerda SET – to'plam ma'nosini, OF – dan ma'nosini anglatadi.

To'plamlar 2-usulda tavsiflanishi mumkin:

1. TYPE bo'limida.

TYPE <toifa nomi> = SET OF <baza toifasi>;

2. VAR bo'limida

VAR <identifikator> : SET OF <baza toifasi>;

Masalan:

TYPE TOPL = SET OF 1..9;

FASL = (YOZ, KUZ, QISH);

SEZON = SET OF FASL;

VAR SON : TOPL; NN : SEZON;

To'plam toifasida o'zgaruvchilarga qiymatlar operatorlar bo'limida konstruktor yordamida beriladi. Ularni kiritish operatori READ yordamida kiritib bo'lmaydi. Konstruktor bu “[]” qavsga olingan baza toifasidagi elementar ro'yxatidan iborat, ya'ni

<konstruktor> ::= [] yoki [<elementlar ro'yxati> {, elementlar ro'yxati}]

Masalan:

SON:=[1,3,6]; NN:=[YOZ,KUZ]; NN:=[KUZ,QISH];

SON:=[] (bo'sh to'plam);

Agar to'plamda N ta element bo'lsa, 2^n ta to'plam osti yaratish mumkin.

To'plam elementlari ustida bajariladigan amallar

To'plamlar ustida quyidagi amallarni bajarish mumkin:

1. To'plamlarni qo'shish:

$$C = A + B$$

bu yerda C,A,B, - to'plam nomlari.

To'plamlarning yig'indisi deb, ya'ni C to'plam deb yoki A to'plamga yoki B to'plamga tegishli bo'lgan elementlar to'plamiga aytiladi.

2. To'plamlarni ayirish.

$$C = A - B$$

To'plamlarning ayirmasi deb, ya'ni C to'plam deb, faqat A to'plamga tegishli bo'lgan elementlar majmuasiga aytiladi.

3. To'plamlarni kesishtirish.

$$C = A * B$$

To'plamlarning kesishmasi deb, bir vaqtning o'zida ham A ga ham B ga tegishli bo'lgan elementlar majmuasiga aytiladi.

Masalan,

[aylana, romb]+[aylana, kvadrat]=[aylana, romb, kvadrat]

[aylana, romb] * [aylana, kvadrat] = [aylana]

[aylana, romb] - [aylana, kvadrat] = [romb]

To'plam elementlari ustida =, <>, <=, >= solishtirish amallarini ham bajarish mumkin:

1. Agar A va B to'plam bir xil elementlardan tashkil topgan bo'lsa, $A=B$ mantiqiy ifodaning qiymati TRUE ga teng.

2. $A<>B$ mantiqiy ifodaning qiymati TRUE ga teng, agar solishtirilayotgan to'plamlarning birida ikkinchi to'plamga kirmagan hech bo'lmaganda bitta element bo'lsa.

3. $A<=B$ mantiqiy ifodaning qiymati TRUE ga teng, agar A to'plamning hamma elementlari B to'plamga kirsas.

4. $A>=B$ mantiqiy ifodaning qiymati TRUE ga teng, agar B to'plam elementlari A to'plam tarkibiga kirsas.

Masalan, quyidagi ifodaning qiymati TRUEga teng:

['C'..'E'] = ['D','E','C']; [7,1,3]<>[2,4,6,8];

[KIEV]<=[MOSKVA, KIEV]; ['A'..'E']>=['A','B'];

IN – to'plamga tegishlilikni tekshirish amali

To'plam elementlariga nisbatan tegishlilik – IN amali belgilangan:

$$X \text{ IN } Y,$$

bu yerda X – to'plam toifasidagi o'zgaruvchi yoki ifoda, Y – to'plam; Bu ifodaning qiymati TRUEga teng, agar X ning qiymati Y to'plamga tegishli

bo'lsa. Yuqorida ta'kidlanganidek, to'plam elementlarini READ operatori yordamida kiritib va WRITE operatori yordamida bosmaga chiqarib ham bo'lmaydi. To'plam elementlarining qiymatini bosmaga chiqarishda IN amaldan foidalaniladi.

Masalan,

```
VAR BB:SET OF 'A'..'Z';
```

```
....
```

```
FOR I:='A' TO 'Z' DO IF IN BB THEN WRITE(1:2);
```

```
....
```

Bu lavhaning bajarilishi natijasida ekranda A dan Z gacha lotin harflari chiqariladi.

To'plam toifasidagi ma'lumotlar ishtirokida masalalar va ularni echish dasturlarini ko'rib chiqamiz.

1-masala. A,B,D do'konlardagi mahsulotlar ro'yxati berilgan. Shular asosida hamma do'kondagi mahsulotlar, faqat bitta magazindagi mahsulotlar, bir vaqtning o'zida hamma do'konlarda bo'lgan mahsulotlar to'plamini yarating.

Masalaning yechish dasturi:

```
PROGRAM Dukon;
```

```
USES PRINTER;
```

```
TYPE MAHS=(NON, YOG', SUT, TVOROG, SIR);
```

```
VAR A,B,C,D:SET OF MAHS; K:MAHS;
```

```
PROCEDURE PR;
```

```
BEGIN FOR K:=NON TO SIR DO
```

```
IF K IN C THEN CASE K OF
```

```
NON:WRITELN(LST,'NON');
```

```
YOG': WRITELN (LST, 'YOG');
```

```
SUT:WRITELN(LST, 'SUT');
```

```
TVOROG:WRITELN(LST,'TVOROG');
```

```
SIR:WRITELN(LST,'SIR')
```

```
END;
```

```
WRITELN; END;
```

```
BEGIN
```

```
A:=[NON,YOG];
```

```
B:=[YOG,SUT,TVOROG];
```

```
D:=[YOG,SUT,SIR];
```

```
C:=A+B+D;
```

```
WRITELN ('MAGAZINLARDAGI MAHSULOTLAR');
```

```
PR;
```

```
C:=A-B-D;
```

```
WRITELN('FAQAT BITTASIDA BO'LGAN MAHSULOTLAR');
```

```
PR;
```

```
C:=A*B*D;
```

```
WRITELN ('BIR VAQTNING O'ZIDA HAMMA DO'KONLARDA  
BO'LGAN MAHSULOTLAR');
```

PR END.

Bu dasturning bajarilishi natijasida ekranga masalada qo'yilgan talablarni qanoatlantiruvchi do'kondagi mahsulotlar ro'yxati chiqariladi.

2-masala. 20 ta butun sonlar to'plamida 5 va 3 soniga karrali bo'lgan sonlar to'plamlarini tashkil qiling.

Masalaning yechish dasturi:

```
Program rr;  
Const n=20;  
Type typ=set of integer;  
Var i:integer; n3,n5: typ;  
Begin n3:=[]; n5:=[];  
For i:=1 to n do  
Begin if i mod 3 then n3:=n3+[i]; if i mod 5 then n5:=n5+[i]  
End;  
Writeln('3 ga karrali sonlar');  
For i:=1 to n do  
if i in n3 then write(i:2);  
Writeln;  
Writeln('5 ga karrali sonlar');  
For i:=1 to n do  
if i in n5 then write(i:2);  
Writeln;  
End.
```

Bu dasturda n3 va n5 mos ravishda 3 ga va 5 ga karrali sonlar to'plami, ularga boshlang'ich qiymat sifatida "bo'sh to'plam" o'zlashtiriladi. Natijada n3 va n5 to'plam elementlari ekranga birma-bir chiqarilgan.

4.7. Ma'lumotlarning faylli toifasi. Toifalashgan va toifalashmagan fayllar.

Paskal algoritmik tilida fayl tushunchasi

Fayl o'zi nima? MS DOS tizimida bu tushuncha kiritilgan edi va fayl deb xotiraning nomlangan sohasiga aytiladi. Faylda turli ma'lumotlar saqlanadi.

Har bir fayl bilan fayl ko'rsatkichi degan tushuncha biriktirilgan. Fayl bir nechta elementlardan iborat bo'lib, faqat faylning ko'rsatkichi ko'rsatayotgan ma'lumotga foydalanuvchi murojat qilishi mumkin. Demak, fizik jihatdan biz faqat ketma-ket fayllarga egamiz. Ya'ni biz oldin birinchi, keyin ikkinchi, uchinchi va h.k. ma'lumotlarni o'qishimiz mumkin. Fayl o'z nomiga ega. Masalan,

d:tp\myfile.dat

Paskal tili dasturiy vositalari yordamida, ya'ni dasturda ham fayllar tashkil qilish va undagi ma'lumotlarni qayta ishlash mumkin.

Shu mavzuga qadar, biz Paskal dasturiy tilida bir nechta o'zgaruvchilarning toifalari bilan ishlab keldik. Bular skalyar, oddiy va murakkab tarkiblashgan toifalardir. Bu toifada ma'lumotlar yordamida masalalarni echishda boshlang'ich ma'lumotlar klaviyaturadan operativ xotiraga kiritiladi va natija ekranga chiqariladi. Ulardan boshqa dasturlarda foydalanib bo'lmaydi, chunki ular tizimidan chiqilgandan so'ng hech qaerda saqlanmaydi. Bu ma'lumotlarni xotirada saqlash uchun Paskal tilida ma'lumotlarning faylli toifasi belgilangan. Fayl toifasi alohida o'rin egallaydi. Fayl toifasi bilan ishlashda ma'lum tushunchalarni o'zlashtirish talab qilinadi.

Birinchi, fayllar toifasi nega va qachon qo'llaniladi? Maqsad nima? zaruriyat nimadan kelib chiqayapti?

Ikkinchisi, boshqa toifalardan nega katta farqi bor?

Bu savollarga faqat foydalanuvchining nuqtai-nazaridan qaragan holda javob bera olamiz:

1. Juda ko'p o'zgaruvchilardan foydalanganicha har doim klaviyaturadan kiritishda ma'lum noqulayliklarga duch kelamiz. Bunga katta massivlar misol bo'la oladi.

2. Shunday masalalar uchraydiki, oldindan kattaliklarning qiymatlar soni noma'lum bo'ladi (masala natijalar), bu kattaliklarni faylga yozish maqsadga muvofiq.

3. Hech qanday toifalar tashqi qurilmalarga murojat qilib, ular bilan ishlashga imkon yaratmaydi (dasturiy til muhitida).

Va nihoyat boshqa toifalardan fayl toifasi farqliligi shundaki, u boshqa toifalar tarkibiga kira olmaydi.

Masalan,

```
Var st:record n:integer;  
    fio:string end;  
    abr:array[1..15] of char;  
    pr:1930..1975;
```

bu toifalar ichida fayl toifasini ishlatish mumkin emas.

Fayllarning toifalari. Fayllar uchun mo'ljallangan umumiy protsedura va funktsiyalar

Faylda saqlanayotgan ma'lumotlar turiga ko'ra Paskal algoritmik tilida fayllar quyidagi turlarga bo'linadi:

- 1) toifalashmagan;
- 2) toifalashgan;
- 3) matnli.

Toifalashgan fayllar bir xil toifali elementlardan tashkil topadi. Ularni faqat ma'lum qurilmalarda uzatish mumkin, lekin ekranda o'qish mumkin emas. Faylning elementlari mashina kodlarida yoziladi va saqlanadi.

Toifalashmagan fayllarda turli toifadagi ma'lumotlarni saqlash mumkin. Ular ham mashina kodlari bilan yozilgan bo'lib, baytlar to'plamini tashkil qiladi.

Matnli (tekst) fayllar ASCII kodlardan tashkil topgan va satrlarga ajratilgan bo'ladi. Matnli fayllarda nafaqat faylning yakunida fayl oxiri belgisi, balki har satrning yakunida maxsus satr oxiri belgisi qo'yiladi.

Fayl toifasidagi o'zgaruvchi fayl o'zgaruvchisi deyiladi, u faylning mantiqiy nomini belgilaydi va u mantiqiy fayl bilan tashqi fayl (fizik) o'rtasida "vositachi" vazifasini o'ynaydi.

Fayl toifasi uchun arifmetik amallar belgilanmagan. Hatto fayllarni solishtirish va bir faylning qiymatini ikkinchi faylga o'zlashtirish amallari ham aniqlanmagan.

Har bir turdagi fayllar ustida umuman olganda quyidagi amallarni bajarish mumkin va bu amallar uchun maxsus protsedura va funktsiyalar ishlatiladi:

1. Turbo Paskal muhitida fayl bilan ishlashdan oldin faylning fizikaviy va mantiqiy nomlarini bog'lash lozim.

Bu alohida protsedura yordamida amalga oshiriladi:

```
Assign (<fayl o'zgaruvchisi>,'<name:string>');
```

Bu yerda name – faylning fizik nomi bo'lib, bunda faylning tashqi (doimiy) xotirada saqlangan yo'li ko'rsatiladi, masalan,

```
Assign(F, 'd:\TP\myfile.dat');
```

Bu protseduraning ma'nosi shundaki, u fayl uchun yo'l ochib dasturdan tashqi qurilmaga murojat qilishi va axborot almashtirish imkonini yaratib beradi.

2. Faylga ma'lumot yozish uchun faylni ochish. Buning uchun quyidagi protsedura ishlatiladi:

```
Rewrite (<fayl o'zgaruvchisi>);
```

Bu protsedura bajarilganda xotirada Assign protsedurasida ko'rsatilgan nom bilan yangi fayl unga ma'lumot yozish uchun ochiladi va fayl ko'rsatkichi faylning boshiga o'rnatiladi. Lekin bu protsedurani ehtiyotkorlik bilan ishlatish kerak, chunki ko'rsatilgan fayl oldindan xotirada bo'lsa undagi ma'lumotlar butunlay o'chirib tashlanadi.

3. Faylni undan ma'lumotlarni tezkor xotiraga o'qish uchun ochish:

```
Reset (<fayl o'zgaruvchisi>);
```

Reset protsedurasi bajarilganda Assign protsedurasida ko'rsatilgan fayl undagi ma'lumotlarni o'qish uchun tayyorlanadi, ya'ni fayl ko'rsatkichi faylning birinchi elementiga keltirib qo'yiladi.

4. Faylga ma'lumotlarni yozish, kiritish. Buning uchun bizga tanish bo'lgan Write operatori quyidagi ko'rinishda ishlatiladi:

```
Write (<fayl o'zgaruvchisi>,<kattalik>);
```

bu yerda kattalik o'rnida o'zgaruvchi yoki ifoda ishlatilishi mumkin. Protsedura bajarilganda kattalikning qiymati fayl o'zgaruvchisi bilan bog'langan faylda fayl ko'rsatkichi o'rnatilgan joyga yoziladi. So'ngra fayl ko'rsatkichi bitta keyingi pozitsiyaga suriladi. Write protsedurasini Rewrite protsedurasi bajarilgandan keyingina ishlatish mumkin.

5. Fayldan ma'lumotlarni o'qish. Bu maqsadda quyidagi protseduradan foydalaniladi:

Read (<fayl o'zgaruvchisi>,<o'zgaruvchi>);

Bu protsedura bajarilganda Reset protsedurasi bilan ochilgan faylda fayl ko'rsatkichi o'rnatilgan joydagi elementning qiymati protseduradagi o'zgaruvchiga o'zlashtiriladi. So'ngra fayl ko'rsatkichi yana bitta pozitsiyaga suriladi.

6. Turli maqsadda ochilgan barcha mantiqiy fayllar albatta yopilishi kerak. Buning uchun quyidagi protsedura mo'ljallangan:

Close (<fayl o'zgaruvchisi>);

Bu protsedura bajarilganda axborot uzatishning barcha kanallari yopiladi.

7. Fayl oxirini aniqlash funktsiyasi:

Eof (<fayl o'zgaruvchisi>);

Bu funktsiyaning qiymati Boolean toifasida bo'lib, u fayl ko'rsatkichi faylning oxiriga o'rnatilganda True qiymatiga ega bo'ladi, aks holda uning qiymati False gat eng.

Fayllarning ixtiyoriy ko'rinishi bilan ishlashda shuni nazarga olish kerakki, bir vaqtning o'zida bitta fayldan unga ma'lumot yozish uchun va undan ma'lumotlarni o'qish uchun foydalanib bo'lmaydi. O'qish uchun yoki yozish uchun ochilgan fayl albatta Close protsedurasi yordamida yopilgan bo'lishi shart.

Endi turli ko'rinishdagi fayllar bilan yaqindan tanishib chiqamiz.

Toifalashgan fayllar va ular bilan ishlash

Toifalashgan fayllar bir xil toifali elementlardan tashkil topadi. Ular dasturda quyidagicha beriladi:

<fayl toifasi>::=file of <elementlar toifasi>

<elementlar toifasi>::=<toifa>

Bu yerda elementlar toifasi faylning tashkil etuvchilari, ya'ni fayldagi ma'lumotlarning toifasi bo'lib, bu toifa sifatida oddiy va murakkab toifalari (fayldan tashqari) ishlatish mumkin.

Toifalashgan fayllarni har doimgidek, Type va Var bo'limlarida tavsiflash mumkin. Masalan,

type fint=file of integer;

tal=file of char;

num=file of real;

```
var p, q:file of integer; f:file of char;  
s:file of real;  
p, q:fint;  
Ff:tal; s:num;
```

Elementlar toifasi o'rnida murakkab toifalarni ham ishlatish mumkin, masalan, yozuvlarni:

```
Type Student=Record  
    Fio: string[12];  
    TY: 1975..1982;  
    Adress:string[15]  
End;
```

```
Var St:File of Student;
```

Bu toifadagi fayllar ustida yuqorida keltirilgan umumiy protsedura va funktsiyalar satrida yana qo'shimcha protsedura va funktsiyalarni ishlatish mumkin. Ba'zi bir protseduralar faylni bevosita murojat fayli sifatida ishlatish imkonini berai.

1. FileSize (<fayl o'zgaruvchisi>); - funktsiyasi fayldagi elementlar sonini aniqlaydi, funktsiyaning toifasi Integer (yoki Longint) bo'lishi kerak.

2. FilePos (<fayl o'zgaruvchisi>):Integer; - funktsiyasi elementning fayldagi o'rnini aniqlab beradi, joriy element deb fayl ko'rsatkichi o'rnatilgan elementga aytiladi.

3. Seek (<fayl o'zgaruvchisi>,<elementning n-tartib raqami>); - protsedura fayl ko'rsatkichini n-elementga o'rnatadi.

4. Truncate (<fayl o'zgaruvchisi>); - protsedurasi o'qilgan fayl elementining keyingisidan boshlab qolgan yozuvlarni olib tashlash uchun ishlatiladi va faylning yakuniy belgisi qo'yiladi.

Toifalashgan fayllar ishtirokida masalalar ko'rib chiqamiz.

```
1) Program F1;  
Var f: file of char;  
ch: char;  
l: integer;  
Begin  
Assign (f,'myfile.dat');  
Rewrite(f);  
For l:=1 to 10 do  
Begin  
Readln(ch);  
Write(f,ch);  
End;  
Close (f);  
Reset (f);  
While not Eof(F) do  
Begin read(F,ch)  
Write(ch,' ');  
End;
```

```
Close (f);  
End.
```

Bu dasturda myfile.dat tashqi fayliga f fayl o'zgaruvchisi yordamida 10 ta ixtiyoriy belgi (Char toifasidagi) yoziladi va bu belgilar vergul orqali ekranga ketma-ket fayldan o'qib chiqariladi.

```
2) Program M;  
Var f:file of Char;  
c: char; l:integer;  
procedure SdF; begin  
reset(f);  
for l:=1 to FileSize(f) do  
begin  
read(f,ch); write (ch,',')  
end;  
close(f);  
end;  
begin Assign(f,'Chfile.txt');  
rewrite(f);  
for l:=1 to 10 do begin  
Read(c); Write(f,c) end;  
SdF; Writeln('fayl')  
Close(f);  
End.
```

4.8. Matnli fayllar va ular uchun mo'ljallangan protsedura va funksiyalar

Yuqorida ta'kidlab o'tilganidek, matnli fayl satrlardan tashkil topgan fayldir. Matnli fayllarda:

- a) ma'lumot matn shaklida ASCII kodlar jadvalining simvollarida tasvirlanadi;
- b) ma'lumotlar satrlarga bo'linishi mumkin;
- c) faylning oxiri "^Z" belgi bilan belgilanadi;
- d) sonlar, mantiqiy qiymatlar, satrlar Char toifasidagi ma'lumotlarga aylantiriladi va mashina kodlarida yoziladi.

Dasturda matnli fayllar Text xizmatchi so'zi yordamida beriladi:

```
<fayl o'zgaruvchisi> : Text;
```

Turbo Paskal muhitida ikkita matnli fayl o'zgaruvchilari standart ravishda aniqlangan. Bu INPUT va OUTPUT o'zgaruvchilaridir. Bular avtomatik ravishda 'CON' (konsol) mantiqiy qurilma bilan bog'langan, ya'ni dasturda ma'lumotlarni kiritish va chiqarish aynan shu fayllar orqali amalga oshiriladi, lekin dastur sarlavhasida va tavsiflash bo'limida ularni e'lon qilish

shart emas. Bu fayllar Turbo Paskal tizimi yuklanishi bilan avtomatik tarzda tezkor xotiraga yuklanadi.

Umumiy fayl o'zgaruvchisiga mansub protseduralardan tashqari matnli fayllarga quyidagi funktsiya va protseduralar mo'ljallangan:

1. *Readln(<fayl o'zgaruvchisi>,<o'zgaruvchi>)* – fayldan simvollar satrini o'qish protsedurasi. Bu protsedura bajarilganda fayl ko'rsatkichi o'rnatilgan satr o'zgaruvchiga o'zlashtiriladi, faylda satr “satr oxiri belgisi” yordamida ajratib olinadi.

2. *Writeln(<fayl o'zgaruvchisi>,<satr>);* - faylga simvollar satrini yozish protsedurasi. Protседura bajarilganda fayl ko'rsatkichi o'rnatilgan joyga <satr> yoziladi.

3. *Append(<fayl o'zgaruvchisi>);* - protsedurasi.

Bu protsedura faylni unga qo'shimccha elementlarni (ma'lumotlarni) yozish uchun ochadi. Bu protsedura xotirada saqlangan fayllar uchun ishlatilib, Rewrite protsedurasi o'rnida keladi.

4. *Eoln(<fayl o'zgaruvchisi>);* - funktsiyasi fayldagi joriy satrning oxirini aniqlaydi. Funktsiyaning toifasi Boolean (mantiqiy) bo'lib, fayl ko'rsatkichi satr oxiriga o'rnatilganda funktsiyaning qiymati TRUEga, aks holda False ga teng bo'ladi.

5. *SeekEoln(<fayl o'zgaruvchisi>);* - funktsiyasi satr yakuniga erishilganini aniqlaydi.

5. *SeekEof(<fayl o'zgaruvchisi>);* - funktsiyasi faylni yakuniga erishilgan yoki erishilmaganligini aniqlaydi.

Eof va SeekEof, Eoln va SeekEoln funktsiyalarining farqi shundaki, Eof va Eoln fizikaviy faylga nisbatan, SeekEof va SeekEoln esa mantiqiy faylga nisbatan qo'llaniladi.

1-misol: Matnli faylning satrlarini bosmaga chiqarish.

```
Program f5;  
Var  
ft1:text;  
fln:string;  
Begin  
Assign(ft1,'d:\myfile.txt');  
Reset (ft1);  
While not eof(ft1) do  
Begin  
readln(ft1,fln);  
Writeln(flн)
```

```
End;  
Close(ft1)  
End.
```

Bu dasturning bajarilishi jarayonida S o'zgaruvchi uchun 20 ta belgilar satri klaviyaturadan kiritiladi va ular ketma-ket file mantiqiy fayliga yoziladi. So'ngra fayl o'qish uchun ochiladi va undagi ma'lumotlar satrma-satr o'qiladi.

2-misol: Matndagi "a" harflar sonini aniqlash.

```
Program ah;  
Var  
file:text; s:string; n:byte; c:char; l:integer;  
Begin  
Assign(file,'c:\textfile.txt');  
Rewrite(file);  
For l:=1 to 20 do  
Begin readln(s);  
writeln(file,s);  
End;  
Close(file);  
Reset(file); n:=0;  
While not eof(file) do  
    While not eoln(file) do  
Begin read(file,c) do  
If (c='a') or (c='A') then n:=n+1; end;  
Write('a – xarflar soni',n,' ta');  
Close(file) end.
```

Fayldagi a harflari soni ekranga chiqariladi (n).

Toifalashmagan fayllar

Turbo Paskal dasturiy tilida alohida ahamiyatga ega bo'lgan fayllardan, ya'ni toifalashmagan fayllardan foydalanish mumkin. Bu fayllarni umumlashgan toifa deb atasak yanglishmaymiz. Faylning toifalashmagan deb atalishidan maqsad, fayl turli toifadagi ma'lumotlardan tashkil topadi.

Toifalashmagan fayllarni tavsiflashda elementlar toifasi ko'rsatilmaydi, faqat File xizmatchi so'zidan foydalaniladi:

```
Var <fayl o'zgaruvchisi>: File;
```

Toifalashmagan fayllarga nisbatan ma'lumotlarni kiritish, ya'ni ma'lumotlar faylini yaratish, ma'lumotlarni fayldan o'qish, tezkor xotirada fayl elementlarini qayta ishlash kabi amallarni bajarish mumkin.

Fayldagi elementlar toifasi oldindan ma'lum bo'lmagani uchun undagi ma'lumotlar bir xil uzunlikdagi bloklarga (yozuvlarga) ajratilib o'qiladi va shu

tarzda faylga yoziladi. Blok uzunligi baytlarda olinadi. Shu holdan kelib chiqqan holda bu funktsiyalarni ishlatish mumkin:

1. Reset(F, S); - protsedurasi faylni o'qish uchun ochadi (yuqorida berilgan Reset protsedurasi o'rnida ishlatiladi), bu yerda F - fayl o'zgaruvchisi, S – har bir blok uchun belgilangan xotira hajmi (baytlarda olinadi).

2. BlockRead (F,V,N); - protsedurasi, bu yerda F - fayl o'zgaruvchisi, N – o'qilishi lozim bo'lgan bloklar soni (integer), V – o'qilgan bloklar joylashtiriladigan xotiradagi birinchi adres raqami (integer, Word). Bu protsedura bajarilganda F va V da joylashgan S uzunlikdagi N bloklar o'zlashtiriladi.

3. Rewrite (F, S); - protsedurasi F fayliga S uzunlikdagi yozuvlarni yozish uchun faylni ochadi.

4. BlockWrite(F,V,N); - protsedurasi F fayliga tezkor xotiraning V adresli joydagi N ta yozuvni joylashtiradi.

5. FilePos(F) – funktsiyasi joriy blokning tartib raqamini aniqlaydi.

6. FileSize(F) – funktsiyasi fayldagi bloklar uzunligini aniqlab beradi.

Masalan, quyidagi dastur lavhada F-faylini ochib unga uchta blok ma'lumotlarini yozishga yordam beradi:

```
Assign (F,'ABC.dat');
```

```
Rewrite (F,size);
```

```
BlockWrite(f,a,3);
```

```
Close(F);
```

Bu ma'lumotlarni fayldan o'qish quyidagi lavha yordamida bajariladi:

```
Reset(F,size);
```

```
BlockRead(F,A,3)
```

```
Close(F);
```

Yana shuni ta'kidlash lozimki, toifalashmagan fayllarni qo'llash tizim doirasidagi xotiradan unumli foydalanishga yordam beradi.

4.9. Ma'lumotlarning murojaat toifasi. Ko'rsatkichlar Ko'rsatkich tushunchasi

Biz shu vaqtgacha ishlatib kelgan o'zgaruvchilar, ya'ni tavsiflash bo'limining Var bo'limida tavsiflangan barcha o'zgaruvchilar static o'zgaruvchilar bo'lib, bu o'zgaruvchilarga dastur bajarilishiga qadar, kompilyatsiya vaqtida ularning toifasiga qarab ma'lum miqdorda xotiradan aniq joy ajratiladi. Kompyuterda xotiraning bu maydoni (katta) ma'lumotlar segmenti deb ataladi.

Segment data -65536 bayt

Katta miqdordagi ma'lumotlar (katta o'lchamdagi massivlar) ishlatiladigan masalalarni yechishda, hamda kompyuterning tovush va grafik imkoniyatlarini ishlatishda xotira hajmi yetmasligi mumkin. Bunday hollarda dinamik xotiradan foydalanish maqsadga muvofiqdir (near). Dinamik xotira bu shaxsiy kompyuterning dasturga ma'lumotlar segmentidan tashqari yuklatilgan tezkor xotirasidir. Bu xotira taxminan 200-300 kbaytni tashkil qiladi. Shuning uchun katta hajmdagi ma'lumotlarni dinamik xotiraga joylashtirish yaxshi samara beradi.

Dinamik xotiradan foydalanish uchun ma'lumotlarning maxsus turi – murojaat toifasi aniqlangan. Bu turning kattaligi sifatida maxsus dinamik o'zgaruvchi, ya'ni ko'rsatkichlar aniqlangan. Bu o'zgaruvchilarni xotirada joylashtirishni esa kompilyator amalga oshiradi.

Ko'rsatkich shunday o'zgaruvchiki, uning qiymati o'zgaruvchi qiymatiga emas, balki shu o'zgaruvchi qiymati joylashgan xotira adresiga tengdir.

Ko'rsatkichlarni ishlatishning o'ziga xos xususiyati shundaki, ularga static o'zgaruvchilar singari biror nom bilan murojaat qilib bo'lmaydi. Bundan tashqari, bu toifadagi o'zgaruvchilar dasturning bajarilishi davomida yaratilishi va yo'qotilishi mumkin.

Tezkor xotira tarkibi

Ko'rsatkichlar bilan ishlashdan avval tezkor xotira tarkibi bilan qisqacha tanishib chiqamiz. Tezkor xotira maydoni 9 ta qismga bo'linadi, bu qismlarning har biri aniq turdagi axborotni saqlash uchun xizmat qiladi.

1. Amaliyot tizimi va qobiqlar;
2. Paskal kutubxonalarini;
3. Interfeys, tahrirlagich, kompilyatorlar;
4. Xatolar haqida ma'lumot beruvchi fayl;
5. Boshlang'ich modul;
6. Ob'ekt kodi;
7. Dinamik xotira;
8. Rekursiv stek;
9. Markaziy protsessor styoki;

Kompyuter xotirasining 1-6 qismlari avtomatik ravishda shakllanadi va amaliyot tizimini, kutubxona funktsiya va modullarini, tahrirlovchi, qayta ishlovchi dasturlardan tashkil topadi. Foydalanuvchi dasturi boshlang'ich modul sifatida yozilib, dasturning ob'ekt kodiga kompilyatsiya qilinadi. Dinamik xotira faqat dasturchi qo'llashi mumkin bo'lgan maxsus ko'rsatkichlar bilan to'ldiriladi.

Rekursiv stek rekursiv protseduralar bilan ishlash jarayonida avtomatik ravishda shakllanadi. Markaziy protsessor styoki maksimal qo'llash mumkin bo'lgan adres bilan boshlanadi. Ko'rsatkich doimo adres kamayib boorish tomoniga siljib boradi. Dinamik xotiraning boshlang'ich nuqtasi «adres» Heap org deb, oxiri – Heap end deb ataladi.

Ko'rsatkich turlari va ularni tavsiflash

Yuqorida ta'kidlab o'tganimizdek, ko'rsatkich ma'lum toifadagi o'zgaruvchining qiymati yozilgan xotira adresini saqlaydigan o'zgaruvchidir, ya'ni u o'zgaruvchining qiymati joylashgan adresga murojaat qiladi.

Turbo Paskalda ko'rsatkichlar aniq bir toifadagi ma'lumotlar yoki turli toifadagi ma'lumotlar yozilgan adreslarga murojaat qilishi mumkin. Shunga ko'ra ular toifali va toifasiz ko'rsatkichlarga bo'linadi. Toifali ko'rsatkichlar quyidagicha tavsiflanadi:

```
<Ko'rsatkich>:^<toifa>;
```

bu yerda ^ - ko'rsatkich belgisi, <toifa> - ko'rsatkich murojaat qilayotgan toifa nomi yoki toifaning berilishi. Masalan:

```
TYPE Mas=array[1..100] of real;  
VAR P: ^Integer; F: ^Mas;
```

Ko'rsatkich biror qiymatni qabul qilgandan so'ng, ya'ni berilgan adres bo'yicha aniq fizik baytlarni ko'rsatgandan so'ng, u yerga dasturning bajarilishi davomida mos toifaning ixtiyoriy qiymatini joylashtirish mumkin.

Ko'rsatkichlar toifasiz bo'lishi ham mumkin. Bu holda ular yordamida toifasi oldindan noma'lum bo'lgan va dastur bajarilishi davomida o'zgaradigan kattaliklarni xotiraga dinamik joylashtirish mumkin. Shunday qilib, Turbo Paskalda ko'rsatkichni aniq bir toifaga bog'lamasdan e'lon qilish mumkin. Bu esa maxsus – Pointer so'zi yordamida amalga oshiriladi:

```
<Ko'rsatkich> : Pointer;Masalan:  
Var k: Pointer;
```

Ko'rsatkichlarni va dinamik xotirani ishlatishda qo'llaniladigan asosiy protseduralar

Dasturda ko'rsatkichlar bilan ishlashdan avval ularni yaratish, ular hajmini aniq belgilash, ko'rsatkich murojaat qiladigan dinamik xotira sohasini aniqlash lozim. Buning uchun quyidagi protsedura va funktsiyalar mo'ljallangan.

I. Toifali ko'rsatkichlar ustida quyidagi amallarni bajarish mumkin:

1) NEW (<ko'rsatkich>) protsedurasi – yangi toifalashgan dinamik o'zgaruvchi (ko'rsatkich) uchun joy ajratadi va uning manzilini ko'rsatkichga joylaydi.

Har qanday ko'rsatkich bilan ishlashdan avval, uni dinamik xotirada yaratish kerak. So'ngra u murojat qilayotgan adresga ma'lumot kiritish mumkin.

Parametrlar (ko'rsatkich) bir necha bo'lishi mumkin. Kompyuter dinamik o'zgaruvchining qiymatini saqlash uchun maxsus yacheyka ajratadi. Agar bu qiymat yozuv ko'rinishida bo'lsa, u holda kompyuter uning

ixtiyoriy varianti (elementi) uchun zarur bo'lgan xotira ajratadi. Buning uchun quyidagi protsedura ishlatiladi.

NEW(p,t1,t2,...,tn)

Shunday qilib, bu protsedura tanlash konstantasiga mos variantli dinamik o'zgaruvchisini yaratadi. Tanlash konstantalari yozuvda qanday tavsiflangan bo'lsa shunday tartibda ko'rsatilishi kerak.

Misol.

```
Tupe aq(a1,a2);
    rec=record
    h:integer;
    case:a of
    a1(ch:char);
    a2(v:array[1..1000] of real);
end;
var p:=^rec;
```

Butun, qayd qilingan, belgili o'zgaruvchilarning bitta qiymatini saqlash uchun xotiraning bitta so'zi yetarli bo'ladi, haqiqiy o'zgaruvchi uchun esa ikkita. Masalan, NEW(p)-p^ dinamik o'zgaruvchi uchun 2002 ta so'z ajratiladi,

```
NEW(p,a1)-p^ uchun 3 ta so'z,
NEW(p,a2)-p^ uchun 2002 ta so'z.
```

2) DISPOSE(<ko'rsatkich>) protsedurasida – toifalashgan ko'rsatkichni yo'qotadi. Aslida bu protsedura NEW protsedurasiga teskari protsedura. Bu protsedura NEW protsedurasida yaratilgan dinamik o'zgaruvchi kerak emasligini ko'rsatadi. Dinamik o'zgaruvchi uchun ajratilgan joy bo'shaydi. Dinamik xotiraning bu sohasini boshqa maqsadda ishlatish mumkin. Ko'rsatkichli o'zgaruvchilarning qiymati NEW protsedurasida bajarilishidan oldin qanday bo'lsa, shunday noaniq bo'lib qoladi.

Masalan, P: DISPOSE(p)

NEW protsedurasida singari DISPOSE protsedurasida ham ko'pvariantli parametrdan iborat bo'lishi mumkin.

Bu holda protsedura quyidagicha yoziladi:

```
DISPOSE (p,t1,t2,...,tn)
```

DISPOSE protsedurasida identifikatorni va o'zgaruvchining qiymatini yo'q qilib tashlaydi.

Masalan:

```
procedure p;
var i: integer;
    ri: ^integer;
begin
    i:=5;
```

```
new(ri);
ri^:=i+1
i:=ri^*2;
dispose(ri);
end.
```

II. Toifasiz ko'rsatkichlar uchun qo'llaniladigan protsedura va funktsiyalar:

1) GETMEM (p:pointer; size:word);

Bu yerda p – ko'rsatkich, SIZE – xotira o'lchami.

Bu protsedura yuqorida ko'rib o'tilgan NEW protsedurasining funktsiyasini toifasiz ko'rsatkichlar uchun bajaradi, faqat bunda ko'rsatkich uchun SIZE hajmda dinamik xotira ajratiladi.

Misol,

GETMEM (p, 1000)

dinamik o'zgaruvchi uchun 1000 bayt joy ajratilgan.

2) FREEMEM (p:pointer; size:word); - bu protsedura SIZE hajmdagi dinamik xotirani yo'qotadi.

Masalan,

FREEMEM (p,1000) – protsedura xotiradan 1000 bayt joyni bo'shatadi.

3) MARK (p:pointer) – protsedurasi joriy vaqtdagi egallangan dinamik xotiraning oxirini p o'zgaruvchisiga yozadi.

4) MEMAVAIL: Longint – funktsiyasi dinamik xotirada bo'sh maydonlar hajmini aniqlaydi.

5) MAXAVAIL: Longint – funktsiyasi eng katta bo'sh maydonni tanlab, uning hajmini ko'rsatadi.

Ikki turdagi ko'rsatkichlar uchun NIL konstantasi belgilangan.

Const NIL toifasizdir, uning qiymatini va so'zini ixtiyoriy ko'rsatkichga berish mumkin:

1. Ko'rsatkichga NIL konstantasining qiymatini berish mumkin. Boshqa toifadagi ko'rsatkichlar ishlatilganda amalning chap tomonida joylashgan o'zgaruvchining toifasi o'ng tomonga mos kelishi kerak.

2. Ko'rsatkichlar ustida taqqoslash amallarini – (=,<,>,<>) qo'llash mumkin. Amalning ikki chetidagi operandlar bir xil toifada bo'lishi kerak.

Misol:

```
Log:=ri <> NIL,
    if (rr=rrr) or (ri<>NIL) or (ri=NIL) then...,
    while ri <> NIL do.
```

O'z tuzilishiga ko'ra ko'rsatkich yozuvni tashkil etadi. Uning birinchi maydoni qiymatni, ikkinchi maydoni unga mos adresni ko'rsatishi kerak.

Bu esa juda ko'p masalalarni yechish imkonini beradi.

Ko'rsatkichlar hali tavsiflanmagan kattaliklar toifasiga murojat qilishi mumkin.

```

Type ppointer = ^PRecord
                Precord=record
                Name: string;
                Job: string;
                Next: ppointer
                End;

```

```

Var p:ppointer;

```

Misol: bir necha ketma-ketlikdagi butun sonlardan iborat fayl berilgan. Har bir ketma-ketlik manfiy son bilan tugallangan. Bu sonlar ketma-ketligini chiqish fayliga shunday joylashtirish kerakki, har bir ketma-ketlikda sonlar teskari tartiblansin, masalan,

```

1.37 – 1      12067 – 1 kirish fayli
7.31 – 1      76021 – 1 chiqish fayli

```

Ketma-ketliklar uzunligi oldindan noma'lum bo'lgani uchun ma'lumotlarning dinamik strukturasi ishlatiladi.

```

program listf;
type ppointer=^dataset
                dataset=record
                data : integer;
                point: pointer;
                end;
var r1, r2:pointer;
    i:integer;
    inp, outp:file of integer;
begin reset(inp);
      rewrite (outp);
while not(eof (inp)) do
  begin r1:=nil;
        read (inp,i);
while i<>-1 do
  begin new(r2);
        r2^data:=i
        r2^point:=r1; r1:=r2;
        read (inp,i);
        end;
write (outp, -1);
end.

```

Kompyuterda barcha turdagi ko'rsatkichlar bir xil ko'rinishga ega, ya'ni bular kompyuter xotirasining adresidir. Shuning uchun kompyuterda fizik nuqtai nazardan ko'rsatkichlar toifasiz deb hisoblanadi. Toifali ko'rsatkichlarni qo'llash ko'rsatkichlarni noto'g'ri ishlatish natijasida yuzaga keladigan xatolarni aniqlashda yordam beradi.

4.10. Dinamik o'zgaruvchilardan foydalanish

Ko'rsatkichlardan dinamik xotirada saqlanadigan ro'yxatlar tuzishda samarali foydalanish mumkin.

Ro'yxatlar o'z navbatida dinamik strukturaning quyidagi toifalariga bo'linadi.

1. Stek (oqim)
2. Navbat
3. Daraxt

Ro'yxat – bu tartiblangan shunday strukturaki, uning har bir elementi keyingi element bilan bog'lanuvchi murojaatdan tashkil topadi.

Ro'yxat elementi yozuv singari tashkil etilib, asosiy va qo'shimcha deb ataladigan ikkita qismdan iborat bo'ladi. Bular aniq qiymatli har xil toifali o'zgaruvchilar bo'lishi mumkin.

Qo'shimcha qismda ro'yxatning keyingi yozuvida yozuviga murojat qiluvchi ko'rsatkich joylashadi. Ro'yxatning boshi doimo dasturda ishtirok etuvchi o'zgaruvchi orqali ko'rsatiladi.

Agar ro'yxat bo'sh deb atalsa, qo'shimcha qismga NIL qiymat beriladi. Ro'yxatni tugatish uchun ham ro'yxat oxirgi elementining qo'shimcha qismiga NIL qiymat berish kerak.

Stek – dinamik strukturasi.

Stek bu shunday ro'yxatki, uning elementiga faqat bir nuqtadan murojar qilish muki. Stek elementini olib tashlash yoki qo'shish faqat uning cho'qqisi orqali amalga oshiriladi. Bunday struktura LIFO – “oxirida kirib birinchi chiqish” deb ataladi.

Ro'yxatli strukturalr ustida asosan formatlash, qo'shish, olib tashlash, ko'rish kabi amallarni bajarish mumkin.

```
TYPE ukar=^stack;
      Stack=record
      Inf:integer; {axborot qismi}
      Next: ukar; {qo'shimcha qism}
      End;
Var top, kop, nevel:ukar;
Procedure sozds;
Begin
  top:=NIL;
  while TRUE do
    begin read(value);
      if value = 999 then exit;
    NEW (kop);
    Korl.next:=top
    Korl.inf:=value;
    top:=kop
  end;
end.
```

Stekni ishlatganda quyidagi holatlar yuzaga kelishi mumkin:

1. Stekning to'lib ketishi, ya'ni stek xotirasida joy qolmaslik..
2. To'lmaslik holati – stekdan u bo'sh bo'lganda o'qishga harakat qilish.

Navbat-ma'lumotlarning shunday strukturasi, uning bir tomonida element qo'shib borilsa, ikkinchi tomonidan olib tashlanadi.

Bunday strukturani tashkil qilish uchun LEFT va RIGHT o'zgaruvchilari ishlatiladi.

Navbatga element qo'shilayotganda, elementlar RIGHT o'zgaruvchisining qiymatiga mos xotiraga joylashadi. Shunday qilib, RIGHT xotiraning bo'sh joyini ko'rsatadi.

Navbatdan elementlarni tanlash navbatining keyingi elementini ko'rsatuvchi qiymat orqali amalga oshadi. Agar LEFT qiymati RIGHT qiymatiga teng bo'lsa, u holda navbat bo'sh hisoblanadi.

Navbat ustida ham quyidagi amallarni bajarish mumkin.

1. navbatni tashkil qilish;
2. navbatga qo'shish;
3. navbatdan olib tashlash;
4. navbat elementlarini ko'rish.

Shunday qilib, navbat aylana shaklidagi ro'yxatlardan iboratdir.

V bob. Turbo Pascal dasturining grafik imkoniyatlari

5.1. Turbo Pascal dasturining grafik operatorlari

Ma'lumki kompyuter monitori grafik va matnli ma'lumotlarni ekranga chiqaradi. Monitor maxsus videoadapter boshqaruvida matnli va grafikli holatda ishlaydi. Matnli ish holatida kompyuter ekрани har bir satrda 80 tadan pozitsiya bo'lgan 25 ta satrga bo'linadi. Grafikli ish holatida esa monitordagi har qanday tasvir xuddi televizor ekranidek, ya'ni ma'lum bir ranga bo'yalgan nuqtalar to'plami kabi hosil bo'ladi. Tasvirning qanchalik tiniqligi videoadapterning imkoniyatiga, ya'ni gorizontal va vertikal bo'yicha chiqariladigan nuqtalar miqdoriga qarab belgilanadi. Masalan, videoadapter EGA ekranga har bir gorizontal satr bo'yicha 640 ta nuqta, vertikal bo'yicha 350 ta nuqtani beradi, ya'ni ekranda $640 \times 350 = 224000$ ta nuqta hosil qiladi. Nuqtalar soni qancha ko'p bo'lsa tasvir shuncha tiniq chiqadi.

VGA - 640x480

SVGA- 640x480, 800x600, 1024x768

Ekranni grafik holatga o'tkazish

Oddiy matnli holatdan grafik holatga o'tish uchun GRAPH modulining InitGraph prosedurasi ishlatiladi.

InitGraph(Gd,Gm,Path);

bu erda Gd - drayver nomeri;

Gm - rejim nomeri;

Path - kerakli drayverga yo'l ko'rsatadi.

Agar Path=' ' bo'sh bo'lsa drayverni joriy katalogdan izlaydi. Izoh: drayver kerakli videoadapterni ishga tushuradigan dastur.

Agar Gd=0 bo'lsa, kerakli drayverni o'zi avtomatik ravishda tanlaydi, ya'ni Gd=Detect. Detect nolga teng parametr.

Graph - modulini chaqirish dastur bosh qismida beriladi, ya'ni

Uses Graph;

Grafik rejimni oldingi holatiga qaytarish uchun, ya'ni yopish uchun **CloseGraph**; prosedurasi ishlatiladi.

Quyidagi dastur grafik holatni ishga tushurib, yana uni yopadi.

Program Gra;

uses Graph;

Var Gd,Gm: Integer;

Begin

Gd:=Detect; {avtomatik ravishda drayverni tanlaydi}

InitGraph(Gd,Gm,' c:/Tp'); {grafik holatga o'tish}

Readln; {Enter tugmasini bosishni kutish}

```
CloseGraph; {gafik holatni yopadi}  
End.
```

5.2. Rasm chizish va rang berish protseduralari

Ekranida har bir nuqta o'z koordinatasiga ega. Koordinata boshi ekranning yuqori chap burchagi bo'lib, u (0,0) dan boshlanadi. Nuqtalar koordinatasi pastga va o'ngga qarab o'sib boradi.

Quyidagi rasm chizish prosedura va funktsiyalarini ko'rib chiqamiz:

PutPixel(x,y,color); -prosedura ekranida koordinatasi (x,y) bo'lgan nuqtani beradi. Color parametri shu nuqtaga rang beradi.

Misol: PutPixel(100,120,Red);

bu erda ekranning (100,120) nuqta qizil rangda tasvirlanadi. Red=4 nuqtaga qizil rang beradi.

GetPixel(x,y); -funktsiyasi nuqta rangini aniqlaydi.

Misol: Col:=GetPixel(50,80); bu erda Col parametriga nuqta rangi yuboriladi.

Line(x1,y1,x2,y2); -prosedura koordinatalari (x1,y1) va (x2,y2) bo'lgan kesmani chizadi.

Circle(x,y,Radius); - markazi (x,y) nuqta bo'lgan va radiusi Radius bo'lgan aylana chizadi.

Arc(x, y, a, b, r); – markazi x va y nuqta bo'lgan yoy chizadi. a va b yoyning boshlang'ich va oxirgi burchaklarini aniqlaydi. R yoy radiusi hisoblanadi.

Rectangle(x1,y1,x2,y2); -prosedura to'g'ri to'rtburchak chizadi.

(x1,y1) -yuqori chap burchak koordinatasi;

(x2,y2) -pastki o'ng burchak koordinatasi.

SetColor(Color); -prosedura rasmga rang beradi. Color rang nomeri.

Quyidagi ranglar ishlatiladi.

Black=0	qora
Blue=1	ko'k
Green=2	yashil
Cyan=3	bryuzarang
Red=4	qizil
Magenta=5	malinarang
Brown=6	jigarrang
LightGray=7	ochiq kulrang
DarkGray=8	to'q kulrang
LightBlue=9	och havorang
LightGreen=10	och yashil
LightCyan=11	och bryuzarang
LightRed=12	och qizil
LightGray=13	och malinarang
Yellow=14	sariq
White=15	oq

Misol 1.

14 xil rangda kesmalarni chizish dasturini tuzing (qora va oq ranglardan tashqari)

Program M_1;

Uses Graph;

```

Var
Gd, gm, c, x, y, i: integer;
Begin
    Gd:=detect;
    InitGraph(gd,gm,' ');
    Setbkcolor(white);
    ClearDevice;
    x:=0;
    for c:=0 to 14 do
    begin
        x:=x+35;
        Setcolor(c);
        Line(x, 0, x, 400);
        for i:=1 to 5 do
            Line(x+i, 0, x+i, 400);
        end;
        readln;
    CloseGraph;
    end.

```

Misol 2.

Quyidagi dastur to 'g 'ri to 'rtburchak chizadi.

```

Program TT;
Uses Graph;
Var gd,gm: Integer;
Begin
    gd:=detect;
    InitGraph(gd,gm,' ');
    Rectangle(30,30,120,120);
    Readln;

```

CloseGraph;
End.

5.3. GRAPH modulining boshqa funksiya va proseduralari

Bar(x1,y1,x2,y2); -rangli yoki shtrixlangan to'g'ri to'rtburchak chizish;

Bar3d(x1,y1,x2,y2,depth,top); -rangli yoki shtrixlangan paralelepiped chizish;

FileEllipse(x,y,xradius,yradius); -rangli yoki shtrixlangan ellips chizish;

SetFillStyle(Style,Color); -shtrix va rang berish. Bu erda Style shtrix, Color rang tanlovchi o'zgarma parametr.

Style o'zgarma parametr bo'lib, u har xil shtrixlar bilan figuralarni to'ldiradi. Ular quyidagilardir:

Const

EmptyFill=0;	{ fon rangi }
SolidFill=1;	{ joriy rangni beradi }
LineFill=2;	{ qalin gorizantal chiziq }
StstashFill=3;	{ ingichka qiyshiq chiziq }
StoshFill=4;	{ qalin qiyshiq chiziq }
BkStashFill=5;	{ qalin qiyshiq chiziq }
LtstashFill=6;	{ qiyshiq yo'l-yo'l chiziq }
HatchFill=7;	{ katakchalar bilan to'ldirish }
XhatchFill=8;	{ qiyshiq katak bilan to'ldirish }
InterLeaveFill=9;	{ qiyshiq shtrix chiziq }
WideDotFill=10;	{ kam nuqtalar bilan to'ldirish }
CloseDotFill=11;	{ bo'lak-bo'lak nuqtalar }
UseFill=12;	{ foydalanuvchi shtrixi }

Grafik rejimda matn yozish uchun quyidagi proseduralar ishlatiladi.

SetTextStyle(Font,Detection,Style); -kerakli shriftni ishga tushiradi. Bu erda

Font - shriftni tanlash;

Detection - yozuv yo 'nalishini belgilash;

Style - shrift o 'lchamini tanlash.

Shrift va matn yozuvi yo 'nalishi quyidagi o 'zgarmas bilan aniqlanadi.

Const

{ shrift }

DefaultFontq0; { standart shrift }

TriplexFontq1; { vektorli shrift }

{ tekst yo 'nalishi }

HarizDir=0; { chapdan o 'nga }

VertDir=1; { pastdan yuqoriga }

OutTextXY(x,y,TextString); -(x,y) koordinatali nuqtadan TextString nomli matn satrini kiritadi.

VI bob. Turbo Pascal tilida amaliy va laboratoriya mashg'ulotlarni tashkil etish.

Turbo Pascal dasturlash tilidan olingan nazariy ma'lumotlarni amalda qo'llash uchun mavjud misol va masalalarga uning imkoniyatlaridan foydalangan holda dasturlar tuzish va dasturlarni natijalarini hosil qilish, tahlil qilishda amaliy – laboratoriya mashg'ulotlarining ahamiyati katta. Amaliy mashg'ulotlarda berilgan masalalarga blok sxemalar va dasturlar tuzish ishlarini bajarsa, laboratoriya mashg'ulotlarida esa tuzilgan dasturlarning natijalari olinadi va tahlil qilinadi.

Amaliy va laboratoriya ishlarini o'tkazish uchun ko'rsatmalar:

- I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).
- II. O'quvchilarga mavzu bo'yicha masalalarni dasturini tuzib, natijalar olish taklif etiladi.
- III. O'quvchilar amaliy ish masalalarini o'qituvchiga topshiradilar va baholanadilar.

6.1. Amaliy mashg'ulotlar. 1-amaliy ish.

Mavzu: TURBO PASCAL dasturlash tilining asosiy operatorlari yordamida chiziqli dasturlar tuzish.

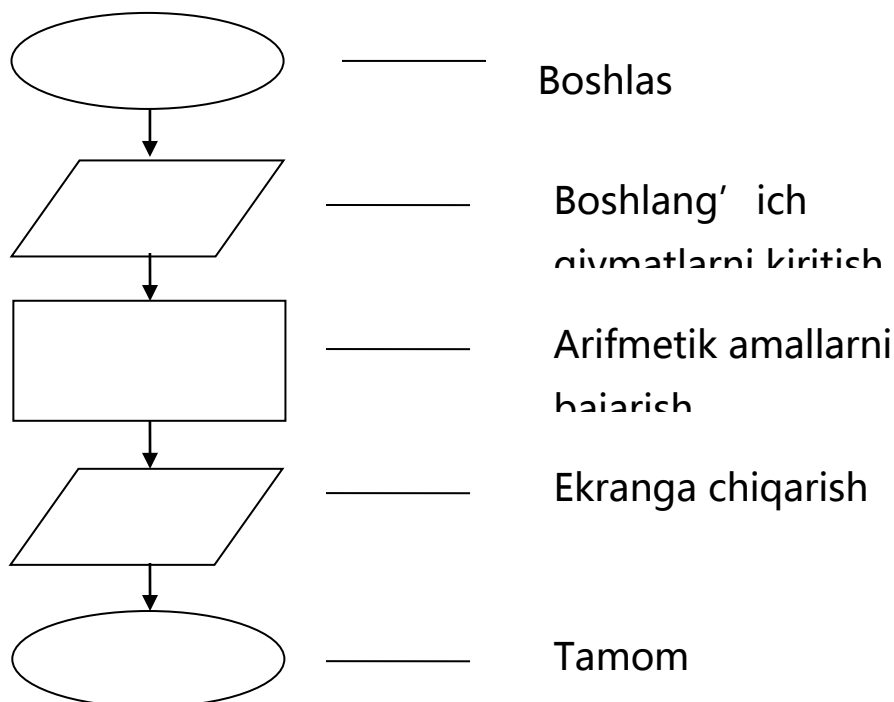
MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: chiziqli dastur tuzish uchun quyidagi operatorlar zarur bo'ladi. := -O'zlashtirish operatori, read(a); va readln(a); - kiritish operatorlari write(a); va writeln(a);- Chiqarish operatori.

Chiziqli dastur tuzish uchun quyidagilarni eslab olish kerak:

- Chiziqli dasturda buyruqlar ketma-ket bajariladi.
- Chiziqli dastur algoritmi quyidagicha ko'rinish olishi mumkin:



Kerakli dasturiy va metodik qo'llanmalar: Turbo-Pascal dasturlash muhiti, tarqatma materiallar, Turbo-Pascal dasturida tuzilgan chiziqli dasturlar.

AMALIY ISHNI BAJARISH TARTIBI.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Paralleloipedning hajmini hisoblash dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan. (Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shriftlarda ko'rsatilgan).

Paralleloipedning hajmini hisoblash.

Eni (sm): **7.5**

Bo'yi (sm): **2.5**

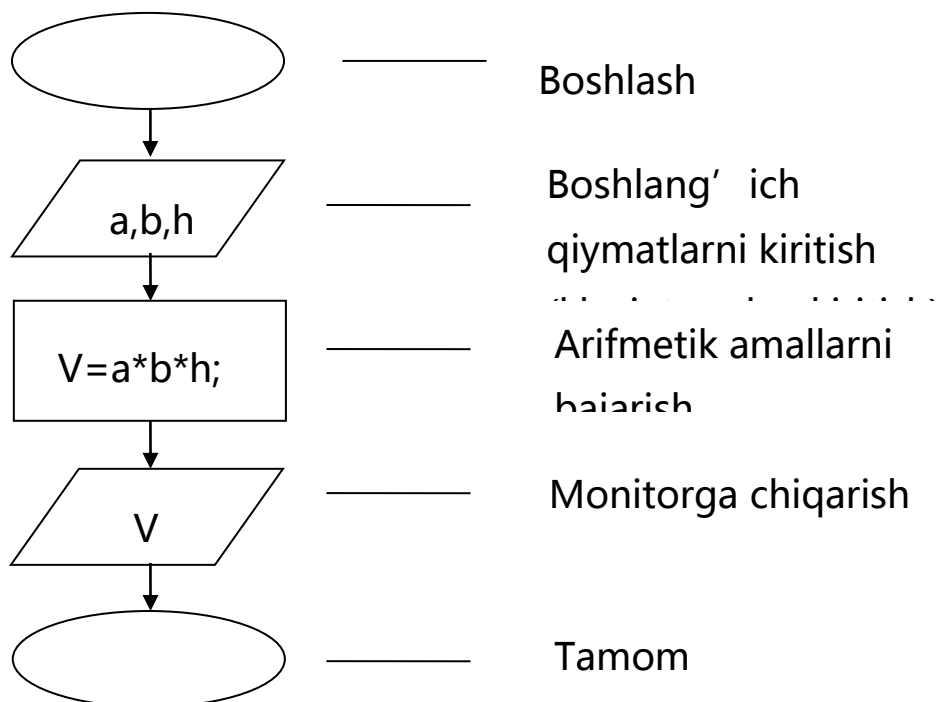
Balandligi (sm): **3**

Paralleloiped hajmi: 56.25 kub.sm

Ushbu masalani yechish uchun quyidagi bosqichlardan o'tish kerak.

- Matematik formula.
- Algoritm tuzish.
- Dastur tuzish.

$V=a*b*h$ formuladan foydalaniladi.



```

Program M_1;
Var
  V,a,b,h:real;
Begin
  Writeln('Parallelopipedning hajmini hisoblash. ');
  Write('Eni (sm): ');
  Readln(a); {kiritish operatori}
  Write('Bo''yi (sm): ');
  Readln(b);
  Write('Balandligi (sm): ');
  Readln(h);
  V:=a*b*h;
  Writeln('parallelopipedning hajmi:',V:3:2,' kub.sm');
  Readln;
end.
  
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Parallelopipedning hajmini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

(Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shrift bilan ko'rsatilgan).

Parallelopiped hajmini hisonblash.

Qiymatlarni kiriting:

Bo'yi (sm) -> **9**

Eni (sm) -> **7.5**
Balandligi (sm)->**5**
Hajmi:337.50 kub.sm.

2. Kubni hajmini hisoblash dasturi tuzilsin. (Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shrift bilan ko'rsatilgan).

Kubni hajmini hisoblash dasturi.

Kubni tomoni uzunligini (sm) kiriting va <Enter > ni bosing.

-> **9.5**

Kub hajmi: 857.38 kub.sm

3. Bir nechta daftar va qalamni sotib olingandagi harajatni hisoblash dasturi tuzilsin.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Harajatni hisoblash:

Daftar narxi (so'm)-> **100**

Daftarlar soni -> **5**

Qalam narxi (so'm)->**50**

Qalamlar soni -> **3**

Jami harajat: 650 so'm

Dasturni tugatish uchun <Enter> ni bosing.

4. Bir necha kilogrammdan iborat olmani narxini hisoblash dasturi tuzilsin.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Harajatni hisoblash dasturi.

Boshlang'ich qiymatlarni kiriting:

Bir kg olma narxi(so'm) -> **425**

Necha kg -> **2.5**

Harajat: 1062.5

5. Agar uchburchakning ikki tomoni va ular orasidagi burchak ma'lum bo'lsa, uchburchak yuzini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Uchburchak yuzini topish:

Bit satrda probel orqali uchburchakni ikki tomonini (sm) kiriting->**25 17**

Ular orasidagi burchak-> **30**

Uchburchak yuzini: 106.25 sm.kv.

6. Elektr zanjirdagi tok kuchni hisoblash dasturi tuzilsin.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Elektr zanjirdagi tok kuchni hisoblash:

Tok quvvati (volt) -> **36**

Qarshilik (OM) -> **1500**

Tok kuchi: 0.024 Amper.

7. Minutda berilgan qiymatlarni soatga aylantirish dasturi tuzilsin.
Quyida dasturni ishlash jarayoni ko'rsatilgan.
Minutda vaqtni kiriting -> **190**
190 minut bu – 3 soat 10 minut.

8. Yuguruvchni tezligini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Yuguruvchni tezligini topish.
Masofani kiriting (metr): **1000**
Vaqti (min.sekund): **3.25**
Yuguruvchni tezligini: 17.56 km/soat.

9. Uchburchakning asosi uzunligi va balandligiga ko'ra uchburchakni yuzasini topish dasturi tuzilsin

Quyida dasturni ishlash jarayoni ko'rsatilgan.
Boshlang'ich qiymatlarni kiriting:
Asosi (sm)-> **8.5**
Balandligi (sm)-> **10**
Uchburchak yuzi: 42.5 sm.kv.

10. a burchak radian o'lchovida berilgan, uni gradusda ifodalovchi dastur tuzing.

III. Amaliy ish himoyasi.

2- Amaliy ish.

MAVZU: SHARTLI VA SHARTSIZ O'TISH OPERATORLARI YORDAMIDA DASTURLAR TUZISH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Shartli o'tish operatori buyruqlar bajarilish ketma-ketligini shart asosida bajarilishini ta'minlaydi.

SHARTLI O'TISH OPERATORI ko'rinishi:
IF B THEN A1 ELSE A2; (To'liq ko'rinishi).

IF B THEN A1; (To'liqsiz ko'rinishi).

Bu yerda B - mantiqiy ifoda, A1 va A2 - ixtiyoriy ifoda. Bu operatorida B mantiqiy ifoda rost qiymat qabul qilganda A1 ixtiyoriy ifoda bajariladi, aks holda A2 ixtiyoriy ifoda bajariladi.

Shartsiz o'tish operatori ko'rinish: GOTO <belgi>. Bu operator yordamida boshqaruv ushbu belgi qo'yilgan satrga o'tadi. O'z navbatida belgini dasturning LABEL bo'limida e'lon qilinishi kerak.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala.

Ikkita sondan kattasini topish dasturi tuzilsin.

Program m_1;

Var

a,b:integer;

begin

writeln('Ikkita butun sonni kiriting: ');

Write('->');

Readln(a);

Write('->');

Readln(b);

If a>b then writeln(a,'>',b)

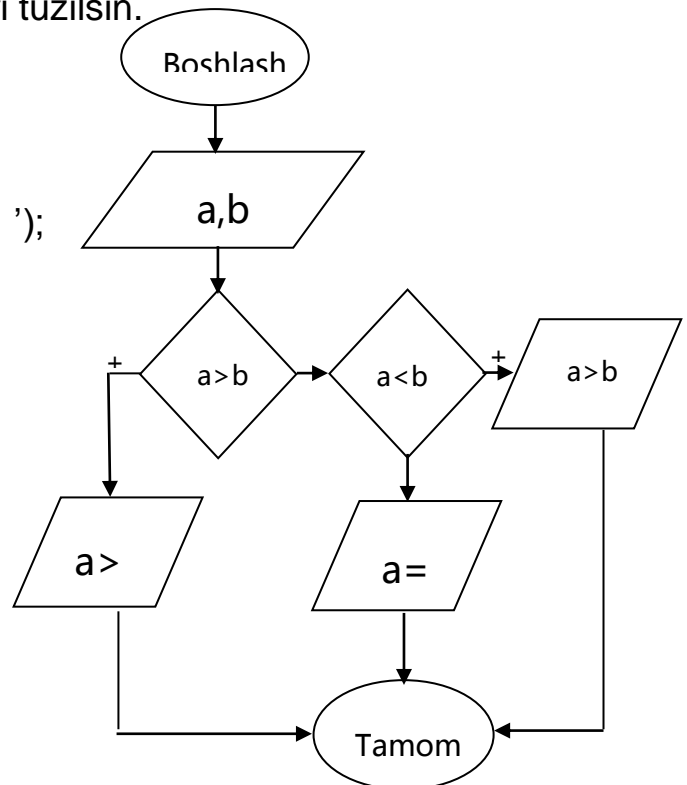
else

If a<b then writeln(a,'<',b)

else Writeln(a,'=',b) ;

Readln;

End.



2-masala.

$$y = \begin{cases} x^2, & \text{agar } x < 0 \\ -5, & \text{agar } x = 0 \\ \sqrt{x}, & \text{agar } x > 0 \end{cases}$$

Program M_2;

Label 1,2,3;

Var

x,y:real;

begin

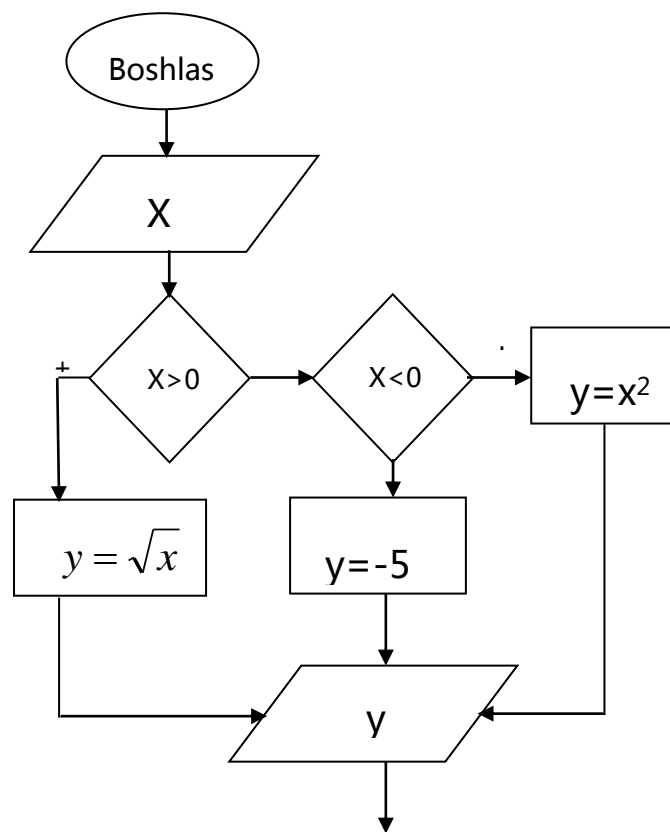
readln(x);

if x>0 then goto 1;

if x< then goto 2;

y:=-5; goto 3;

1: y:=sqr(x); goto 3;



```
2: y:=sqrt(x);
3: writeln('y=',y:4:2);
readln;
end.
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Elektr zanjirdagi ikkita ulangan qarshilikni hisoblash dasturi tuzilsin. Qarshiliklar parallel yoki ketma-ket ulangan bo'lishi mumkin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Elektr zanjirdagi qarshilikni hisoblash.
Boshlang'ich qiymatlarni kiriting:
Birinchi qarshilik (OM) -> **15**
Ikkichi qarshilik (OM)-> **27.3**
Ulanish turi (1-ketma-ket, 2-parallel)->**2**
Zanjirdagi qarshilik: 9.68 OM

2. Kvadrat tenglamani ildizini topish dasturi tuzilsin. Dastur kiritilgan koeffitsiyetlar orqali diskriminantni tekshirishi va kerakli ma'lumot chiqarishi kerak.

Kvadrat tenglamani ildizini topish.

Bir satrda tenglama koeffitsiyetlarini kiriting va <Enter> tugmasini bosing.

-> **1 -5 6**

Tenglama ildizlari:

X1=2, X2=3

3. Chegirma bilan harajatni hisoblash dasturi tuzilsin. Agar harajat 20000 so'mdan oshsa, 10% li chegirma taklif qilinadi. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Chegirma bilan harajatni hisoblash.

Harajatni (so'm) kiriting va <Enter> ni bosing.

-> **21000**

Sizga 10% li chegirma beriladi.

Chegirma bilan to'lanadigan summa: 18900 so'm.

5.O'quvchini bilimini tekshirish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Alisher Navoiy qaysi yili tug'ilgan?

Yilni kiriting va <Enter> ni bosing.

->**1401**

Sizning javobingiz noto'g'ri. Alisher Navoiy 1441 yilda tug'ilgan.

6. O'quvchini bilimini tekshirish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Qaysi satrda o'zgaruvchilarni e'lon qilish bo'limi to'g'ri ko'rsatilgan.

1. Const
2. Var
3. Type
4. Uses

To'g'ri javob nomerini kiriting va <Enter>ni bosing.

-> **2**

Javobingiz to'g'ri.

7. Foydalanuvchi tomonidan kiritilgan ikkita sonni solishtiruvchi dasturni tuzing.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Bir satrda ikkita sonni kiriting va <Enter > ni bosing:

-> **22 45**

22 45 dan kichik.

8. Foydalanuvchi tomonidan kiritilgan butun sonni 3 ga bo'linishini aniqlovchi dasturni tuzing. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Butun sonni kiriting va <Enter> ni bosing.

-> **67**

67 soni 3 ga bo'linmaydi.

9. Foydalanuvchi tomonidan kiritilgan butun sonni juft toqligini aniqlash dasturi tuzilsin

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Butun sonni kiriting va <Enter> ni bosing.

->**44**

44 soni juft son.

10. Foydalanuvchi tomonidan kiritilgan oynning tartib nomeriga mos yil faslini aniqlovchin dasturni tuzing. Agar foydalanuvchi noto'g'ri qimat kiritgan bo'lsa, „Kiritishdagi hatolik“ xabarini bersin.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Oynning tartib nomerini kiriting (1 dan 12 gacha)

-> 11

Qish.

3-AMALIY ISH.

MAVZU: VARIANT OPERATORI. (Case).

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Case operatorini qo'llay olishni o'rganish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Case operatori shartli o'tish operatori va shartsiz o'tish operatoridan farqli ravishda 3 va undan ortiq shartlar bilan ishlashda juda qulay operator hisoblanadi.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Hafta kuni tartib nomeriga mos holda „Ish kuni“ yoki „Dam olish kuni“ xabarini beruvchi dastur tuzing.

```
Case kun of
1..5:writeln('Ish kuni');
6,7:writeln('Dam olish kuni')
else
writeln('Kiritish hatosi');
end;
```

Ushbu dasturda kun identifikatori qabul qilgan qiymatga mos ravishda ekranga natija chiqadi. Ushbu holda kun faqatgina 1 dan 7 gacha sonlarni qabul qiladi, aks holda dastur "Kiritish hatosi" xabarini ekranga chiqaradi. Ushbu masalani to'liq dasturi matnini ko'rib chiqamiz.

```
Program hafta;
Var
kun:integer;
begin
write('Hafta kuni tartib nomerini kiriting:');
readln(kun);
Case kun of
1..5:writeln('Ish kuni. ');
6,7:writeln('Dam olish kuni. ');
else
writeln('Kiritish hatosi. ');
end; readln; end.
```

Ushbu dasturni ishlash jarayoni quyidagicha (foydalanuvchi tomonidan kiritilgan son qalin shrift bilan belgilangan):

```
Hafta kuni tartib nomerini kiriting: 6
Dam olish kuni.
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Xisoblang

$$y = \begin{cases} 0, \text{ agar } -2 < x < 2 & \text{bo'lsa} \\ 4, \text{ aks holda} \end{cases}$$

2. Xisoblang

$$y = \begin{cases} x^2 + 4x + 5, \text{ agar } x < 2 & \text{bo'lsa} \\ \frac{41}{x^2 + 4x + 5} & \text{aks holda.} \end{cases}$$

3. Xisoblang

$$y = \begin{cases} 0, \text{ agar } x < 0 & \text{bo'lsa} \\ x, \text{ agar } 0 < x < 1 & \text{bo'lsa} \\ x^4, \text{ aks holda.} \end{cases}$$

4. Xisoblang

$$y = \begin{cases} 0, \text{ agar } x < 0 & \text{bo'lsa} \\ x^2 - x, \text{ agar } 0 < x < 1 & \text{bo'lsa} \\ x^4 - \sin(x^2 - 1), \text{ aks holda.} \end{cases}$$

III. Amaliy ish himoyasi.

4- AMALIY ISH.

MAVZU: SHARTI OLDINDAN BERILUVCHI, SHARTI KEYIN BERILUVCHI, PARAMETRLI QAYTARILISH OPERATORLARIDAN FOYDALANIB DASTURLAR TUZISH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qaytarilish jarayonlarini dasturlashda uch xil operatorlarni qo'lashni o'rganish

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Sharti oldindan beriluchi qaytarilish operatori ko'rinish:

WHILE B DO S; Bu yerda B- mantiqi ifoda, S-ixtiyoriy ifoda.

Bu operatora: B mantiqiy ifoda yolg'on qiymat qabul qilgunga qadar S ixtiyoriy ifodani bajarilishi ta'minlanadi.

Sharti keyin beriluvchi qaytarilish operatori korinishi:

REPEAT S UNTIL B; Bu yerda B- mantiqi ifoda, S-ixtiyoriy ifoda.

Bu operaatorda: B-mantiqiy ifoda rost qitmat qabul qilgunga qadar S ixtiyoriy ifodani bajarilishi ta'minlanadi.

Parametrli qaytarilish operatori ko'rinishi:

1. For i:=n1 to n2 do

2. For i:=n2 downto n1 do

Be yerda: i – qaytarilish operatori, n1- parametrnning quyi chegarasi, n2- parametrnning yuqori chegarasi. $n1 < n2$ uhun bu operator o'rinli hisoblanadi.

Kerakli dasturiy va metodik qo'llanma: Kompyuter, Turbo-Pascal uhati, tarqatma materiallar.

Amaliy ishni bajarish tartibi:

I. O'quvchilar o'qitiuvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala.

1 dan 10 gacha bo'lgan toq sonlarni yig'indisini topish dasturi tuzilsin.

Ushbu masala uchun algoritm va dastur tuzamiz;

```
Program m_1;
```

```
Const n=10;
```

```
Var
```

```
    i,S:integer;
```

```
begin
```

```
    write('1 dan 10 gacha bo'lgan sonlarni yig'indisi:');
```

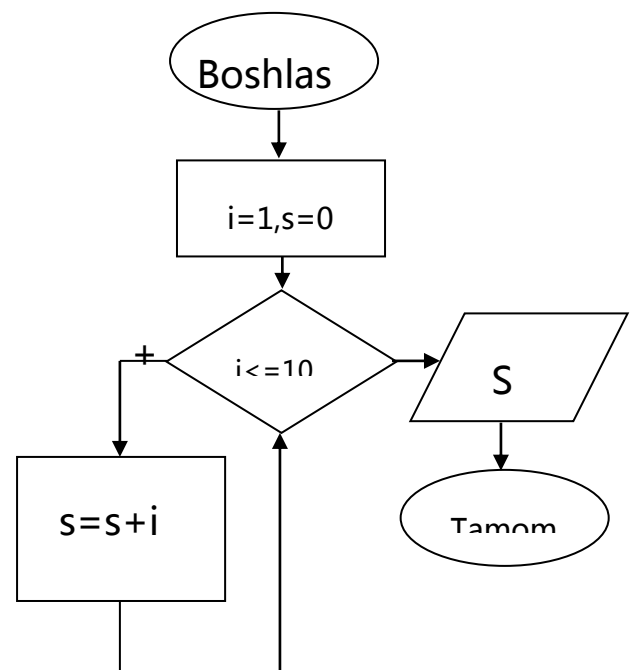
```
    S:=0;
```

```
    i:=1;
```

```

while i<=n do
begin
S:=S+i;
i:=i+2;
end;
writeln(S);
readln;
end.

```



II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. N sonigacha bo'lgan juft sonlarni yig'indisini topish dasturi tuzilsin.
2. M va N sonlarini EKUB ni toping.
3. Shahar aholisi soni yiliga 1/40 ga oshadi. Necha yilda shahar aholisining soni 3 marta oshadi.
4. $15x+20y+30z=270$ tenglamaning barcha x, y, z yechimlarini topish dasturini tuzing.
5. Natural sonlar berilgan. Bu sonlarining yig'idisi toq son bo'ladimi.
6. A va B ($A < B$) butun sonlar berilgan. Shu sonlar orasidagi barcha sonlarni kamayish tartibida chop etuvchi va ularni sonini aniqlash dasturini tuzing.
7. A va N ($N > 0$) butun sonlari berilgan.
 $1 + A + A^1 + A^2 + \dots + A^n$ ifodani hisoblash dasturini tuzing.
8. N ($N > 0$) butun son berilgan $2 + 1/(2!) + 1/(3!) + \dots + 1/(N!)$ ifodani hisoblash dasturini tuzing.
9. A va B ($A < B$) butun sonlar berilgan. Shu sonlar orasidagi barcha sonlarni (A va B sonlaridan tashqari) o'sish tartibida chop etuvchi va ularni sonini aniqlash dasturini tuzing.
10. 6 xonali sonlar uchun "Baxtli bilet" dasturini tuzing. Bunda avvalgi 3 xona sonlar yig'indisi keingi 3 xona sonlar yig'indisiga teng bo'lish kerak. Barcha "Baxtli bilet" lar sonini toping.
11. N sonining barcha bo'luvchilarini aniqlash dasturini tuzing.

III. Amaliy ish himoyasi.

5- AMALIY ISH.

MAVZU: BIR O'LCHOVLI MASSIV BILAN ISHLASH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, Massiv bilan ishlash ko'nikmalarini shakllantirish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Massiv o'zida bir xil tipdagi qiymatlarni saqlovchi kattalik xisoblanadi. Massiv bilan ishlash uchun quyidagilarni amalga oshirish kerak:

Massiv o'zgaruvchilar yoki ozgarmaslar bo'limida e'lon qilinishi mumkin:

Massivni kiritish, chiqarish, elementlari bilan ishlash uchun qaytarilish operatorlaridan foydalanish kerak bo'ladi. (while, repeat. for).

Agar dasturda massiv ishtirok etsa uni quyidagicha e'lon qilinadi:

```
var  
    massiv_nomi:array[boshlang'ich indeks..oxirgi indeks] of [qiymatlar  
turi];
```

Bu yerda massiv_nomi uchun pascal tili qonuniyati bo'yicha ixtiyoriy identifikator tanlanadi, Array-massivni e'lon qiluvchi so'z, Boshlang'ich va oxirgi indekslar – butun sonlar bo'lishi shart. Misol uchun yanvar oyidagi temperatura qiymatlarini o'zida saqlovchi massivni e'lon qilish uchun:

```
var  
    Yanvar:array[1..31] of real;
```

Bu holda biz bo'sh massiv e'lon qildik. Masalani hal qilish davomida bu massini to'ldirish, qiymatlari ustida amallar bajarish kerak bo'ladi, buning uchun massivning har bir elementiga murojaat qilish kerak bo'ladi. Quyida massivni to'ldirish ko'rsatilgan:

```
For i:=1 to 31 do  
Begin  
    Write(l,'-elementni kiriting:');  
    Readln(a[i]);  
End;
```

Amaliy ishni bajarish tartibi:

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala.

Foydalanuvchi tomonidan kiritilgan son massivda necha marta qatnashishini aniqlash dasturi tuzilsin. (Massivda biror sonni qidirish).

```
Program m_1;
Const
    N=7;
var
    a:array[1..n] of integer;
    s,q,i:integer;
begin
    writeln(n,' ta butun sonni kiriting: ');
    for i:=1 to n do
        begin
            write(i,'-son:');
            readln(a[i]);
        end;
    write('Qidirilayotgan sonni kiriting:');
    readln(q);
    s:=0;
    for i:=1 to n do
        if a[i]=q then s:=s+1;
        if s<>0 then writeln(q,' soni ', s, ' marta qatnashdi' )
        else writeln('Bunday element massivda mavjud emas');
    readln;
end.
```

II.O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Massivning o'rtacha arifmetigini aniqlash dasturi tuzilsin.
2. Massiv elementlarini o'sish tartibida saralash dasturini tuzing.
3. Massiv elementlari ichidan eng kattasini topish dasturi tusilsin.
4. Massiv elementlari o'sish tartibida joylashganini aniqlash dasturi tuzilsin.
5. N ta elementli massiv berilgan. Eng katta va eng kichik elementlar o'rnini almashtirish dasturini tuzing.
6. N ta elementdan iborat massiv elementlarini teskari tartibda chop etuvchiu dasturni tuzing.
7. N ta elementdan iborat massiv elementlarinining faqat toq sonlarini chop euvchi dastur tuzing.
8. a[1..10] massivni tasodifiy xolda 0 va 1 sonlari bilan to'ldiring va 0 ni 1 soniga, 1 ni 0 soniga almashtiruvchi dastur tuzing.
9. Shunday massiv tuzingki, unda dastlabki 100 ta 13 va 17 ga bo'linadigan sonlar joylashsin.

10. $xy+x+y=1000$ tenglamani barcha ildizlarini topish dasturini tuzing.
III. Amaliy ish himoyasi.

6-AMALIY ISH.

MAVZU: IKKI O'LCHOVLI MASSIV BILAN ISHLASH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Massiv elementlari ustida amallar bajarish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Ikki o'lchovli massiv bilan ishlash uchun ikkita o'lcham bilan ishlashga to'g'ri keladi, satr nomeri va ustun nomeri. Misol uchun beshinchi satr, yettinchi ustunda 8 soni oylasgan bo'lsa, «pascal» tilida quyidagicha ko'rinishga ega bo'ladi:

$A[5,7]:=8$

«pascal» tilida ikki o'lchovli massiv quyidagicha e'lon qilinadi:

Var

massiv_nomi:array

[Birinchi_satr_nomeri..

Oxirgi_satr_nomeri,

Birinchi_ustun_nomeri..

Oxirgi_ustun_nomeri] of massiv_elementlari_tipi;

Kompyuter o'z hotirasida 9 ta satr, 9 ta ustundan iborat jadvalda butun solarni saqlovchi bo'sh massivni e'lon qilish uchun:

Var a:array[1..9,1..9] of integer;

Amaliy ishni bajarish tartibi:

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala.

program M_1;

const

n=9;

var

a:array[1..9,1..9] of integer;

i,j:integer;

begin

for i:=1 to n do {Jadvalni to'ldirish}

```

for j:=1 to n do
a[i,j]:=i+j;
for i:=1 to n do
begin
  for j:=1 to n do
    write(a[i,j],'|'); {i-sartni ekranga chiqarish}
    writeln;          {Yangi satrdan boshlash}
end;
readln;
end.

```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Ikki o'lchovli massivning diagonal joylashgan elementlari yig'indisini topish dasturi tuzilsin.
2. Ikki o'lchovli massivning o'rta arimetigini hisoblash dasturi tuzilsin.
3. Ikki o'lchovli massivda elementlar yig'indisini topish.
4. Ikki o'lchovli massivning eng kichik elementini va joylashgan o'rnini topish dasturi tuzilsin.
5. $A(m,n)$ massivning elementlari yig'indisi eng katta va eng kichik bo'lgan satrlarini topuvchi dastur tuzing.
6. $A(5,5)$ massivning eng katta elementi indeksini topish dasturini tuzing.
7. $A(m,n)$ massivning 3 ga bo'linadigan sonlari yig'indisini topish dasturini tuzing.
8. Pifagor jadvalini chop etuvchi dastur tuzing.
9. M kvadrat massivning bosh diagonalida joylashgan elementlar o'rtacha arifmetigini topish dasturini tuzing.
10. 5×10 kvadrat massiv berilgan. Faqat musbat elementlardan tashkil topgan satrni chop etuvchi dasturini tuzing. Agar bunday satr bo'lmasa 0 ni chop etsin.

III. Amaliy ish himoyasi.

7-AMALIY ISH.

Mavzu: Qism dasturlar. Protseduralar.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qism dasturlar bilan ishlash ko'nikmasini shakllantirish

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY QISM: Dasturlash jarayonida murakkab dasturlarning bir necha joyida bir xil vazifani bajaruvchi operatorlar guruhini qo'llashga to'g'ri keladi va dasturda bir-biriga aynan o'xshash bir necha qism dasturlar vujudga keladi. Kompyuter otirasini va dasturtuzuvvchi vaqtini tejash maqsadida ushbu qismdagi o'xshash peratorlardan bir martrta asosiy dastur dasturdan ajratib yoziladi va unga asosiy dastur bajarilishi jarayonida murojaat qilinadi.

Dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatilishi mumkin bo'lan buday operatorlar guruhida qism dastur deb ataladi va u asosiy dastur bilan birr butunni tashkil qiladi.

Qism dasturlarni ishlatish dasturning hajmini kichraytiradi va uning ko'rinishini , o'qilishini hamda xatolar sonini kamaytiradi.

Protsedura quyidagicha e'lon qilinadi:

Procedure<nom>(<sohta parametrlar ro'yhati>);

Bu yerda PROCEDURE-xizmatchi so'z; <nom>-protseduraning nomi; <soxta paramtrlar ro'yxati>-turlari ko'rsatilgan erilganlarni va natijalarni ifodalaydigan nomlar turlari sanab o'tiladigan satr.

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quchilarga tushintirib beradi.

1-misol.

$y = x^n$ ni hisoblash dasturi protsedura orqali hal qilinsin.

```
program M_1;
```

```
var a,z:real;
```

```
    b:integer;
```

```
procedure daraja(n:integer;x:real;var y:real);
```

```
var i:integer;
```

```
begin
```

```
    y:=1;
```

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

```
        y:=y*x;
```

```
end; {daraja protsedurasiga tugadi}
```

```
begin
```

```
    writeln('sonni va darajasini kiriting:');
```

```
    write('->');
```

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

```
    if b=0 then z:=1
```

```
    else if b>0 then daraja(b,a,z) {daraja protsedurasiga murojaat qilindi}
```

```
    else daraja(-b,1/a,z);
```

```
    writeln(z:6:2);
```

```
    readln;
```

```
end.
```

2-misol.

```
program m_2;
```

```

var a,b,c,d:integer;
    mab,mcd,max:integer;
procedure max1(x,y:integer;var z:integer);
begin
if x>y then z:=x else z:=y;
end; {max1 protsedurasi tugadi}
begin
writeln('4 ta butun sonni kiriting: ');
readln(a,b,c,d);
max1(a,b,mab); {protseduraga murojaat}
max1(c,d,mcd);
max1(mab,mcd,max);
writeln('Eng kattasi:',max);
readln;
end.

```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. * dan tashkil topgan satrni chop etuvchi satrni dasturini tuzing. * lar soni protsedura parametri bo'lsin.
2. Parallelopipedni hajmini, asos yuzasini hisoblash protsedurasini tuzing. Protsedura parametrlari: bo'yi, eni, balandligi.
3. Agar M1, M2, M3 X(30), Y(15), Z(20) massivlarining maksimum elementlari bo'lsa, $K=(M1+M2+M3)/3$ ni hisoblash dasturini tuzing
4. To'rtburchakning burchaklari koordinatalari berilgan. To'rtburchak perimetri va yuzasini topish dasturini tuzing.
5. A(15), B(10), C(20) massivlari berilgan. Massiv elementlarini o'sish tartibida chop etuvchi dastur tuzing.
6. Agar m1 va m2 sonlar X(20) va X(35) massivlarning eng kichik elementlari bo'lsa, $Z=(m1+m2)/2$ ni hisoblash dasturini tuzing.
7. X(5,5) va Y(7,7) massivlarining bosh dioganallarida joylashdan elementlar yig'indisini topish dasturini tuzing.
8. A(m,n) massivdagi eng ko'p ketme-ket keluvchi bir xil qiymatli sonlar ketma-ketligini aniqlovchi dastur tuzing.
9. Faqat toq satrlaridan iborat bo'lgan massivn elementlarining:
 1. yig'indisini topish.
 2. O'rtacha arifmetigini topish.
 3. Eng kattasini topish.
 4. Eng kichigini topish dasturini tuzing.
10. Uchburcakni berilgan 3 ta tomoniga ko'ra perimetrini va yuzasini topish dasturi tuzilsin. Dastur foydalanuvchi tomonidan kiritilgan qiymatlarni tekshirishi kerak va 'Kiritish xatosi' a'lumotini chop etsin.

III. Amaliy ish himoyasi.

8-AMALIY ISH.

MAVZU: "PASCAL" TILIDA QISM DASTUR-FUNKSIYA.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qism dasturlar bilan ishlash ko'nikmasini shakllantirish.

TOPSHIRIQ: Talabalarning ma'ruza va amaliy mashg'ulotlarda olgan bilimlarini mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

NAZARIY MA'LUMOT: funksiya bu shunday qism dasturki, bunda bitta natija chiqadi va u funksiya qism dasturning nomida hosil bo'ladi. Shuning uchun funksiya qism dasturi protsedura qism dasturining xususiy holi bo'lib, undan farqi, birinchidan, funksiya qism dasturining natijasi faqat bitta qiymatdan iborat bo'ladi. Protседura qism dasturida natija esa bitta yoki bir nechta no'lishi mumkin. Funksiya qism dasturining natijasi asosiy dasturga funksiya natijasi sifatida beriladi. Protседura qism dasturi natijalari esa uning parametrlari qiymati sifatida hosil bo'ladi.

Funksiya qism dasturining ko'rinishi quyidagicha:

Function<nom>(<soxta parametrlar ro'yxati>):<tur> ;

Bu yerda function-xizmatchi so'z, <nom>-funksiya nomi, <soxta parametrlar ro'yxati>-kiritiladigan soxta parametrlar ro'yxati sana o'tiladi, <tur>-funksiya natijasi turi. Bunda natija funksiya nomida hosil bo'ladi. Bundan tashqari, funksiyaning parametrsiz ko'rinishini ham yozish mumkin;

Function <nom>:<tur>;

Har ikkala holda ham funksiya natijasi funksiya qism dasturi tanasida funksiya nomiga hech bo'laganda bir marta o'zlashtirilishi shart;

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quchilarga tushintirib beradi.

1-misol. Qaysi biri katta (Funksiya qism dasturi misolida).

```
program m_1;
```

```
var
```

```
  a,b,c,d:integer;
```

```
function max1(x,y:integer):integer;
```

```
begin
```

```
  if x>y then max1:=x else max1:=y;
```

```
end; {max1 funksiyasi tugadi}
```

```
begin
```

```
  writeln('4 ta butun sonni kiriting: ');
```

```
  readln(a,b,c,d);
```

```
writeln('Bu sonlardan eng kattasi:',max1(max1(a,b), max1(c,d)));
```

```
  readln; end.
```

2-misol. Ikkita registrdan tashkil topgan zanjirdagi qarshilikni hisoblash dasturi tuzilsin. Funksiya parametrlari bo'lib ulanish turi (ketma-ket yoki parallel) turi qabul qilinsin.

```
program zanjir;
uses crt;
var
  q1,q2:real;
  tur:integer;
function qq(r1,r2:real;t:integer):real;
begin
  if t=1 then qq:=r1+r2;
  if t=2 then qq:=r1*r2/(r1+r2);
end;
begin
clrscr;
write('Qarshiliklarni kiriting:');
readln(q1,q2);
write('Ketma-ket ulanish uchun 1/Parallel ulanish uchun 2:');
readln(tur);
writeln('Ushbu zanjidagi qarshilik:',qq(q1,q2,tur):6:2,' OM');
readln;
end.
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Silindrni hajmini hisoblash funksiyasi dasturini tuzing. Funksiya parametrlari bu_silindr asosi radiusi va balandligi.

2. Ikkita sonni solishtirib, mos xolda $>$, $<$, yoki $=$ korinishlarida natijani ko'rsatuvchi funksiya dasturini tuzing.

3. Ikkita qarshilikdan iborat zanjirdagi umumiy qarshilikni topish funksiyasi dasturini tuzing. Funksiya parametrlari bo'lib ulanish turi ko'rsatilsin.

4. a^b fuksiyani hisoblash funksiyasi dasturini tuzing.

5. Berilgan sonni N foizini hisoblash funksiyasi dasturini tuzing.

6. Bankka qo'yilgan puldan ma'lum vaqtdan keyin olinadigan foydani hisoblash funksiyasi dasturini tuzing. Fuksiya paramtrlari: Boshang'ich summa, yillik (oylik) foiz, muddati.

7. $n!$ ni hisolash fuksiyasi dasturini tuzing.

8. n natural son va $n > 2$. Quyagi yig'indini hisoblash dasturi tuzilsin

$$S=1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$$

9. Chziqli tenglamani ildizini topish dasturini tuzing. Funksiya parametrlari: a , b koeffitsiyentlar.

10. $A(n)$ masivni shuday hosil qilinki, $a(1)=1!$, $a(2)=2!$... $a(n)=n!$

III. Amaliy ish himoyasi.

9-AMALIY ISH.

MAVZU: SATRIY KATTALIKLAR.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish.

TOPSHIRIQ: berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoya qilish.

NAZARIY MA'LUMOT: Satrlar ASCII kodining belgilarining apostroflar (') ichiga olingan ketma-ketligian iborat. Masalan 'mart', '123', va h.k. Satr uzunligi deb unda qatnashayotgan belgilar soniga aytiladi. Bunda chegaralovchi apostrof belgilar hisobga olinmaydi. Masalan, ' ' satrning uzunligi 1 ga teng, " uzunligi esa 0 ga teng. Satriy o'zgaruvchilar quyidagicha e'lon qilinadi:

Var

O'zgaruvchi_Nomi:string;

yoki

var

O'zgaruvchi_Nomi:string[m];

Bu yerda M – ushbu satrga ajratilgan bayt hisobi. Satrning har bir belgisi hotirada 1 bayt joy egallaydi. Agar satr uzunligi ko'rsatilmagan bo'lsa, ushbu satr 255 belgidan iborat bo'lish mumkin.

«pascal» tilida matnli ma'lumotlar bilan ishlash uchun bir satr funksiya va protseduralar ishlab chiqilgan:

Copy(S:string; start,len:integer):string;

Bu yerda

S satrning start belgisidan boshlab len usunlikdagi satrni belgilaydi (aniqlaydi).

1-masala.

Program m_1;

Var s,s1:string;

Begin

S:='algoritm';

S1:=copy(s,5,4);{s1=ritm}

Writeln(s1);

Readln;

End.

Natijada ekranda ritm so'zi chop etiladi

Delete (var s:string;start,len:integer);

S satrning start simvolidan boshlab len usunlikdagi satrini o'chiradi.

S:='algoritm';

Delete(s,1,4);

Natija Ritm

Delete(s,3,1); ritm satrini aniqlaydi.

```
Insert(subs:string;var s:string; start:integer) ;
S satrga start o'rindan boshlab subs satrini qo'shadi.
S:='alm';
Subs:='gorit';
Insert(subs,s,3);
Pos(subs,s:string);byte;
Subs satri s satriga tegishli ekanligini aniqlaydi.
```

Misol.

```
S:='algoritm';
Subs:='ritm';
X:=pos(subs,s);
If x<>0 then writeln(subs,' satri ',s,' satrining ',x,' o'rnidan boshlanadi');
```

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quvchilarga tushintirib beradi.

1-misol.

Berilgan so'zni Polindrom yoki Polindrom emasligini tekshruvchi dasturni tuzing.

```
program m_1;
uses crt;
var s,t:string;
    k,i:integer;
begin
clrscr;
write('Tekshirilayotgan so'zni kiriting:');
readln(s);
k:=length(s);
t:="";
for i:=1 to k do
t:=copy(s,i,1)+t;
if s=t then writeln('Palindrom') else writeln('Palindrom emas');
readln;
end.
```

Ushbu masalada berilgan so'ni polindromligini topish uchun shu so'zni har bir harfini aniqlash uchun qaytarilish jarayoni ichida copy(s,i,1) funksiyasidan foydalaniladi va uni teskarida yig'ib bo'rish uchun copy(s,i,1)+t amali bajariladi.

Buni biz katak so'zi misolida ko'rib chiqamiz:

Qaytarilish jarayoni bajarilganda copy(s,i,1)+t buyrug'i

k

ak

tak

atak

katak

so'zni hosil qiladi va if s=t then writeln('Palindrom') buyrug'i bajarilgandan so'ng ekranda Palindrom javobi paydo bo'ladi.

II .O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Biror so'zda qatnashuvchi belgilar sonini aniqlash va ustun ko'rinishida chop etuvch dastur tuzing

2. Berilgan literli kattalikdagi "a" va "b" harflar soni yig'indisini hisoblovchi dastur tuzing.

3. Berilgan literli kattalikdagi hamma "a" harflarini "b" harfga almashtiruvchi dastur tuzing..

4. Berilgan matndagi har bir harfni ikkilantiruvchi dastur tuzing

5. Qirqib olish va birlashtirish yordamida "sobit" so'zining qismlaridan "obit", "bit", "it" so'zlarini tuzing.

6. NS so'zi KS so'zida necha marta uchrashini aniqlash dasturi tuzilsin.

7. "bol" so'zi berilgan so'zning qismligini aniqlash dasturi tuzilsin. Javob "ha" yoki "yo'q" bo'lishi kerak. Misol uchun "futbol" va "bolalar" so'zlari uchun "ha", "kitob" so'zi uchun "yo'q".

8. Berilgan gapda uchraydigan xar bir so'zning «dan» qo'shimchasini «ning» qo'shimchasi bilan almashtiring.

9. Satrda uchraydigan barcha '+' belgilari sonini aniqlash va '-' belgisiga o'zgartirish dasturini tuzing.

10. Berilgan gap nechta so'zdan iborat ekanligini topish dasturini tuzing.

III. Amaliy ish himoyasi.

10-AMALIY ISH.

MAVZU: Fayllar bilan ishlash

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish.

TOPSHIRIQ: berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoya qilish.

NAZARIY MA'LUMOT: TURBO PASCAL tilida fayllar bilan ishlash uchun bir qancha funksiya va protseduralar mavjud:Assign, Reset, Rewrite, Close, Rename, Erase, Eof,

Assign(var f; FileName: String) - protsedurasi f o'zgaruvchini fayl nomi FileName bilan bog'laydi. Dasturda faylga murojaat f o'zgaruvchi orqali amalga oshiriladi.

Reset(var f) – protsedurasi f fayldan ma'lumotlarni o'qish uchun ishlatiladi.

Rewrite(var f) – protsedurasi f faylga ma'lumot yozish uchun ishlatiladi.

Close(var f) - protsedurasi ochilgan faylni yopish uchun ishlatiladi. Close protsedurasi fayl bilan ishlashni tugatish uchun zarur hisolanadi. Agar biror sabab bilan Close protsedurasi ishlatilmasi faylga kiritilgan oxirgi o'zgartirishlar kiritilmaydi.

EOF(var f) – mantiqiy funksiyasi faylni oxirigacha o'qilganda TRUE qiymatni beradi.

Rename(var f; NewName: String) – protsedurasi diskdagi f o'zgaruvchi bilan bog'langan faylni qayta nomlash uchun ishlatiladi. Faylni qayta nomlash uchun fayl yopiq bo'lishi kerak.

Erase(var f) - protsedurasi diskdagi f o'zgaruvchi bilan bog'langan faylni o'chiradi. Faylni o'chirish uchun fayl yopiq bo'lishi kerak.

1- masala. Matli fayl tashkil eting va ma'lumot kiriting.

```
Program M_1;
Var F : text;
    s : string;
begin
Assign(f,'maqol.txt');
Rewrite(f);
Writeln('Matnni kiriting');
Readln(s);
While s<>' ' do
begin
    writeln(f, s);
    readln(s);
end;
close(f);
end.
```

Ushbu dasturda avval Assign(f,'C:\maqol.txt'); - buyrug'i C: diskda maqol.txt matnli fayli tashkil etildi va f o'zgaruvchiga bog'landi. Rewrite(f); operatori faylni yozish uchun tayyorlaydi. Klaviaturadan kiritilgan har bir belgini s satr qabul qiladi va f o'zgaruvchi orqali tashqi faylga yoziladi. Bu jarayon bo'sh satr kiritilguncha davom etadi. close(f); - buyrug'i orqali tashqi fayl yopiladi.

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quvchilarga tushintirib beradi.

1-misol. Matnli faylni o'qish.

```
Program M_2;
Var
F : text;
s: string;
begin
```

```

Assign(f, 'C:\maqol.txt');
Reset(f); {faylni o'qish uchun tayyorlash}
While not EOF(f) do {faylni o'qish, faylni oxirigi belgisigacha davom
etadi}
begin
    Readln(f,s) ; {f fayldan s satrga o'qish}
    Writeln(s); {o'qilganlar satrlarni ekranga chiqarish}
end;
close(f); {faylni yopish}
readln;
end.

```

II .O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Fayldagi sonlarni qo'shish.
2. Abituriyentlar haqida quyidagi ma'lumotlarni saqlovchi fayl tashkil eting:
 - Familiyasi, ismi va sharifi;
 - Tug'ilgan sanasi;
 - Shahidatnomasidagi o'rtacha bali;
 - Kirish imtihonidagi bahosi;
3. Fayldagi ma'lumotlarni boshqa faylga teskari (oxirgi belgidan birinchi belgiga qarab) yozuvchi dastur tuzing.
4. $y=x^3-5x^2+1$ funksiyaning $(-2 ; 5)$ oraliqda 0.5 qadam bilan hisoblash va har bir x va y ning qiymatlarini jadval ko'rinishida faylga yozuvchi dastur tuzing.

5.

$$y = \begin{cases} x^5 - \cos x, & \text{agar } x < 0 \\ \sin|x - a| + \operatorname{tg} x, & \text{agar } x = 0 \\ x^5 - a, & \text{agar } x > 0 \end{cases}$$

X argumentning qiymatlari myfile.dat fayli ichiga yozilgan. Myfile.dat faylining ichidan qiymatlarni olib Y funksiya qiymatlarini toping.

6.2. Laboratoriya mashg'ulotlari.

1-LABORATORIYA ISHI.

MAVZU: TURBO PASCAL DASTURLASH TILINING ASOSIY OPERATORLARI YORDAMIDA CHIZIQLI DASTURLAR TUZISH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

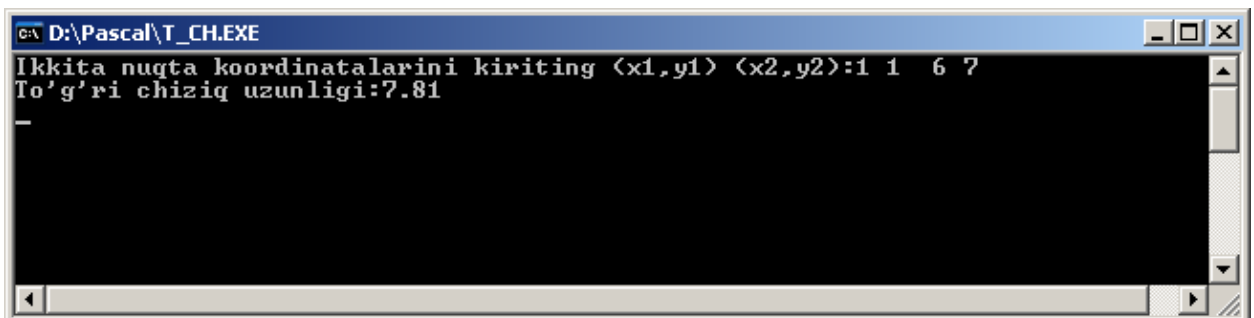
TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

MASHG'ULOTNI O'TKAZISH TARTIBI.

I.O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. (X_1, Y_1) va (X_2, Y_2) koordinatalarga ega bo'lgan ikkita nuqta orasidagi masofani toping.

```
program m_1;  
var  
  x1,x2,y1,y2:integer;  
  a:real;  
begin  
  write('Ikkita nuqta koordinatalarini kiriting (x1,y1) (x2,y2):');  
  readln(x1,y1,x2,y2);  
  a:=sqrt(sqr(x1-x2)+sqr(y1-y2));  
  writeln('To'g'ri chiziq uzunligi:',a:3:2);  
  readln;  
end.
```



```
c:\D:\Pascal\T_CH.EXE  
Ikkita nuqta koordinatalarini kiriting (x1,y1) (x2,y2):1 1 6 7  
To'g'ri chiziq uzunligi:7.81
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. To'g'ri burchakli to'rtburchakni perimetrini va yuzasini topish dasturini tuzing.
2. Ikkita son berilgan. Bu sonlarning kublarini o'rta arifmetigini va bu sonlarning modullari o'rta geometrigini toping.

3. Ikkita chiziqning kesishishidan hosil bo'lgan burcaklardan biri ikkinchisidan 50° ga katta bo'lsa, shu burcaklarni toping.
4. a, b -sonlar berilgan. Ikkita tomoni a va b ga, ular orasidagi burcak esa α ga teng bo'lgan uchburcak yuzasini toping.
5. a, b, c -haqiqiy sonlar berilgan. Bu sonlarning o'rta arifmetik qiymati, summasi va ko'paytmasini chop etuvchi dastur tuzing.
6. To'g'riburcakli uchburcakning katetlari berilgan bo'lsa, gipotenuzasi va yuzasini xisoblovchi dastur tuzing.
7. Uchburcak yuzasini uning tomonlari berilgan bo'lsa, Geron formulasi bo'yicha yuzasini xisoblovchi dastur tuzing.
8. N -balandlikdan erkin tushish vaqtini aniqlovchi dastur tuzing
9. a, b, c –parallelopiped tomonlari bo'lsa, uning xajmini aniqlang.
10. a, b kattaliklarning qiymatlari berilgan. r -kattalikdan foydalanib ularning qiymatlarini almashtiring.

2 - LABORATORIYA ISHI.

MAVZU: SHARTLI VA SHARTSIZ O'TISH OPERATORLARI YORDAMIDA DASTURLAR TUZISH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

MASHG'ULOTNI O'TKAZISH TARTIBI.

- I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Chiziqli tenglamani yechish dasturini tuzing.

```
program m_2;
```

```
label 1;
```

```
var
```

```
a,b:real;
```

```
begin
```

```
  1:write('Tenglama koeffitsiyentlarini kiriting:');
```

```
  readln(a,b);
```

```
  if (a<>0) and (b<>0) then
```

```
    writeln('x=',-b/a:3:2)
```

```
  else
```

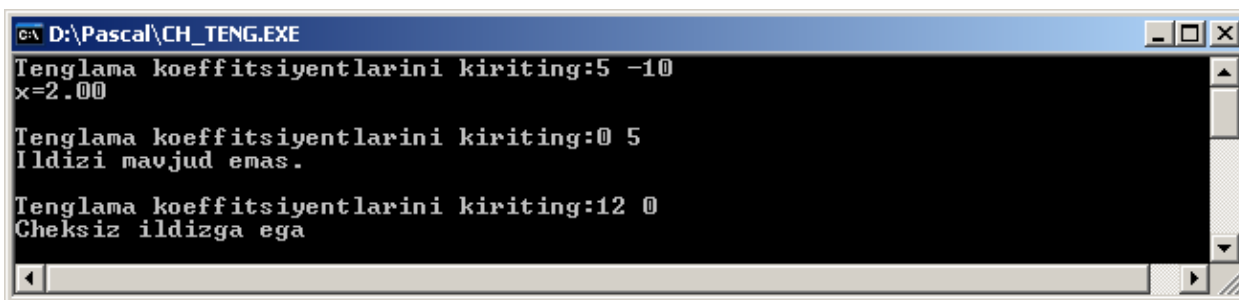
```
    if b=0 then writeln('Cheksiz ildizga ega')
```

```
  else
```

```
    writeln('ildizi mavjud emas.');
```

```
  goto 1;
```

readln; end.



```
C:\Pascal\CH_TENG.EXE
Tenglama koefitsiyentlarini kiriting:5 -10
x=2.00

Tenglama koefitsiyentlarini kiriting:0 5
Ildizi mavjud emas.

Tenglama koefitsiyentlarini kiriting:12 0
Cheksiz ildizga ega
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Tomoni berilgan kvadratga berilgan radiusli aylana joylashadimi?
2. Tomoni berilgan radiusli aylanaga kvadrat joylashadimi?
3. Berilgan yil necha kundan iborat ekanligini topuvchi dastur tuzing. (kabisa yili bo'lishi mumkin)
4. Uchta haqiqiy son berilgan (1.3) oraliqqa tegishlisini toping
5. Ikkita butun sonni bo'lish amalini bajaruvchi dastur tuzing. Dastur foydalanuvchi tomonidan kiritiluvchi qiymatlarni tekshirishi, agar xato kiritilgan bo'lsa, (bo'luvchi 0 ga teng) xatoni ko'rsatishi kerak. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Bir satrda bo'linuvchi va bo'luvchini kiriting va <Enter> ni bosing.

12 0

Kiritish xatosi. Bo'luvchi 0 ga teng bo'lishi mumkin emas.

6. Kvadrat tenglamani ildizini topish dasturini tuzing.

$$7. y = \begin{cases} x^2 - a, & \text{agar } x < 0 \\ \sin x + \cos x, & \text{agar } x = 0 \\ x^3 - a, & \text{agar } x > 0 \end{cases}$$

$$8. y = \begin{cases} x^5 - \cos x, & \text{agar } x < 0 \\ \sin|x - a| + \operatorname{tg} x, & \text{agar } x = 0 \\ x^5 - a, & \text{agar } x > 0 \end{cases}$$

9. Uchta son berilgan. Bu sonlar ichida manfiy qiymatga ega bo'lmagan sonning raqamlarini kvadratga ko'tarib, yig'indisidan yangi son xosil qiling. Musbat sonlarni o'zgartirishsiz qoldiring.

10. a, b va c ixtiyoriy sonlar berilgan. Agar shunday uzunlikdagi tomonlardan iborat uchburchak yasash mumkin bo'lmasa «0» ni, aks holda «3» ni, agar uchburchak teng tomonli bo'lsa «2»ni, agarda teng yonli bo'lsa 1 ni chop eting.


```

1:writeln('100 balli natija');
2:writeln('80 balli natija');
3:writeln('60 balli natija');
4:writeln('40 balli natija');
5:writeln('20 balli natija');
end;
readln;
end.

```

```

D:\Pascal\O'YIN.EXE
***Kompyuter o'ylagan sonni toping <1 dan 10 gacha>!!!!***
1-urinish:2
2-urinish:6
3-urinish:8
Barakalla, Topdingiz!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
60 balli natija

```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ ketma-ketlikning N-hadini topish dasturini tuzing.

2. N sonigacha bo'lgan toq sonlarni yig'indisini topish dasturi tuzilsin.

3. N sonini faktorilalini hisoblash dasturi tuzilsin.

4. 2 ning darajalari jadvalini chop etuvchi dastur tuzilsin.

5. Natural sonning bo'luvchilarini aniqlash dasturi tuzilsin.

6. $y = x^2 - 3x + 4$ ifoda uchun $-2 \leq x \leq 2$ oraliqda 0,5 qadam bilan hisoblash jadvalini ekranga chiqarish dasturi tuzilsin.

7. Klaviaturadan ketma-ket kiritilgan musbat sonlar ichidan eng kattasini topish dasturini tuzing. Quyida dasturni ishlash jarayoni ko'ratilgan:

Eng katta qiymatni topish dasturi. Kiritishni to'rtatish uchun 0 ni kiriting.

→ 56

→ 75

→ 43

→ 0

Eng katta son: 75.

8. 1 dan 10 gacha bo'lgan 10 ta tasodifiy sonlarni o'rtacha arifmetigini topish dasturini tizing.

Quyida dasturni ishlash jarayoni ko'rsatilgan:

*** Tasodifiy sonlar ***

1 3 4 2 7 4 9 6 2 1 O'rtacha arifmatigi: 3.9

9. 1 dan N gacha bo'lgan sonlar ichidan toqlarini va ular yig'indisini xisoblash dasturini tuzing.

10. $1+1/2+1/3+1/4 + \dots$ satrning birinchi N ta hadi yig'indisini topish dasturini tuzing. N soni dastur ishlash jarayonida kiritiladi.

Quyida dasturni ishlash jarayoni korsatilgan:

Yig'indini xisoblash dasturi:

Xadlar sonini kiriting: 15

Birinchi 15 ta xadlari yig'indisi: 3.3182

4-LABORATORIYA ISH.

MAVZU: BIR O'LCHOVLI MASSIV BILAN ISHLASH.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Massiv elementlari ustida amallar bajarish.

MASHG'ULOTNI O'TKAZISH TARTIBI.

I. O'quvchilar o'qitiuvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. 10 ta elementdan tashkil topgan massivning 5-o'rniga yangi element joylashtirish kerak. Bu xolda massivda berilgan elementlar ketma-ketligi buzilmasligi kerak. Avvalgi 5-element yangi element qo'yilgandan so'ng oltinchi o'ringa, 6-element yettinchi o'ringa v.x.zo o'tishi kerak. Oxirgi element esa yo'qotiladi.

```
program Beshinchi_element;  
const  
  n=10;  
  a:array[1..n] of integer=(11,12,13,14,16,17,18,19,20,21);  
var  
  i:integer;  
begin
```

```

writeln('Avvagi xolat');
for i:=1 to n do
write(a[i], ' ');
  for i:=10 downto 6 do
    a[i]:=a[i-1];
    a[5]:=15;
  writeln;
writeln('5-element qo\'shilgan xolat');
for i:=1 to n do
write(a[i], ' ');
readln; end.

```

The screenshot shows a window titled 'D:\Pascal\5-ELEMEN.EXE'. The output displayed is:

```

Avvagi xolat
11 12 13 14 16 17 18 19 20 21
5-element qo'shilgan xolat
11 12 13 14 15 16 17 18 19 20 _

```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Oyli o'rtacha xaroratni xisoblash dasturini tuzing.
2. Massivdagi bir xil qiymarli sonlarni sonini aniqlovchi dastur tuzing.
3. Massivda qidirilayotgan sonni o'rnini topish dasturini tuzing.
4. Sinf o'quvchilarini algebra fanidan olgan baholarini tahlil qiluvchi dastirni tuzing.
5. Massiv elementlari ichidan eng kattasini topish dasturini tuzing.
6. a_1, a_2, \dots, a_n butun sonlar ketma-ketligi berilgan. Mazkur ketma-ketlikda bir martadan keluvchi butun sonlar sonini aniqlang.
7. Komponentalari butun sonlardan iborat a_1, a_2, \dots, a_n massiv berilgan. Massivning xar bir elementini quyidagicha o'zgartiring: musbat elementlarini 1 bilan, manfiylarini -1 bilan almashtiring, 0 larni esa o'zgartirishsiz qoldiring.
8. $N_1, N_2, N_3, \dots,$ va N_n sonlar ketma-ketligi berilgan bo'lsin. Bu ketma-ketlikga tegishli bo'lgan berilgan R musbat sondan katta sonlar sonini xisoblang.
9. a_1, a_2, \dots, a_n butun va n natural son berilgan. a_1, a_2, \dots, a_n ketma-ketlikning 5 ga bo'linadigan va 7 ga bo'linmaydigan xadlari sonini va ularning yig'indisini toping.
10. Tasodifiy sonlar bilan A [1..10] massivni to'ldirish va o'rtacha arifmetigini topish dasturini tizing.

5-LABORATORIYA ISH.
MAVZU: IKKI O'LCHOVLI MASSIV BILAN ISHLASH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Massiv elementlari ustida amallar bajarish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

MASHG'ULOTNI O'TKAZISH TARTIBI.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Birinchi 25 ta toq va 3 ga bo'linmaydigan sonlarni 5*5 jadvalda chop etuvchi dastur tuzing.

```
program m_1;
uses crt;
label 1;
const n=25;
var
  a:array[1..n] of integer;
  i,j,k:integer;
begin
  clrscr;
  j:=0;
  for i:=1 to n do
  begin
    1:j:=j+1;
    if (j mod 2<>0) and (j mod 3<>0)
    then a[i]:=j
    else goto 1;
  end;
  writeln;
  writeln;
  writeln('*** Xosil qilingan jadval ***');
  for i:=1 to n do
  begin
    write(a[i]:3, ' ');
    if i mod 5=0 then writeln;
  end;
  writeln('Chiqish uchun <Enter> ni bosing');
  readln;
```

end.

```
*** Xosil qilingan jadval ***
 1   5   7  11  13
17  19  23  25  29
31  35  37  41  43
47  49  53  55  59
61  65  67  71  73
Chiqish uchun <Enter> ni bosing
_
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Birinchi 10 ta sonni kubini 2×5 jadvalda chop etuvchi dasturni tuzing.
2. Berilgan matritsaning barcha satr elementlarining maksimumni toping.
3. Berilgan matritsaning barcha satr elementlarining minimumini toping.
4. Berilgan butun sonli matritsa uchun matritsaning bosh diagonaliga parallel bo'lgan diagonal elementlarining eng kattalari yig'indisini toping.
5. Berilgan kvadrat matritsaning yuqori chap burchakdan boshlab, avval birinchi satr elementlarini o'ng tomon yo'nalishda, ikkinchi satr elementlarini oxiridan boshlab chap tomon yo'nalishida chop eting. Ushbu jaraenni oxirgi satrgacha davom ettiring.
6. Berilgan butun sonli matritsaning faqat toq elementlaridan iborat satrlari orasidan elementlari yig'indisining moduli maksimal bo'lgan satrini toping.
7. Quyida ko'rsatilgan jadvalda * bilan belgilangan qiymatlarni yig'indisini topish dasturini tuzing:

8. Ikki o'lchovli massivning eng kichik elementni va uning o'rnini topish dasturini tuzing.

9. 10×10 o'lchami massivni quyida ko'rsatilgandek to'ldirish dasturini tuzing.

				0

10. Bir xonali sonlarni qo'shish jadvalini chop etuvchi dastur tuzing.

6-LABORATORIYA ISH.

MAVZU: MAVZU: QISM DASTURLAR. PROTSEDURALAR.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

MASHG'ULOTNI O'TKAZISH TARTIBI.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala.

Quyidagi imkoniyatga ega dastur tuzing. 1-aylana uzunligini, 2-doira yuzasini, 3-shar xajmini, 4-dasturdan chiqishni ta'minlovchi dastur tuzing.

```
program m_1;
uses crt;
label 1,2,3;
var
  u,r:integer;
```

```

procedure qq(r:integer; var f:integer);
const
  pi=3.14;
begin
  if f=1 then writeln('Aylana uzunligi:', 2*pi*r:4:2);
  if f=2 then writeln('Doira yuzasi:',pi*sqr(r):4:2);
  if f=3 then writeln('Shar hajmi:',4/3*sqr(r)*pi:4:2) else
  writeln('Dasturdan chiqish uchun <ENTER> ni bosing.');
```

```

end;
begin
  clrscr;
  write('Aylana radiusini kiriting:');
  readln(r);
  writeln('1-Aylana uzunligi');
  writeln('2-Doira yuzasi');
  writeln('3-Shar hajmi');
  writeln('4-Dastur ishini tugatish');
  write('Mos usulni kiriting(1-4):');
  readln(u);
  qq(r,u);
  readln;
end.

```



II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. a,b,c xaqiqiy sonlar berilgan,bo'lsa xisoblang

$$\frac{\max(a, a + b) + \max(a, b + c)}{1 + \max(a + bc)}$$

2 x,y,z berilgan. Hisoblang:

a) $\max(x+y+z, xyz)+3$

b) $\min(x^2 + y^2, y^2 + z^2)-4.$

3. Radian qiymatli burchakni gradus qiymatga o'tkazuvchi protsedura tuzing va undan dasturda foydalaning

4. 10-elementli massiv berilgan. Uning musbat elementlarini 4-ga ko'paytiruvchi dastur tuzing

5. $1!+2!+\dots+15!$ –ni xisoblash dasturini tuzing.
6. To'g'ri chiziq koordinatalariga ko'ra to'g'ri chiziq uzunligini topish dasturini tuzing.
7. Berilgan uchburchak uchlari koordinatalariga ko'ra, teng tomonli uchburchakligini aniqlash dasturi tuzilsin.
8. Berilgan ikkita sonning EKUB va EKUK larini topish dasturini tuzing.
9. $y=kx+b$ tenglama bilan berilgan to'g'ri chiziqda M nuqtalari yotadimi?
10. Berilgan soat, minut va sekundni sekundda aniqlovchi dastur tuzing.

7-LABORATORIYA ISH.

MAVZU: MAVZU: "PASCAL" TILIDA QISM DASTUR-FUNKSIYA

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

MASHG'ULOTNI O'TKAZISH TARTIBI.

II. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

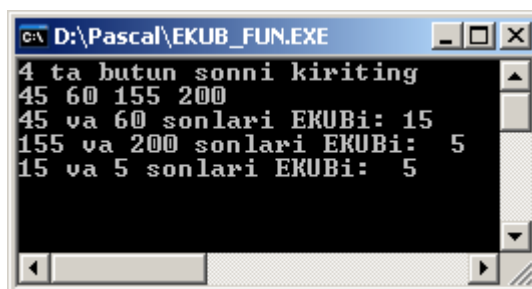
1-masala. Berilgan 4 ta natural sonning eng kichik umumiy bo'luvchi EKUB ini toping.

```

program m_1;
function ee(a,b:integer; var z:integer):integer;
begin
  while a<>b do
    if a>b then a:=a-b
    else
      b:=b-a;
      z:=a;
end;
var a,b,c,d:integer;
    m,n,k:integer;
begin
  writeln('4 ta butun sonni kiriting');
  readln(a,b,c,d);
  ee(a,b,m);
  ee(c,d,n);
  ee(m,n,k);
  writeln(a,' va ',b,' sonlari EKUBi:',m:3);

```

```
writeln(c,' va ',d,' sonlari EKUBi:',n:3);  
writeln(m,' va ',n,' sonlari EKUBi:',k:3);  
readln;  
end.
```



```
D:\Pascal\EKUB_FUN.EXE  
4 ta butun sonni kiriting  
45 60 155 200  
45 va 60 sonlari EKUBi: 15  
155 va 200 sonlari EKUBi: 5  
15 va 5 sonlari EKUBi: 5
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Kvadrat tenglamani ildizini topish dasturini tuzing.
2. 4 xonali son berilgan. Div va mod amallaridan foydalanib, xar bir xonadagi sonlar yig'indisini topish dasturini tuzing.
3. 4 xonali sonni palindromga tekshiruvchi dastur tuzing.
4. Berilgan sonni radiandan o'lchovdan gradus o'lchovga o'tkasuvchi dastur tuzing.
5. Sonlar ustida quyidagi amallarni bajaruvchi dasturni tuzing. 1- qo'shish, 2- ayirish, 3- ko'paytirish, 4- bo'lish.
6. Haqiqiy a,b va c sonlari berilgan. $S=\max(a-b,a,a+b)+\max(a,b+c,a-c)$ ifodani qiymatini topish dasturini tuzing.
7. Agar $a_1=1!$, $a_2=2!$, ... , $a_n=n!$ bo'lsa, A(n) massivni hosil qilish dasturini tuzing.
8. X(5,5) va Y(7,7) kvadrat massiv berilgan. Bosh dioganalarda joylashgan elementlar yig'indisini topish dasturini tuzing.
9. Quyidagi fikrni tekshiruvchi dastur tuzing: "Berilgan 3 xonali sonning xonalarida joylashgan sonlar geometrik progressiyani tashkil etadi".
10. 3 ta A(15), B(10), C(20) massivlari berilgan. Har bir massiv yi'gindisini hisoblab, kamayish tartibida chop etuvchi dastur tuzing.

8-LABORATORIYA ISH.

MAVZU: SATRIY KATTALIKLAR.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish.

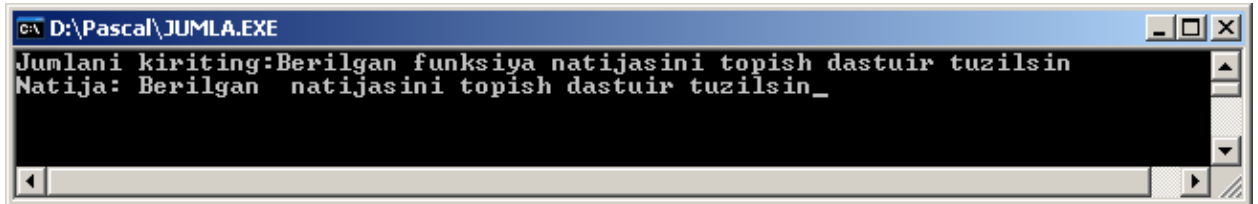
TOPSHIRIQ: berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoya qilish.

MASHG'ULOTNI O'TKAZISH TARTIBI.

- I. O'quvchilar o'qitiuvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Berilgan jumlada "funksiya" so'zini o'chruvchi dastur tizing.

```
uses crt;
var a : string;
    i,l : longint;
label m;
begin
  clrscr;
  write('Jumlani kiriting:');
  readln(a);
  l:=length(a);
  m : for i:=1 to l do
    if (a[i]='f')and(a[i+1]='u')and(a[i+2]='n')and
      (a[i+3]='k')and(a[i+4]='s')and(a[i+5]='i')and
      (a[i+6]='y') and (a[i+7]='a')then
      begin
        l:=l-8;delete(a,i,8);goto m;
      end;
  write('Natija: ',a);
  readln;
end.
```



II. O'quvchilar o'qitiuvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1. Berilgan so'zda istirok etgen "A" belgilari sonini aniqlash dasturini tuzing.
2. Berilgan jumlada nechta so'z ishtirok etayotganini aniqlovchi dastur tuzing.
3. Berilgan so'zni Polindromligini aniqlash dasturini tuzing.
4. Biror belgi va so'z berilgan. Ushbu belgi so'zda necha marta ishtirok etadi.
5. Xarflarni alfavitda joylashgan o'rni bo'yicha kodlash, berilgan kodga mos so'zni aniqlash dasturi tuzilsin.
6. Berilgan so'zni teskarilash dasturini tuzing.
7. S1 va S2 satrlar berilgan. S1 satrida S2 satrini mavjudligini tekshiruvchi dastur tuzing.
8. Berilgan satrdagi "a" harfini "o" harfiga almashtiruvchi dastur tuzing.

9. Berilgan satrdagi eng uzun va eng kichik so'zlarni aniqlash dasturini tuzing.

10. 10 lik sanoq sistemasida berilgan sonni 2 likka o'tkazish dasturini tuzing.

Adabiyotlar ro'yxati:

1. Информатика. Под. ред. Проф. Н. В. Макаровой.-М; 1997.
2. Немнюгин С. А. "Turbo Pascal. Программирование на языке высокого уровня" Учебник для вузов. 2-е изд СПб.: Питер–2006.-544
3. Немнюгин С. А. Turbo Pascal. Учебник. – М.,2002.
4. Axmedov A., Tayloqov N. Informatika. Akademik litsey va kasb-hunar kollejlari uchun darslik. – T.: O'zbekiston, 2001. - 272 b.
5. Епанешников А, Епанешников В "Программирование в среде Turbo Pascal 7.0" М.: "Диалог-МИФИ", 1993.-288 с
6. Фаронов В. В. "Turbo Pascal" — СПб.: ВХВ-Питербург, 2004. - 1056 с
7. Немнюгин С., Перколаб Л,. "Изучаем Turbo Pascal" — СПб.: Питер, 2001.
8. Немнюгин С. "Turbo Pascal практикум" — СПб.: Питер, 2003.