

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

**ABU RAYHON BERUNIY NOMIDAGI
TOSHKENT DAVLAT TEXNIKA UNIVERSITETI**

**Fakultet “Avtomatika va Boshqaruv”
“Boshqarishda axborot texnologiyalari” kafedrası**

Giyosov Tohir

**Mavzu:
Algoritmik tillarda programmalash strukturasi
(paskal misolida)**

**bakalavr darajasini olish uchun taqdim qilingan
Bitiruv malakaviy ish**

<i>Taqrizchilar:</i>	t.f.n. dots. Abdurahmanov B.A.,
	Prof. T.f.n Yusupov Firnafas
<i>Bajardi:</i>	Giyosov T

TOSHIKEHT – 2012

MUNDARIJA	
KIRISH	12
I. ILMIY-TEXNIK MASALALARNI KOMPYUTERDA ECHISH BOSQICHLARI. ALGORITMLASH BOSQICHI	
1. Masalalarni kompyuterda echish bosqichlari.....	13
2. Masalalarni kompyuterda echishning algoritmlash bosqichi.....	20
3. Algoritm tushunchasi va unga misollar.....	21
II. ALGORITMNING ASOSIY XOSSLARI, IFODALASH USULLARI, TURLARI	
1. Algoritmning asosiy xossalari.....	23
2. Algoritmni ifodalash usullari va misollar.....	24
3. Dasturlash tillari va ularni tasniflash.....	28
4. Algoritmning asosiy turlari	30
III. ALGORITMLASHTIRISH ASOSLARI. TURBO PASKAL ALGORITMIK TILI	
1. Asosiy konstruktsiyasi. Ma'lumotlarning sodda turi.....	34
2. Ma'lumotlar turlari.....	37
3. Ifodalar.....	39
IV. TURBO PASKAL ALGORITMIK TILI DASTURINING TARKIBIY QISMI	
1. Belgilar bo'limi.....	47
2. Toifalar bo'limi.....	48
3. O'zgaruvchilar bo'limi.....	48
V. TURBO PASKAL TILIDA ODDIY ALGORITMLARNI DASTURLASH ASOSLARI	
1. O'zlashtirish operatori.....	52
2. Ma'lumotlarni kiritish va chiqarish.....	53
VI. TARMOQLANUVCHI HISOBLASH JARAYONLARINI ALGORITMLASH VA DASTURLASH	
1. Shartsiz o'tish operatori.....	62

2. Shartli o'tish operatori.....	62
3. Tanlash operatori.....	64
VII. TAKRORLANUVCHI HISOBLASH JARAYONLARINI ALGORITMLASH VA DASTURLASH	
1. Takrorlanish jarayonlarini tashkil qilish.....	74
2. Sharti avval tekshiriladigan takrorlanish jarayoni.....	75
3. Sharti keyin tekshiriladigan takrorlanish jarayoni.....	78
4. Parametrli takrorlanish jarayoni.....	79
5. Murakkab takrorlanish jarayonlari.....	80
VIII. MA'LUMOTLARNING QAYD QILINGAN, CHEGARALANGAN VA MUNTAZAM TOIFALARI	
1. Ma'lumotlarning qayd qilingan toifasi.....	87
2. Ma'lumotlarning chegaralangan toifasi.....	89
3. Ma'lumotlarning murakkab toifasi. Muntazam toifa.....	90
IX. FUNKTSIYA VA PROTSEDURALAR TUSHUNCHASI	
1. Turbo Paskal tilida protsedura konstruksiyasi.....	99
2. Funktsiya qism dasturi haqida.....	100
3. Protседura va funktsiya qism dasturiga murojaat.....	102
4. Protседura va funktsiya qism dasturlarining parametrlari.....	102
X. MA'LUMOTLARNING ARALASH TOIFASI. YOZUVLAR	
1. Dasturda yozuvlarni tavsiflash.....	107
2. Yozuv elementlari va ular ustida bajariladigan amallar.....	109
3. Birlashtirish operatori.....	110
XI. TO`PLAMLAR. TO`PLAM USTIDA BAJARILADIGAN AMALLAR	
1. Turbo Paskal tilida to`plam tushunchasi.....	115
2. To`plamlarni tavsiflash.....	115
3. To`plam elementlari ustida bajariladigan amallar.....	116
XII. MA'LUMOTLARNING FAYLLI TOIFASI. TOIFALASHGAN VA TOIFALASHMAGAN FAYLLAR	

1. Turbo Paskal algoritmik tilida fayl tushunchasi.....	123
2. Toifalashgan fayllar va ular bilan ishlash.....	127
3. Toifalashmagan fayllar	131
XIII. MA'LUMOTLARNING MUROJAAT TURI. KO`RSATKICHLAR	
1. Ko`rsatkichlarni tavsiflash.....	137
2. Dinamik o`zgaruvchilarni ishlatish.....	142
3. Stek dinamik strukturasi.....	143
XIV. TURBO PASKAL MODULLARI. GRAPH MODULI	
1. Foydalanuvchi modullarini tashkil qilish va ishlatish.....	147
2. Turbo Paskal standart modullari.....	148
3. Graph modulining protseduralari.....	153
ADABIYOTLAR RO'YXATI.....	195

KIRISH

Xozirgi kunda ishlab chiqarishning turli texnologiyalaridagi muammolarni hal qilishda zamonaviy hisoblash texnikasi vositalarini keng qo'llash xar bir mutaxassisdan, xox u texnolog bo'lsin xox iqtisodchi bo'lsin albatta zamonaviy kompyuterlardan, hamda axborot texnologiyalaridan kerakli darajada foydalanish mahorati va tegishli ma'lumotga ega bo'lishni talab qiladi.

Informatikada masala echish tushunchasi deganda axborotlarni qayta ishlab, natijani oldindan belgilangan ma'lum bir ko'rinishga olib kelish tushuniladi.

Kompyuterdan foydalanib masalani echish - yaratilgan algoritmgaga asoslangan xolda dastlabki ma'lumotlar ustida avtomatik tarzda amallar bajarilib izlangan natija (natijalar) ko'rinishiga keltirish demakdir.

I. ILMIY-TEXNIK MASALALARNI KOMPYUTERDA ECHISH BOSQICHLARI. ALGORITMLASH BOSQICHI.

1. Masalalarni kompyuterda echish bosqichlari.
2. Masalalarni kompyuterda echishning algoritmlash bosqichi.
3. Algoritm tushunchasi va unga misollar.

1. Masalalarni kompyuterda echish bosqichlari.

Kompyuterdan foydalanib "ilmiy - texnik masalani echish" tushunchasi keng ma'nodagi so'z bo'lib, quyidagi bosqichlarga bo'linadi.

Ishdan maqsad – programmashtirish bosqichlarning qaysi birlarini mutaxassis kompyuterdan foydalanmasdan va qaysi bosqichlarini kompyuterdan foydalanib bajarishini aniqlash xamda bosqichlarni to'la o'rganib chiqishdan iborat.

Ilmiy - texnik masalalarni (ITM) kompyuterdan foydalanib echish bosqichlari:

1. Masalaning qo'yilishi va maqsadning aniqlanishi;
2. Masalani matematik ifodalash;
3. Masalani echish uslubini ishlab chiqish, sonli usullarni tanlash;
4. Masalani echish algoritmini ishlab chiqish;
5. Ma'lumotlarni tayyorlash va tarkibini aniqlash (tanlash);
6. Dasturlash;
7. Dastur matnini va ma'lumotlarni axborot tashuvchiga o'tkazish;
8. Dastur xatolarini tuzatish;
9. Dasturning avtomatik tarzda kompyuterda bajarilishi;
10. Olingan natijalarni izohlash, tahlil qilish va dasturdan foydalanish uchun ko'rsatma yozish;

Bu bosqichlar tom ma'noda to'laligicha mutaxassislikni egallash davomida maxsus kurslar vositasida o'rgatiladi.

8- va 9-bosqichlarni bajarishda mutaxassis kompyuterdan foydalanadi.

7-bosqichda kompyuterdan foydalanish ham, foydalanmaslik ham mumkin.

ITM ni kompyuterda echish bosqichlarini aloxida ko'rib chiqamiz.

1-bosqich. MASALANING QO'YILISHI VA MAQSADNI ANIQLASH.

Xalq xo'jaligining muayyan sohasi (texnika, iqtisod, lingvistika, ta'lim va x.k.) bo'yicha ishlayotgan (ishlagan) malakali va etakchi mutaxassis tomonidan bajariladigan ish, masalani qo'yish va maqsadni aniqlash uchun malakali mutaxassis bir necha kun, oy, xattoki yillab izlanishi mumkin.

Qo'yilgan maqsadni amalga oshirish uchun kerakli ma'lumotlar tarkibi (strukturasi), tuzilishi, ifodalanishi aniqlangan bo'lib, ular orasidagi bog'lanishlar aniq ifodalangan bo'lsa, "masala qo'yilgan" deb aytiladi.

2-bosqich. MASALANI MATEMATIK IFODALASH.

Bu bosqichda masalani echish uchun kerakli va etarli bo'lgan dastlabki ma'lumotlarning tarkibi, tavsifi, turi, tuzilishi xsobga olingan xolda matematik terminlarda ifodalanadi xamda masalani echishning matematik modeli yaratiladi.

Buning uchun xar xil (sohasiga qarab) matematik apparat ishlatilishi mumkin.

Masalan iqtisod soxasidagi mutaxassislar - chiziqli dasturlash, dinamik dasturlash, stoxastik dasturlash, bashorat (prognoz) qilish bilan bog'liq masalalarni echish matematik apparatini bilishlari kerak; texnika soxasidagi mutaxassislar oddiy differentsial tenglamalar va ularning tizimlari, mexanikaning chetki (kraevie) masalalarini, gaz dinamikasiga oid masalalarni, integral ko'rinishdagi masalalarni ifodalash va echish uchun ishlatiladigan matematik apparatni to'liq tushunib etgan bo'lishi kerak.

Mutaxassis o'z soxasini xar tomonlama yaxshi o'rgangan va amaliy jixatdan puxta o'zlashtirgan va qo'llaniladigan har xil matematik apparatning barcha imkoniyatlarini to'liq tushunib yetgan va amaliyotga qo'llay oladigan bo'lishi kerak.

Bu bosqichda 2 ta asosiy savolga javob topish kerak:

1. Masalani ifodalash uchun qanday matematik strukturalar maqsadga muvofiq keladi?

2. Echilgan o'xshash masalalar bormi?

Tanlangan matematik struktura (apparat)da masalaning elementida ob'ektlari to'la ifodalanishi zarur.

3-bosqich. MASALANI EChISH USULINI IShLAB ChIQISH, SONLI USULNI TANLASH.

Agar dastlabki ma'lumotlar bilan izlanayotgan natijalar (miqdorlar, ma'lumotlar) o'rtasida aniq bog'liqlik (qonuniyat) o'rnatilgan bo'lib va masalani echish uslubi ishlab chiqilgan bo'lsa yoki o'sha bog'lanishni amalga oshirish uchun tayyor sonli usul (lar) tanlab olinib (masala uchun, masalaning bir qismi uchun) masalaning echish uslubi yaratilgan bo'lsa, "masalani echish uslubi ishlab chiqilgan" deyiladi.

Bunda: X - dastlabki ma'lumotlar; Y - natija, maqsad funktsiyasi, izlanayotgan miqdor (lar) bo'lsa, ular orasidagi bog'lanish $Y = f(X)$ kabi olinishi mumkin.

f -dastlabki ma'lumotlar bilan natijani bog'lovchi qonuniyat, qoidalar majmuasi, ya'ni X ma'lumotlar ustida bajariladigan amallar ketma-ketligi yoki tanlab olingan usul.

Masalani echishning ishlab chiqilgan uslubi yoki tanlab olingan usulning to'g'riligi, samaradorligi keyingi bosqichlarda tekshirib aniqlanadi.

4-bosqich. MASALANI ECHISH ALGORITMINI YARATISH.

4-bosqichda asosan masalani echish algoritmi yaratiladi. Masalani echish algoritmi kompyuterning imkoniyatlarini, echish aniqligini xamda masalani kompyuterda echish vaqtini va qiymatini xisobga olgan xolda yaratilsa maqsadga muvofiq kelgan bo'lar edi.

Masalaning algoritmini yaratishda oraliq ma'lumotlarni iloji boricha kamaytirish, tashqi qurilmalar bilan bo'ladigan aloqalarni minimumga keltirish kerak.

Dasturning samarador va unumdorligi, masalani echish algoritmining qanchalik puxta tashkil qilinganligiga bog'liq.

3-4 bosqichlar bir-biri bilan jips, mustahkam bog'langan. Ya'ni yaratilgan uslubni har xil usullar bilan amalga oshirish mumkin, shu sababdan masalani echish uslubi va algoritmining bir nechta variantlari bo'lishi mumkin va keraklisi tanlab olinadi.

Murakkab masalaning algoritmini yaratishda qadamma-qadam oydinlashtirish uslubidan foydalangan ma'qul, har bir qadamda algoritmnining tarkibi sodda va tushunarli bo'lib qolishiga erishmoq kerak.

Masalani algoritmlash jarayonida, algoritmning ba'zi bo'laklarini, lavhalarini, mantiqan alohida qismlarini ifodalashda tipik algoritmlar va amaliyotda tekshirilgan algoritmlardan iloji boricha ko'p foydalangan ma'qul.

Algoritmlashda modullik printsiptidan foydalanish algoritmni o'qishda va dasturlashda qulayliklar yaratadi. Oxir oqibatda masalani echish algoritmi ishchi holatga keltiriladi, ya'ni algoritm grafik ko'rinishda biror algoritmik til vositasida ifodalash darajasiga keltiriladi.

Masalani algoritmlash - masalani kompyuterdan foydalanib echish algoritmini yaratish jarayonidir.

Algoritmlash - masalani echish bosqichi bo'lib, masalaga qo'yilgan shart va talablar asosida oxirgi natijani, masalaning echimini olish uchun ishlab chiqilgan algoritmlarni yaratish bilan shug'ullanadigan informatikaning bo'limidir.

5-bosqich. MA'LUMOTLARNI TAYYORLASH VA TARKIBINI ANIQLASH.

Ma'lumotlarni tasvirlash usulini tanlash algoritmning bajarilishi bilan chambarchas bog'langan. Shu sababdan ma'lumotni tasvirlashning shunday turini, usulini tanlash kerakki, masalani echish jarayoni sodda va tushunarli bo'lsin.

Ma'lumotlar oddiy o'zgaruvchilar ko'rinishida (bu xol juda kam uchraydi), massiv ko'rinishida, alohida ma'lumot fayllari (ketma-ket o'qiladigan yoki bevosita o'qiladigan) ko'rinishida axborot tashuvchida joylashgan bo'lishi mumkin.

6-bosqich. DASTURLASH.

Masalani ishchi xolatga keltirilgan echish algoritmini tanlangan algoritmik til vositasida ifodalash (tavsiflash, tasvirlash) "dasturlash" deyiladi.

Algoritmning xar bir mayda bo'lagi algoritmik tilning operatorlari yordamida, tilning sintaksis va semantika qoidalari asosida yozib chiqiladi. Algoritm mukammal tuzilgan bo'lsa dasturlashda qiyinchilik tug'ilmaydi. Dasturlash jarayonida quyidagi takliflar inobatga olinsa xatolarni tuzatish jarayoni engillashadi.

1. Dastur umumiy bo'lishi kerak, ya'ni ma'lumotlarni aniq biror turiga bog'liq bo'lmasligi kerak, massivning chegara parametrlarini tekshirmoq lozim. Massiv elementlarining soni 0 yoki 1 bo'lib qolish, yoki yuqori chegarasidan oshib ketish xolati.

2. O'zgarmas kattalik xamda o'zgaruvchi kattalik ko'rinishida ishlatish. (Biror o'zgarmas kattalikni boshqasi bilan almashtirish zarurati bo'lib qolsa, dastur matnini chaqirib o'zgartirish kerak - bu noqulay xolat EXE, COM fayllarida aslo mumkin emas).

Dasturda kiritiladigan ma'lumotlarni nazorat qilish qismi bo'lishi kerak.

3. Dasturdagi arifmetik amallarni kamaytirish va dasturning ishlashini tezlatish uchun:

- darajaga oshirish amallari ko'paytirish amali bilan almashtirilgani ma'qul;
- bir xil ma'lumot bilan xisoblanayotgan arifmetik (algebraik) ifodalarni bir marta xisoblab qiymatini biror o'zgaruvchida saqlab ishlatish.
- takrorlashlarni tashkil qilishda takrorlanishning chegarasini tekshirish uchun ifodalardan emas balki oddiy o'uzgaruvchilardan foydalanish.
- takroriy xisoblashlar tarkibida uchraydigan va takrorlanish davomida qiymatini o'zgartirmaydigan ifodalarni takrorlanishdan tashqarida xisoblash.

4. Dasturning xar bir bo'lagi, moduli qismiga tushuntirishlar yozilgan bo'lishi kerak. Dasturdagi tushuntirishlar, masalani echish ketma-ketligini ifodalovchi mantiqiy ketma-ketlikdan iborat bo'lmog'i kerak.

Dasturdagi modullar, qismlar aniq ko'rsatilgan bo'lishi kerak. Takrorlanish boshi va takrorlanish oxiri aloxida qatorda turgani ma'qul.

7-bosqich. DASTUR MATNINI VA MA'LUMOTLARNI AXBOROT TASHUVCHIGA O'TKAZISH.

Kompyuter uchun axborot tashuvchi vositalar bo'lib: perfokarta, perfolenta, magnitli tasma, magnitli disk (egiluvchi magnitli disk, magnitli karta), fleshkalar xizmat qilishi mumkin.

Dastur matni aloxida maxsus qurilmalar yordamida yoki kompyuterdan foydalanib axborot tashuvchiga o'tkaziladi.

8-bosqich. DASTURNING XATOSINI TUZATISH.

Bu bosqich masalani kompyuterda echish bosqichlari ichidagi ko'p vaqt talab qiladigan, mutaxassisdan sabr, qanoat, chidam, aql, zakovat, mantiqiy tez fikrlash, algoritmik tilning barcha imkoniyatlarini, tuzatish (otladka) qilish uslubini, yo'llarini,

masalaning mag'zini ikir-chikirlarigacha mukammal bilishni talab qiladigan murakkab izlanuvchan jarayondir.

Bu bosqich “dasturni test bo'yicha tekshirish” deb xam yuritiladi. Dasturning to'g'ri ishlashi va yo'l qo'yilgan xatoliklarni aniqlab tuzatish algoritmini yaratishda yo'l qo'yilgan kamchiliklarni bartaraf qilish xamda tanlangan usulning yaroqli yoki yaroqsiz ekanligini aniqlab beruvchi jarayondir.

Test - maxsus tayyorlangan dastlabki malumotlar bo'lib, ular ustida amallar bajarish bilan masalaning echimi-natija olinadi. Test tayyorlash juda murakkab ish bo'lib, qo'lda hisob-kitob ishlarini bajarishni talab qiladi xamda dasturning xamma qismlarini, bo'laklarini, modullarini tekshirish shart.

Dasturning xatosini tuzatish bo'yicha yo'l - yo'riqlar:

1. Maxsus tayyorlangan ma'lumotlar asosida dasturni qo'lda echib chiqish (imkoni bo'lsa) yoki mantiqan alohida bo'lgan bo'laklarini, modullarini qo'lda xisoblash.

2. Dasturni va uning bo'laklarini, modullarini test yordamida tekshirish.

3. Dasturning kerakli joylariga bosib chiqarish buyrug'ini qo'yish (tuzatishlardan keyin olib tashlanadi).

4. Dasturning xatolarini tuzatishda, muloqot rejimida bajarilganda (STOP) to'xtash buyrug'idan foydalanish.

5. Dasturlash tilini va amal bajaruvchi tizimi (AT)ning maxsus xatolarni tuzatish imkoniyatlaridan foydalanish.

6. Xatolarni tuzatish jarayonida kam xajmdagi ma'lumotlar bilan ishlashni tashkil qilish.

9-bosqich. DASTURNING AVTOMATIK TARZDA KOMPYUTERDA BAJARILISHI.

Kompyuter xatolari tuzatilib tayyorlangan dastlabki ma'lumotlardan foydalangan xolda masalaning echimini (echimlarini) avtomatik tarzda xisoblaydi.

Agar natijalar masalaning echimi uchun yaroqli deb topilsa masalani echish tugallangan xisoblanadi, aks xolda yuqoridagi bosqichlar qaytadan ko'rib chiqiladi.

10-bosqich. OLINGAN MA'LUMOTLARNI IZOHLASH, TAHLIL QILISH VA DASTURDAN FOYDALANISH UCHUN YO'RIQNOMA YOZISH.

Masalani echish natijasida olingan sonlar yoki sonlar massivi, matnlar yoki matn ko'rinishidagi massivlar xar taraflama izoxlanib, tushuntiriladi. Dasturdan foydalanish uchun ko'rgazma yozish quyidagilarni o'z ichiga oladi:

- Dastur ishlashi uchun ma'lumotlarni tayyorlash usuli, tuzilishi aniq belgilangan;
- Dasturning ishlashi uchun kompyuterni sozlash yo'llari;
- Dasturni ishga tushirish va ishlash paytida bo'ladigan savol-javoblar;
- Dasturni ishlash jarayonida kelib chiqadigan xar xil xolatlarni bartaraf qilish yo'llari aniq va puxta tushunarli qilib yozilgan bo'lishi kerak.

Masalani echishning uchta bosqichini quyidagi misollarda ko'rib chiqamiz.

1-MISOL.

1. Masalaning qo'yilishi va maqsadning aniqlanilishi. Koptok 29, 5 m / sek tezlik bilan tepaga tik ravishda tepilgan. U qancha balandlikka ko'tariladi? (Havoning qarshiligi xisobga olinmasin).

2. Masalani matematik ifodalash.

Berilgan: $V_0 = 29,5 \text{ m / sek.}$; $V = V_0$.

Koptokni balandlikka ko'tarilish xarakatini ifodalovchi qonuniyat:

$$h = V_0 \cdot t - g \cdot t^2 / 2 \quad (1)$$

bu erda: t - koptokning ko'tarilish vaqti, sek. ; g - erkin tushish tezlanishi ($9,8 \text{ m / sek}$);

3. Masalani echish usulini ishlab chiqish.

Koptokning tezligi eng yuqori balandlikka etganda nolga teng bo'ladi:

$V = 0$. Fizika kursidan ma'lumki, tezlik yo'ldan vaqt bo'yicha olingan xosila.

$$V = dh / dt. \quad (2)$$

(1) dan xosila olsak

$$V = V_0 - g \cdot t \quad (3)$$

(3) -ni nolga tenglab t ning qiymatini topamiz:

$$t = V_0 / g \quad (4)$$

(4)-dan t ni topib (1) ga qo'yamiz.

II. ALGORITMNING ASOSIY XOSSALARI, IFODALASH USULLARI, TURLARI

1. Algoritmning asosiy xossalari
2. Algoritmni ifodalash usullari
3. Dasturlash tillari va ularni tasniflash
4. Algoritmning asosiy turlari

1. Algoritmning asosiy xossalari.

Algoritm quyidagi asosiy xossalarga ega: uzluklilik, aniqlik, natijaviylik va ommaviylik.

UZLUKLILIK. Dastlabki berilgan ma'lumotlarni natijaga aylantirish jarayoni uzlukli ravishda amalga oshiriladiki, bunda vaqtning xar bir keyingi keladigan daqiqasidagi miqdor (kattalik)larning qiymati vaqtning shundan oldingi daqiqasida bo'lgan miqdorlar qiymatidan ma'lum bir qoidalar bo'yicha olinadi.

ANIQLIK. Algoritmning xar bir qoidasi aniq va bir qiymatli bo'lishi zarurki, bunda vaqtning biror daqiqasida olingan miqdorlar qiymati vaqtning shundan oldingi daqiqasida olingan miqdorlar qiymati bilan bir qiymatli aniqlangan bo'ladi.

NATIJAVIYLIK. Algoritm masalaning echimiga chekli sondagi qadamlar ichida olib kelishi yoki masalani "echib bo'lmaydi" degan xabar bilan tugashi kerak.

OMMAVIYLIK. Masalaning echish algoritmi shunday yaratilishi kerakki, uni faqat boshlang'ich ma'lumotlar bilan farqlanadigan masalalarni echish uchun xam qo'llanilishi kerak.

Bunda boshlang'ich ma'lumotlar "algoritmni qo'llash soxasi" deb ataladigan birorta soxadan olinadi.

Masalan, yuqoridagi 1 - misolda koptok o'rniga boshqa narsani tik irg'itilsa va uning boshlang'ich tezligi ma'lum bo'lsa, shu algoritm bilan u erishadigan balandlik aniqlanadi.

2. Algoritmni ifodalash usullari va ularga misollar

Algoritmni ishlab chiqishda uni bir necha xil usul bilan ifodalab bersa bo'ladi.

Shulardan uchta keng tarqalgan. Bular:

1. Algoritmni oddiy tilda ifodalash;
2. Algoritmni tuzim ko'rinishida ifodalash;
3. Algoritmni maxsus (algoritmik) tilda yozish.

2.1 Algoritmni oddiy tilda ifodalash

Algoritmni ifodalashning eng keng tarqalgan shakli - oddiy tilda so'zlar bilan bayon qilishdir. Bu nafaqat hisoblash algoritmida, balki hayotiy, turmushdagi "algoritm"larga ham tegishlidir.

Masalan, biror bir taom yoki qandolat mahsulotini tayyorlashning retsepti ham oddiy tilda tavsiflangan algoritmdir. Shaharlararo telefon - avtomat orqali aloqa o'rnatishning o'ziga xos algoritmidan foydalanasiz. Do'kondan yangi kir yuvish mashinasi yoki magnitofon sotib olinsa, ishni foydalanishning algoritmi bilan tanishishdan boshlaymiz.

Masalani kompyuterda echishda ham, ko'pincha matematika tilini ham o'z ichiga olgan tabiiy tildan foydalanish mumkin. Algoritmning bunday tildagi yozuvi izlanayotgan natijaga olib keladigan amallar ketma-ketligi ko'rinishida bo'lib, odam tomonidan bir ma'noli idrok etilishi kerak. So'zlar bilan ifodalar har bir amal "algoritmning qadami" deb ataladi. Qadamlar tartib nomeriga ega bo'ladi.

Algoritm ketma-ket, qadam-ba qadam bajarilishi kerak. Agar algoritm matnida "N sonli qadamga o'tilsin" deb yozilgan bo'lsa, bu algoritmning bajarilishi ko'rsatilgan N-qadamdan davom etishini bildiradi.

Ko'rinib turibdiki, yuqoridagi uchchala misol algoritmi ham oddiy tilda yozilgan ekan.

Algoritmni oddiy tilda ifodalash kompyuterga kiritish uchun yaramaydi. Buning uchun algoritmni kompyuter tilida shunday bayon qilish kerakki, masalan kompyuterda echish jarayonida bu algoritm ishni avtomatik boshqarib turadigan bo'lsin.

Kompyuter tushunadigan shaklda yozilgan algoritm masalani echish dasturidir.

Algoritmni oddiy tilda yozishda to'rt xil amaldan: hisoblash, N- qadamga o'tish, shartni tekshirish, hisoblashning oxiri, shuningdek kiritish va chiqarish amallaridan foydalanilgan maqul. Bular ichida eng ko'p foydalaniladigani hisoblash amalidir.

2.2 Algoritmni tuzim ko'rinishida ifodalash.

Nisbatan murakkab masalalarni echishda algoritmdan muayyan kompyuter tilidagi dasturga o'tish juda qiyin. Bunday bevosita o'tishda algoritmning alohida qismlari orasidagi bog'lanish yo'qoladi, algoritm tarkibining asosiy va muhim bo'lmagan qismlarini farqlash qiyin bo'lib qoladi.

Bunday sharoitda keyinchalik aniqlash va to'g'rilash ancha vaqt talab qiladigan xatolarga osongina yo'l qo'yish mumkin.

Odatda algoritm bir necha marta ishlab chiqiladi, ba'zan xatolarni to'g'rilash, algoritm tarkibini aniqlashtirish va tekshirish uchun bir necha marta orqaga qaytishga to'g'ri keladi.

Algoritm ishlab chiqishning birinchi bosqichida algoritmni yozishning eng qulay usuli - algoritmni tuzim ko'rinishida ifodalashdir.

Algoritm tuzimi - berilgan algoritmni amalga oshirishdagi amallar ketma-ketligining oddiy tildagi tasvirlash elementlari bilan to'ldirilgan grafik tasviridir. Algoritmning har bir qadami tuzimda biror bir geometrik shakl - blok (blok simvoli) bilan aks ettiriladi.

Bunda bajariladigan amallar turiga ko'ra turlicha bo'lgan bloklarga GOST bo'yicha tasvirlanadigan turli xil geometrik shakllar - to'g'ri to'rtburchak, romb, parallelogramm, ellips, oval va hokazolar mos keladi.

Algoritm tuzimlarini qurish qoidalari GOST 19. 002-80 da (xalqaro standart ISO 2636-73 ga mos keladi) qat'iy belgilab berilgan. GOST 19. 003 -80 (ISO 1028-73ga mos) algoritm va dasturlar tuzimlarida qo'llaniladigan simvollar ro'yxatini, bu simvollarining shakli va o'lchamlarini, shuningdek ular bilan tasvirlanadigan funktsiyalarni (amallarni) belgilaydi.

Tuzim blok(simvol)lari ichida hisoblashlarning tegishli bosqichlari ko'rsatiladi. Shu erda har bir simvol batafsil tushuntiriladi.

Har bir simvol (blok) o'z raqamiga ega bo'ladi. U tepadagi chap burchakka chiziqni uzib yozib qo'yiladi. Tuzimdagi grafik simvollar hisoblash jarayonining rivojlanish yo'nalishini ko'rsatuvchi chiziqlar bilan birlashtiriladi.

Ba'zan chiziqlar oldida ushbu yo'nalish qanday sharoitda tanlanganligi yozib qo'yiladi. Axborot oqimining asosiy yo'nalishi tepadan pastga va chapdan o'ngga ketadi. Bu hollarda chiziqlarni ko'rsatmasa ham bo'ladi, boshqa hollarda albatta chiziqlarni qo'llash majburiydir. Blokka nisbatan oqim chizig'i (potok linii) kiruvchi yoki chiquvchi bo'lishi mumkin. Blok uchun kiruvchi chiziqlar soni chegalanmagan.

Chiquvchi chiziq esa mantiqiy bloklardan boshqa hollarda faqat bitta bo'ladi. Mantiqiy bloklar ikki va o'ndan ortik oqim chizig'iga ega bo'ladi.

Ulardan har biri mantiqiy shart tekshirishining mumkin bo'lgan natijalarga mos keladi.

O'zaro kesiladigan chiziqlar soni ko'p bo'lganda, chiziqlar soni haddan tashqari ko'p bo'lsa va yo'nalishlari ko'p o'zgaraversa tuzimdagi ko'rgazmalik yo'qoladi. Bunday hollarda axborot oqimi chizig'i uzishga yo'l qo'yiladi, uzilgan chiziq uchlariga "birlashtiruvchi" belgisi qo'yiladi.

Agar uzilish bitta sahifa ichida bo'lsa, O belgisi ishlatilib, ichiga ikki tarafga ham bir xil harf-raqam belgisi qo'yiladi.

Agar tuzim bir necha sahifaga joylansa, bir sahifadan boshqasiga o'tish "sahifalararo bog'lanish" belgisi ishlatiladi.

Bunda axborot uzatilayotgan blokli sahifaga qaysi sahifa va blokka borishi yoziladi, qabul qilinayotgan sahifada esa qaysi sahifa va blokdan kelishi yoziladi.

Algoritm tuzimlarini qurishda quyidagi qoidalarga rioya qilish kerak.

Parallel chiziqlar orasidagi masofa 3 mm dan kam bo'lmasligi, boshqa simvollar orasidagi masofa 5 mmdan kam bo'lmasligi kerak. Bloklarda quyidagi o'lchamlar qabul qilingan: $a=10, 15, 20$; $b=1, 5 \cdot a$.

Agar tuzim kattalashtiriladigan bo'lsa, a ni 5 ga karrali qilib oshiriladi. Bu talablar asosan 10-bosqichda, dasturga yo'riqnoma yozishda rioya qilinadi.

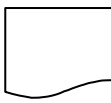
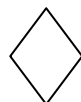
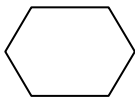
Algoritmnlarni mayda-mayda bo'laklarga ajratishda hech qanday chegaralanishlar qo'yilmagan, bu dastur tuzuvchining o'ziga bog'lik.

Lekin, juda ham umumiy tuzilgan tuzim kam axborot berib, noqulaylik tug'dirsa, juda ham maydalashtirib yuborilgani ko'rgazmalilikka putur etkazadi.

Shuning uchun murakkab va katta algoritmnlarda har xil darajadagi bir nechta tuzim ishlab chiqiladi.

Algoritmning tuzim tarzidagi ifodasining yana bir afzalligi undan uchinchi ko'rinish, ya'ni algoritmik tildagi ifodasi (dastur)ga o'tish ham juda oson bo'ladi. Chunki bunda har bir blok algoritmik tilning ma'lum bir operatori bilan almashtiriladi xolos.

Quyida asosiy bloklar uchun foydalaniladigan shakllar keltirilgan:

ShAKL	Qaysi xolda ishlatiladi	ShAKL	Qaysi xolda ishlatiladi
	Boshlanish va oxirida		Axborotni kiritish va chiqarish
	Xisoblashlar uchun		Natijani chop etish uchun
	Tarmoqlanish shartini tekshirishda		sikl boshlanishida

|

2.3 Algoritmni maxsus tilda ifodalash.

Bu usulda algoritmni ifodalash uchun “dasturlash tillari” deb ataluvchi suniy tillar qo'llaniladi. Buning uchun ishlab chiqilgan algoritm shu tillar yordamida bir manoli va kompyuter tushuna oladigan ko'rinishda tavsiflanishi zarur.

Uning tarkibida cheklangan sondagi sintaksis konstruktsiyalar to'plami bor bo'lib, u bilan algoritm yaratuvchi tanish bo'lishi kerak. Ana shu konstruktsiyalardan foydalanib buyruq va ko'rsatmalar formal ifodalarga o'tkaziladi.

Zamonaviy dasturlash tillari kompyuterning ichki kompyuter tilidan keskin farq qiladi va kompyuter bevosita ana shu tilda ishlay olmaydi. Buning uchun dasturlash tilidan mashina tushunadigan tilga tarjima qiluvchi maxsus dastur - translyatordan foydalaniladi.

Dasturni translyatsiya qilish va bajarish jarayonlari vaqtlarga ajraladi.

Avval barcha dastur translyatsiya qilinib, so'ngra bajarish uslubida ishlaydigan translyatorlar “kompilyatorlar” deb ataladi. Dastlabki tilning har bir operatorini o'zgartirish va bajarishni ketma-ket amalga oshiriladigan translyatorlar “interpretatorlar” deb ataladi.

Dasturlashning ixtiyoriy tili belgilar majmuini va algoritmlarni yozish uchun ushbu belgilarni qo'llash qoidalarini o'z ichiga oladi.

Dasturlash tillari bir biridan alifbosi, sintaksisi va semantikasi bilan ajralib turadi.

Alifbo - tilda qo'llaniladigan ko'plab turli ramziy belgilar (harflar, raqamlar, maxsus belgilar)dir.

Tilning sintaksisi jumlar tuzishda belgilarning bog'lanish qoidalarini belgilaydi, semantikasi esa ushbu jumlalarining mazmuniy izohini belgilaydi.

3. Dasturlash tillari va ularni tasniflash.

Hozirgi kunda dasturlash tillarini u yoki bu belgisi bo'yicha tasniflash mumkin. Dasturlash tilining kompyuterga bog'liqlik darajasi bo'yicha tasniflash eng umumiy hisoblanadi .

Yuqorida aytilgan belgiga qarab, dasturlash tillari kompyutera bog'liq va kompyuterga bog'liq bo'lmagan tillarga bo'linadi.

Kompyuterga bog'lik tillar, o'z navbatida, kompyuter tillari va kompyuterga mo'ljallangan tillarga ajratiladi.

Dasturlash tilining kompyuter tiliga yaqinligi darajasini tariflash uchun til darajasi tushunchasi qo'llaniladi.

Kompyuter tili 0 daraja deb qabul qilingan bo'lib, sanoq boshi hisoblanadi. Odamning tabiiy tili "eng yuqori darajadagi til" deb qaraladi.

Kompyuterga bog'liq bo'lmagan tillar ham ikkita turga bo'linadi: birinchisi protseduraga mo'ljallangan tillar, ikkinchisiga - muammoga mo'ljallangan tillar.

Protseduraga mo'ljallangan tillar turli masalalarni echish algoritmlarini (protseduralarni) tavsiflashga mo'ljallangan; shuning uchun ular ko'pincha oddiy qilib "algoritmik tillar" deb ataladi.

Ushbu tillar echilayotgan masalalar xususiyatlarini to'la hisobga oladi va kompyuterning turiga deyarli bog'liq emas. Bu xildagi tillar tarkibi kompyuter tiliga qaraganda tabiiy tilga, masalan, ingliz tiliga yaqinroq.

Hozirgi kunda hisoblash, muhandis-texnik, iqtisodiy, matnli va sonli axborotlarni taxlil qilish va boshqa masalalarni echish tillari malum.

Masalan: FORTRAN tili 1954 yili ishlab chiqilgan bo'lib, FORMula TRANsla-tor - formulalar translyatori degan manoni anglatadi va ilmiy va muhandis - texnik masalalarni hisoblashlarda qo'llaniladi.

ALGOL tili 1960 yili yaratilgan bo'lib, ALGORITMIC Language -algoritmik til degan ma'noni anglatadi va ilmiy-texnik masalalarni hisoblashlarda qo'llaniladi.

KOBOL tili 1959 yili yaratilgan bo'lib, Common Businees Oriented Language - "savdo-sotiq masalalariga mo'ljallangan til" degan ma'noni anglatadi. Korxona va tarmoqning moddiy boyligini, moliyasini, ishlab chiqargan mahsulotini hisobga olish bilan bog'liq iqtisodiy masalalarni echish uchun qo'llaniladi.

PASKAL tili 1971 yilda e'lon qilingan bo'lib, frantsuz olimi Blez Paskal nomiga qo'yilgan. Turli xildagi masalalar echimini olishda tartiblangan (strukturaviy) dasturlar tuzishda qo'llaniladi.

PL/1 tili 1964 yilda yaratilgan bo'lib, Programming Language/ 1 - 1-tartib raqamli dasturlash tili ma'nosini anglatadi. Ushbu til universal tillar turkumiga kiradi.

Bu tilda ishlab chiqilgan dasturlar kompyuterni yangisi bilan almashtirilganda qaytadan tuzib chiqilishi zarur emas.

BEYSIK (BASIC - Beginner's All Purpose Symbolic Instruction Code - boshlovchilar uchun ko'p maqsadli dasturlash tili) hisoblash algoritmlarini yozish uchun qo'llaniladigan algoritmik tildir. Bu til 1965 yilda Dartmut kolleji xodimlari Kemini va Kurtslar tomonidan ishlab chiqilgan.

Protseduraga mo'ljallangan tillardan masalalarning matematik ifodalari, algoritmlar va dasturlash usullari bilan tanish bo'lgan mutaxassislar foydalaniladilar.

Bunda ulardan kompyuterning tuzilishini mukammal bilish talab qilinmaydi.

Muammoga mo'ljallangan tillar kompyuterda masala echish usullari va dasturlash usullari bilan tanish bo'lmagan foydalanuvchilar uchun yaratilgandir.

Foydalanuvchi masalani tariflashi, boshlang'ich malumotlarni berishi va natijani chiqarishning talab qilingan ko'rinishini aytishi kifoya.

4. Algoritmning asosiy turlari.

Masala echimining algoritmi ishlab chiqilayotgan davrda asosan uch xil turdagi algoritmlardan foydalanib, murakkab ko'rinishdagi algoritmlar yaratiladi.

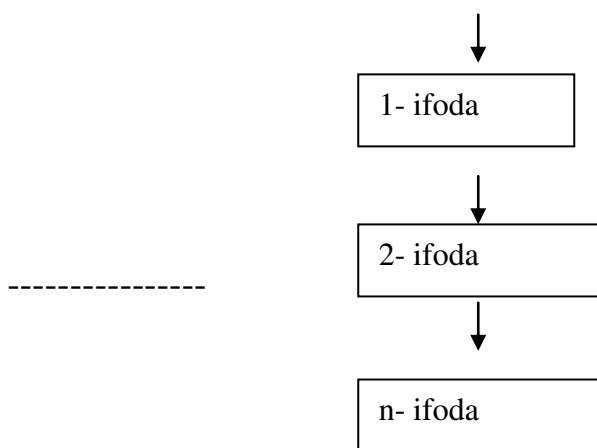
Algoritmning asosiy turlariga chizig'li (a), tarmoqlanadigan (b) va takrorlanadigan (c) ko'rinishlari kiradi.

Murakkab masalalarning echimini olish algoritmlari yuqoridagi turlarining barchasini o'z ichiga olishi mumkin.

Chiziqli turdagi algoritmlarda bloklar biri ketidan boshqasi joylashgan bo'lib, berilgan tartibda bajariladi. Bunday bajarilish tartibi "tabiiy tartib" deb ham yuritiladi.

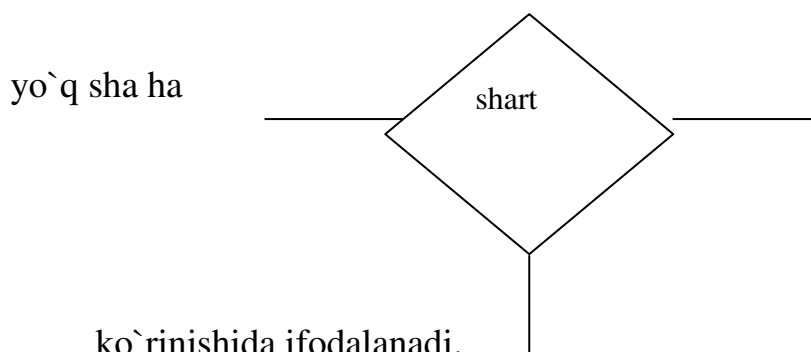
Yuqorida ko'rib o'tilgan birinchi misol chiziqli turdagi algoritmga misol bo'ladi. Amalda hamma masalalarni ham chiziqli turdagi algoritmga keltirib echib bo'lmaydi.

Chiziqli xisoblash jarayonining tuzimi quyidagicha ko'rinishda ifodalanadi.



Ko'p hollarda biron bir oraliq natijaga bog'liq ravishda hisoblashlar yoki u yoki boshqa ifodaga ko'ra amalga oshirilishi mumkin yani birorta mantiqiy shartni bajarilishiga bog'lik holda hisoblash jarayoni u yoki bu tarmoq bo'yicha amalga oshirilishi mumkin. Bunday tuzilishdagi hisoblash jarayonining algoritmi "tarmoqlanuvchi turdagi algoritm" deb ataladi.

Algoritmning bu konstruktsiyasi tuzimda



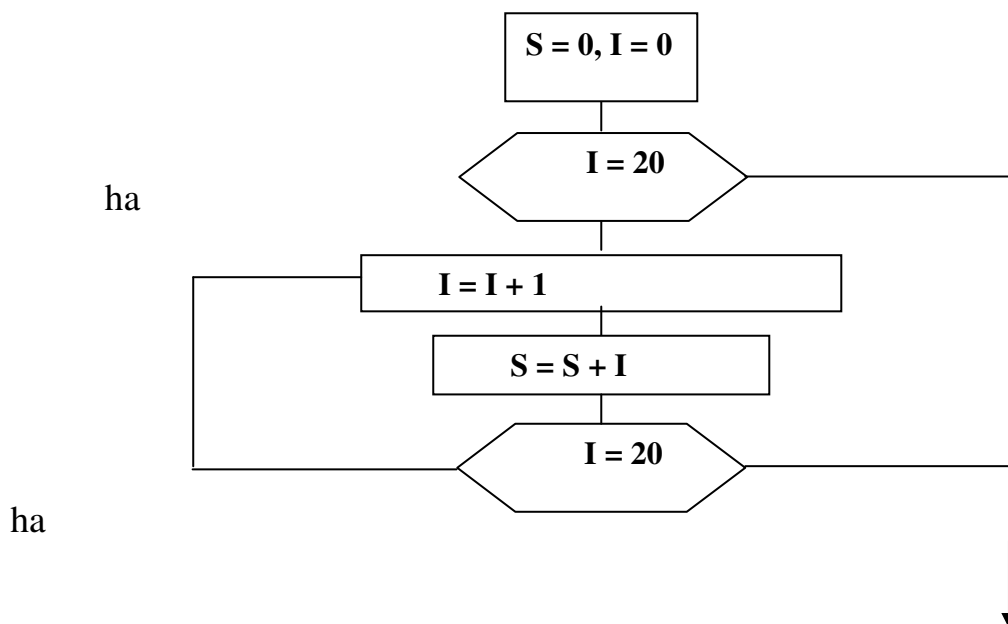
ko`rinishida ifodalanadi.

Ko'pgina hollarda masalalarning echimini olishda bitta matematik bog'lanishga ko'ra unga kiruvchi kattaliklarni turli qiymatlariga mos keladigan qiymatlarini ko'p martalab hisoblash to'g'ri keladi.

Hisoblash jarayonining bunday ko'p martalab takrorlanadigan qismi "takrorlanishlar" deb ataladi. Takrorlanishlarni o'z ichiga olgan algoritmlar "takrorlanuvchi turdagi algoritmlar" deb ataladi. Takrorlanuvchi turdagi algoritmni yozish va chizish o'lchamlarini sezilarli darajada qisqartirish takrorlanadigan qismlarni ixcham ifodalash imkonini beradi.

Yuqoridagi ikkinchi misol takrorlanuvchi turdagi algoritmlarga tegishlidir.

Quyida 1 dan to 20 gacha bo'lgan butun sonlar kvadratlari yig'indisini hisoblash algoritmini tuzim ko'rinishi keltirilgan.



III. ALGORITMLASHTIRISH ASOSLARI. TURBO PASKAL ALGORITMIK TILI

1. Asosiy konstruktsiyasi. Ma'lumotlarning sodda turi
2. Ma'lumotlar turlari
3. Ifodalar

1. Asosiy konstruktsiyasi. Ma'lumotlarning sodda turi

XVII asrda yashagan va dunyoda birinchi hisoblash mashinasini yaratgan frantsuz olimi Blez Paskal sharafiga shunday deb nomlangan.

PASKAL tili Shveytsariyaning Syurix shahridagi Oliy texnika maktabining professori Niklaus Virt tomonidan 70-yillarda yaratilgan bo'lib, 1979 yilda standart Paskal deb tasdiqlangan. O'zining soddaligi, mantiqiyli va samaraliligi tufayli bu til butun dunyoga tezda tarqaldi.

Hozirgi paytda barcha hisoblash mashinalari, xususan, mikrokompyuterlar ham shu tilda ishlash imkoniyatiga ega. Dasturlar matnining tugriligini osonlik bilan tekshirish mumkinligi, ularning ma'nosi yaqqol ko'zga tashlanishi va oddiyligi bilan ajralib turadi.

Paskal tili ancha murakkab va ko'p vaqt oladigan hisoblash ishlarini bajarishga muljallangan tarkiblashtirilgan dasturlar tuzishga imkon beradi.

Yana bir afzalligi shundan iboratki, foydalanuvchi xatolikka yo'l qo'ymasligi uchun yoki xato yozib qo'ygan bo'lsa, tez tuzatib olishi uchun dasturda ishlatilgan o'zgaruvchilar oldindan qaysi turga (toifaga) mansub ekanligi belgilab qo'yilgan bo'ladi.

Shu bilan birga dasturning barcha elementlari haqida ma'lumot tavsiflash bo'limida mujassamlashgan bo'ladi. Operatorlar soni esa minimal darajada kamaytirilgandir.

Turbo Paskalning funktsional tugmachalari. Turbo Paskal muxarririda ishlash jarayonida turli xil funktsional tugmachalardan foydalanish mumkin.

Funktsional tugmachalar klaviaturaning yuqori qismida joylashgan bo'lib, ular F1, ..., F12 tugmachalari xisoblanadi. Bu tugmachalarning xar biriga maxsus buyruqlar birlashtirilgan bo'lib, foydalanuvchiga keng imkoniyatlar yaratadi. Klaviaturadagi maxsus ctrl va alt tugmachalari yordamida funktsional tugmachalar sonini oshirish mumkin.

Bunda maxsus tugmalarning biri bosib turiladi va uni qo'yib yubormagan xolda kerakli funktsional tugmacha bosiladi. Funktsional tugmachalarning fazifalari quyidagilardan iborat:

F1 – muxarrirda ishlash jarayonida yordam xizmatini ko'rsatish;

F2 – dastur matnini EXM xotirasidagi ko'rsatilgan katalogda saqlab qo'yish;

F3 – EXM xotirasidagi dastur matnini muxarrir oynasiga yuklash;

F4 – dasturning kursor turgan joyigacha bo'lgan qismini ishga tushiradi;

F5 – Turbo Paskal muxarriri oynasini butun ekranga yoyish (tugmani qayta bosish bilan avvalgi xolatga qaytiladi);

F6 – muxarrir oynasini EDIT xolatidan WATCH xolatiga o'tkazadi;

F7 – dastur matnini qatorma - qator bajaradi. Bunda xar bir qatorning natijasini aloxida-aloxida ko'rish mumkin;

F8 – bu tugma xam F7 tugmasi kabi bo`lib, faqat protsedurani ishga tushirganda bitta amal kabi qabul qilib, butun protseduraning natijasini qator natijasi kabi ko`rsatadi;

F9 – dasturni kompilyatsiya qilish. Bu tugma dasturni ishga tushirmaydi;

F10 – Paskal muxarririning tavsiyanomasini ishga tushiradi. Tavsiyanoma muxarrir oynasining yuqori qismida joylashgan.

Tavsiyanomadadan chiqish uchun Esc tugmasidan foydalaniladi.

Turbo Paskal muxarririda xam turli xil dastur matnining bo`laklari bilan ishlash imkoniyati mavjud.

Matn bo`lagi ajratib olinishi uchun kursor ajratilishi kerak bo`lgan qismning boshiga olib boriladi va ctrl+K+B tugmasi bosiladi. Kursor ajratilishi kerak bo`lgan matn qismining oxiriga olib boriladi va ctrl+K+K tugmasi bosiladi. Bo`lakni ajratib olishdan voz kechish uchun ctrl+K+N tugmasidan foydalaniladi.

Ajratilgan bo`lakdan nusxa olish uchun kursor nusxa qo`yilishi kerak bo`lgan joyga olib boriladi va ctrl+K+S tugmasi bosiladi.

Ajratilgan bo`lakni boshqa joyga ko`chirish uchun kursor kerakli joyga olib boriladi va ctrl+K+V tugmasi bosiladi.

Ajratilgan bo`lakni o`chirish uchun ctrl+K+Y tugmasi bosiladi.

Turbo Paskal muxarriridan chiqish uchun alt+X tugmasidan foydalaniladi.

Paskal algoritmik tilining alifbosi quyidagilardan iborat:

- 1) 26 ta lotin harflari;
- 2) 0 dan 9 gacha arab raqamlari;
- 3) 32 ta kirill harflari;
- 4) maxsus belgilar - (+, -, *, /, :, ;, [., [., =, >, <)

Paskal tilida so`z deb bir nechta belgilar ketma-ketligi tushuniladi.

Xizmatchi so`z deb Paskal tilidagi standart nom tushuniladi. Bu nom maxsus ma'noni anglatadi va uni ma'lumotlarga berib bo`lmaydi.

Masalan, PROGRAM, BEGIN, END va h.k.

Paskal tilidagi ma'lumotlarning elementlari bo`lib o`zgaruvchilar, o`zgarmaslar, izohlar xizmat qiladi.

O'zgaruvchilar deb hisoblash jarayonida o'z qiymatini o'zgartiradigan kattaliklarga aytiladi. O'zgaruvchilarning nomlari (identifikatorlar) harflardan yoki harf va raqamlardan iborat buladi. Belgilar soni 8 tadan oshmasligi kerak.

O'zgarmaslar (const) deb hisoblash jarayonida o'z qiymatini o'zgartirmaydigan kattaliklarga aytiladi. Bularga ham o'zgaruvchilar kabi nom beriladi.

Izohlar - dasturning ma'lum qismini tavsiflash uchun ishlatiladi va bu qatorda hech qanday amal bajarilmaydi, ya'ni dasturning biror blokini yaxshiroq tushunishga xizmat qiladi.

Izoh (*,*) yoki {} simvollari orasida beriladi. Katta qavslar {*,*} va [*,*] lar ham ishlatilishi mumkin.

2. Ma'lumotlar turlari

Paskal tilida ma'lumotlarning toifalari ikki xil bo'ladi: oddiy-skalyar va murakkab. Skalyar toifa, o'z navbatida, o'zgaruvchi va standart toifalarga bo'linadi.

O'zgaruvchi toifaga qayd qilingan va chegaralangan toifalar kirsa, standart toifaga butun - INTEGER, haqiqiy - REAL, mantiqiy - BOOLEAN, simvol - CHAR toifalar kiradi. Bundan tashqari, Turbo Paskalda qator - STRING turi kiritilgan.

Murakkab toifalarga esa ma'lumotlarning muntazam (massiv), to'plam, aralash (yozuvlar), murojaat va faylli toifalari kiradi.

Ma'lumotlarning standart turlari. Standart turdagi ma'lumotlar dasturda maxsus xizmatchi so'zlar yordamida tavsiflanadi:

INTEGER - BUTUN toifadagi ma'lumotlar faqat butun sonlarni qabul qiladi. Ular Paskal tilida quyidagicha ifodalanadi: -7, 10,89, -35, 0 va h.k.

REAL - haqiqiy toifadagi ma'lumotlar haqiqiy sonlarni qabul qiladi va ikki xil ko'rinishda bo'ladi:

a) qo'zg'almas nuqtali haqiqiy sonlar (3.56, 0.88, -150.45);

b) qo'zg'aluvchan nuqtali haqiqiy sonlar (0.546=546E-03, 96.78*100=96.78E02 va h.k.

CHAR - BELGI (simvol) toifadagi ma'lumotlar qiymat sifatida kodlar jadvalidagi belgilarning bittasini qabul qiladi. Bu ma'lumotlar apostrof ('belgisi) ichiga olib yoziladi. Masalan 'F', 'A', 'R', '5', '8' va h.k.)

BOOLEAN - MANTIQUIY toifadagi ma'lumotlar. Bu nom ingliz matematigi Jorj Bul sharafiga shunday deb ataladi va shu turdagi o'zgaruvchilar faqat ikkita qiymat qabul qiladi, TRUE - ROST yoki FALSE – YOLG'ON.

Bu ma'lumotlar ustida mantiqiy ko'paytirish AND, mantiqiy qo'shish OR va mantiqiy inkor qilish NOT amallarini bajarish mumkin.

Quyidagi jadvalda BOOLEAN turidagi A va B o'zgaruvchilarda mantiqiy amallarning bajarilishi natijasi keltirilgan:

A	B	NOT A	NOT B	A AND B	A OR B
TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE

FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	TRUE	FALSE	

STRING [N] - qator. Bu erda N qatordagi simvollar soni (ko'rsatilmagan bo'lsa, u 256 ga tenglashtiriladi). Bu turdagi ma'lumotlar bitga yoki bir necha belgilar ketma-ketligidan tashkil topib, apostrof ichida beriladi.

Masalan: 'Informatika', 'STUDENT'

Standart funktsiyalar

Paskal tilida quyidagi standart funktsiyalar aniqlangan:

functsiyaningPaskaldagi ifodasi	Functsiyaning matematik ifodasi	Argument turi	Funktsiya turi	functsiyaningPaskaldagi ifodasi	Functsiyaning matematik ifodasi	Argument turi	Funktsiya turi
abs(x)	$ x $	real	real	sqr(x)	x^2	real	real
		integer	integer			integer	integer
Sqrt(x)	$\sqrt{x}$	real	real	exp(x)	e^x	real	real
		integer	real			integer	real
Ln(x)	$\ln x$	real	real	sin(x)	$\sin x$	real	real
		integer	real			integer	real
Cos(x)	$\cos x$	Real integer	real real	arctan (x)	Arctg x	real integer	real
round(x]	X ni yaxlitlash	real	integer	trunc(x)	x ni butun qismini olish	real	integer
pred(x)	X dan oldingi qiymatni olish	Integer Char boolean	Integer Char boolean	succ(x)	x dan keyingi qiymatni olish	Integer Char boolean	Integer Char boolean
adivb	A ni B ga	Integer	Integer	amodb	A ni B ga	Integer	Integer

	bo'lib butun qismini olish	24 div 9	2		bo'lib qoldig'ini olish	17mod 5	2
chr(x)	X songa ko'ra simvolni aniqlash	integer	Char	ord(x)	X simvolning tartib raqamini aniqlash	char	integer
odd(x)	X ning toq eki juftligini aniqlash	integer x-toq x-juft	Boole- an True False				

Turbo Paskal algoritmik tilida ayrim matematik funktsiyalar (x^n , $\text{tg}(x)$, $\text{ctg}(x)$, $\text{arcsin}(x)$, $\text{arccos}(x)$) berilmagan. Lekin bu funktsiyalarni yuqoridagi funktsiyalar yordamida ifodalash mumkin:

$$x^n = \text{EXP}(N * \text{LN}(X));$$

$$\text{tg}(X) = \sin(X) / \cos(X);$$

$$\text{ctg}(X) = \cos(X) / \sin(X);$$

$$\text{arcsin}(X) = \arctan(X / \text{SQRT}(1 - \text{SQR}(X)));$$

$$\text{arccos}(X) = \arctan(\text{SQRT}(1 - \text{SQR}(X)) / X);$$

3. Ifodalar

Paskal algoritmik tilida arifmetik, mantiqiy va belgili ifodalar aniqlangan.

Arifmetik ifodalar o'zgaruvchilar, o'zgarmaslar, arifmetik amal belgilari qavs va standart funktsiyalardan tashkil topadi.

Arifmetik ifodada qatnashayotgan amallar ustivorligiga qarab bajariladi.

Amallar quyidagi ustivorlikka ega:

1. Qavs ichidagi amallar va standart funktsiyalarni hisoblash.

2. Ko'paytirish, bo'lish, DIV, MOD amallari.

3. Qo'shish va ayirish amallari.

Misollar: $\ln x^2 + \cos(x) - \sin^2(x) + e^{3x}$ ifoda berilgan.

Paskal tilida bu ifoda quyidagi ko'rinishda yoziladi:

$\text{Ln}(\text{sqr}(x)) + \cos(x) - \text{sqr}(\sin(x)) + \exp(3*x)$

Quyidagi ifodalar Paskal tilida yozilsin.

1) $5^2 + \sqrt{36} (3-4:6) \text{ sqr}(5) + \text{sqrt}(36)*(3-4/6)$

2) $(2 \cdot 10^5 + 64) : e^x 2E+5 + \exp(4*\ln(6)) \exp(x)$

3) $\cos(5) \cdot \text{tg}(8) - \sqrt{\arcsin(6)} \cos(5) * \sin(8) / \cos(8) - \text{sqrt}(\arctan(6/\text{sqrt}(1-\text{sqr}(x))))$

4) $|x| + [4,5] - (3 \cdot 10^{-6} + \pi) \quad \text{abs}(x) + \text{trunc}(4.5) - (3E-6 + \text{PI})$

Eslatma: Turbo Paskal algoritmik tilida 10^n ko'rinishidagi sonlarni $E+n$, 10^{-n} ko'rinishidagi sonlarni esa $E-n$ kabi tasvirlash mumkin. 10 ning musbat darajasini ifodalashda «+» ishorasini tashlab ketish xam mumkin.

Masalan: $10^6 = E6$; $3000 = 3E3$.

IV. TURBO PASKAL ALGORITMIK TILI DASTURINING TARKIBIY QISMI

1. Belgilar bo'limi
2. Toifalar bo'limi
3. O'zgaruvchilar bo'limi

Dastur deb algoritmik tilda kompyuterda masala echish uchun qadamma-qadam yozilgan aniq va mukammal algoritmgaga aytiladi.

Turbo Paskal tilidagi dastur tarkibi 2 asosiy qismdan iborat bo'ladi. Bular dastur sarlavhasi va dastur tanasi. Dastur tanasidan keyin nuqta qo'yiladi, bu nuqta dastur oxirini bildiradi. Dastur sarlavhasi dastur tanasidan nuqta vergul bilan ajratiladi: <dastur>::q<dastur sarlavhasi>;<dastur tanasi>

Dastur sarlavhasi PROGRAM xizmatchi so'z bilan boshlanadi, bu so'zdan keyin shu dasturga berilgan nom yoziladi (bu nom dastur ichida hech qanday maqsadda ishlatilishi mumkin emas). Dastur nomidan keyin yumaloq qavs ichida dastur parametrlari ro'yxati beriladi. Bu parametrlar fayllar nomi bo'lib, ular orqali dastur tashqi muhit bilan bog'lanadi.

Mavjud bo'lgan ko'p fayllar orasidan 2 ta standart matnli fayllar ajratilgan. Bular dasturni boshlang'ich ma'lumotlar va ma'lumotlarni qayta ishlash natijalari bilan bog'laydigan INPUT va OUTPUTlardir.

Dastur sarlavhasi yozilishini quyidagi misolda ko'ramiz:

PROGRAM PASCAL (INPUT, OUTPUT);

Paskalning ko'p versiyalarida INPUT va OUTPUT yozilmasligi ham mumkin. Umumiy holda dastur tanasi (bloki) 6 bo'limdan iborat va ular aniqlangan ketma-ketlikda joylashgan bo'lishi shart: < blok >::=

- < belgilar bo'limi >
- < konstantalar bo'limi >
- < turlar bo'limi >
- < o'zgaruvchilar bo'limi >
- < protsedura va funktsiyalar bo'limi >

< operatorlar bo'limi >

Dasturning asosiy vazifasi - bu kompyuterga ma'lumotlarni (berilganlarni) qayta ishlash bo'yicha buyruqlar berishdir. Bu buyruqlar operatorlar yordamida beriladi. Shuning uchun ham operatorlar bo'limi asosiy hisoblanadi va har qanday dasturda ishtirok etishi shart.

Operatorlar bo'limidan oldingi bo'limlar ba'zi dasturlarda ishtirok etmasliklari ham mumkin. Ular tavsiflash bo'limlari deyiladi. Tavsiflashning bu bo'limlarida operatorlar bo'limida ishlatilgan kattaliklar tavsiflanadi, ya'ni e'lon qilinadi.

1. Belgilar bo'limi

Dasturning ixtiyoriy operatori oldiga belgi va ikki nuqta qo'yib qolgan operatorlardan ajratish mumkin. Operatorning belgisi bittadan ortiq bo'lishi mumkin emas. Operatorlarning belgilari bir xil bo'lishi mumkin emas.

Operatorning belgisiga uning nomi sifatida xaraladi. Dasturda ishlatilayotgan har bir belgi avval belgilar bo'limida tavsiflash yo'li bilan e'lon qilinishi kerak:

Belgilar bo'limi label (belgi) xizmatchi so'z bilan ochiladi, undan keyin belgilar, ya'ni ketma-ket vergul bilan ajratilgan konstanta shaklida yoziladi. Belgilar bo'limida e'lon qilinayotgan belgilar ixtiyoriy ketma-ketlikda yozilishi mumkin.

Belgilar (0-9999) orasidagi sonlardan olinadi (Turbo Paskalda belgi sifatida alfavitning boshqa elementlarini ham ishlatish mumkin).

Misol: label 24, 25, 26;

Konstantalar-o'zgarmaslar bo'limi

Konstanta deganda aniq bir u yoki bu turdagi qiymatni tushunamiz, dastur ishlash jarayonida bu qiymat o'zgarmaydi shu sababli o'zgarmas deb ham yuritiladi.

Paskal tilida konstantalarga nom beriladi. Konstantalar bo'limi const xizmatchi so'z bilan boshlanadi va nuqta vergul (;) bilan tugaydi.

Misol: const C=34; D=6.3; kaf = 'fizika';

2. Toifalar bo'limi

Turbo Paskal tilida turptta standart turdagi qiymatlar bor, bular: integer (butun), real (haqiqiy), char (matnli), boolean (mantiqiy).

Bu turdagi qiymatlar bilan bir qatorda, Paskal tilida boshqa turdagi qiymatlarni ham ishlatish mumkin. Lekin standart turlardan farqli ravishda boshqa turlar dasturda aniq tavsiflanishi kerak.

Turlar bo'limi type (tur) xizmatchi so'zi bilan boshlanadi, keyin turlar tavsifi yoziladi. Turlar tavsifi bir-biridan nuqta vergul (;) orqali ajratiladi.

Misol:

Ture

Mantiq=boolean;

Hafta=(dush, sesh, chor, payjum, shan, yaksh);

ish kuni= sesh..jum;

Bu misolda uch tur tavsiflari bor. Birinchi qatorda standart mantiqiy turga boshqa nom berilyapti - mantiq. Ikkinchisida yangi qayd qilingan tur kiritilyapti va unga hafta degan nom berilyapti. Uchinchisida yangi chegaralangan tur kiritilyapti, unga ish kuni deb nom berilyapti.

3. O'zgaruvchilar bo'limi

Alohida o'zgaruvchining tavsifi quyidagicha bo'ladi:

<o'zgaruvchi nomi>: <tur>

Misol: a: real; I:integer;

Birinchi qatorda dastur jarayonida ishlatishga real turdagi a uzgaruvchi kiritilyapti, ikkinchisida I nomli butun qiymatli o'zgaruvchi kiritilyapti.

O'zgaruvchilar bo'limi var (variable - o'zgaruvchi) xizmatchi so'z bilan boshlanadi.

Misol: Var i, j, k: integer;

Protseduralar va funktsiyalar bo'limi

Boshqa bo'limlardan farqli ravishda, bu bo'lim maxsus xizmatchi so'z bilan belgilanmaydi. Bu bo'limning boshi Procedure yoki Function so'zi bilan boshlanadi.

Protsedura tavsifi har doim funktsiya tavsifidan oldin turadi. Bu bo'lim dasturda standart protsedura va funktsiyalardan tashqari, protsedura va funktsiyalar ishlatilayotgan bo'lsa va agar ularga dasturdan murojaat qilinayotgan bo'lsa ishlatiladi.

Operatorlar bo'limi

Bu bo'lim dasturning asosiy bo'limi hisoblanadi. Dasturni bajarish bo'limi operatorlik xizmatchi so'zlari begin va end ichiga joylashgan operatorlar ketma-ketligini bajarishga keltiriladi. Operatorlar bir-biridan nuqtali vergul (;) yordamida ajratiladi.

Shuni aytish kerakki, protsedura va funktsiya bo'limidagi algoritmlar faqat operatorlar bo'limidan murojaat qilinganda bajariladi. Operatorlar yordamida masalani echish algoritmini bajarishga kerak bo'lgan amallar tavsiflanadi.

Funksional vazifalari bo'yicha Paskal tilidagi operatorlar quyidagi guruhlariga bo'linadi: o'zlashtirish, kiritish-chiqarish, boshqarish.

O'zlashtirish operatorlari-dasturda o'zgaruvchilarga ma'lum qiymatlarni o'zlashtirish uchun ishlatiladi.

Kiritish-chiqarish operatorlari - tezkor xotiraga boshlang'ich ma'lumotlarni kiritadilar va yozuvga chiqaruvchi qurilmalarga hisoblashlar natijasini chiqaradilar.

Boshqarish operatorlari -dastur operatorlarini ketma-ket bajarilishini tashkil etadi.

V. TURBO PASKAL TILIDA ODDIY ALGORITMLARNI DASTURLASH ASOSLARI

1. O'zlashtirish operatori
2. Ma'lumotlarni kiritish va chiqarish

Shuni eslatib o'tish lozimki, dastur bu masalaning echish jarayonini ifodalovchi, ma'lum bir tilda ifodalangan algoritm bo'lib, kompyuter bajaruvchi vosita hisoblanadi.

Paskal tilida yozilgan dastur to'g'ridan-to'g'ri kompyuterga emas, balki biror bajaruvchiga mo'ljallanganki, unga qanday toifadagi qiymatlar bilan ishlay olishi va qo'shimcha ko'rsatmalarsiz qanday amallarni bajara olishi belgilab berilgan bo'lishi kerak.

U yoki bu masalani echish uchun kattaliklar ustida bajarilishi kerak bo'lgan amallarni ko'rsatish uchun algoritmik tilda operator tushunchasi xizmat qiladi. Dasturdagi har bir operator ma'lumotlarga ishlov berishning mustaqil, mantiqan tugallangan bosqichini ifodalaydi. Paskalda operatorlarning 8 ta turi ko'zda tutilgan. Har qanday masalani echish aniq qoidalar bo'yicha, berilgan kattaliklardan boshqa kattaliklarni keltirib chiqarish jarayonidan iboratdir.

Ko'p oddiy algoritmlarni dasturlashtirishda asosan o'zlashtirish, kiritish-chiqarish operatorlaridan foydalaniladi.

1. O'zlashtirish operatori

Masalani echish jarayoni qator bajariluvchi bosqichlarga bo'linib ketadi. Bu bosqichlarning har birida ma'lum qiymatlar bo'yicha yangi qiymatlar hisoblanadi.

Yangi qiymatlarni hisoblash uchun ifoda tushunchasi xizmat qiladi, har bir ifoda bitta qiymatni hisoblash qoidasini belgilaydi. Hisoblangan qiymatni hisoblash jarayonining keyingi bosqichida foydalanish uchun eslab qolish zarur, bunday eslab qolish hisoblangan qiymatni ma'lum o'zgaruvchiga o'zlashtirish yo'li bilan amalga oshiriladi.

Bunday amalni bajarish, asosiy operatorlardan biri hisoblanuvchi qiymat berish, ya'ni o'zlashtirish operatori bilan bajariladi.

O'zlashtirish operatorining bajarilishida «:=» belgisining o'ng tomonidagi ifodaning qiymati hisoblanib, belgining chap tomonidagi o'zgaruvchi tomonidan o'zlashtiriladi.

Shunday qilib, o'zlashtirish operatori mustaqil mantiqiy tugallangan hisoblash jarayonini ifodalaydi: o'zlashtirish operatori bajarilishi natijasida ma'lum o'zgaruvchilar keyingi bosqichlarda ishlatilishi mumkin bo'lgan yangi joriy qiymatni qabul qiladi.

Masalan:

A:=10; T:=2; H:=4.4;

2. Ma'lumotlarni kiritish va chiqarish

Dasturda ma'lumotlarning qiymatlarini xotiraga kiritishni bir necha usullarda bajarish mumkin.

Sonli o'zgaruvchilarga ularning qiymatini berishda o'zlashtirish operatoridan foydalaniladi. Masalan:

A:=15; B :=-3.1;

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

Kiritish operatori quyidagicha ko'rinishlarda ishlatilishi mumkin:

1) READ(a1,a2,...,an);

bunda, $a_1, a_2, \dots, a_n$ - o'zgaruvchi qiymatlarini ketma- ket standart INPUT protsedura faylidan oluvchi o'zgaruvchilar. O'zgaruvchilarga qiymatlar turiga mos ravishda klaviaturadan kiritiladi.

Aytaylik, A, B, C o'zgaruvchilarga dastur bajarilishi davomida quyidagi qiymatlarni berish kerak bo'lsin: $A=15, B=11, S=2.3$.

Operator READ(A,B,C) ko'rinishiga ega bo'lib, sonlar qiymatlarini dastur bajarilishi davomida quyidagicha kiritish mumkin:

5 17 6.2 [Enter].

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

Masalan:

VAR A, B:REAL;

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

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

3) READLN($a_1, a_2, \dots, a_n$);

- operatorning bajarilishida avval $a_1, a_2, \dots, a_n$ ga qiymat kiritilib, so'ng keyingi satrga o'tiladi. Bu operator oldingi ikki operatorga teng kuchlidir.

KOMPYUTER xotirasidagi ma'lumotlarni displey ekraniga chiqarish operatori - WRITE dir. Operator quyidagi bir nechta ko'rinishlarda ishlatilishi mumkin:

1) WRITE($a_1, a_2, \dots, a_n$);

bunda $a_1, a_2, \dots, a_n$ oddiy o'zgaruvchilar, o'zgaruvchilarning yoki ifodalar bo'lishi mumkin va ular standart OUTPUT protsedura fayliga chiqariladi.

Masalan:

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

B ning qiymati = va undan so'ng B o'zgaruvchining qiymatini chiqaradi.

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

Masalan:

WRITE(Y:4:2); operatori bilan Y ning qiymatini chiqarishda, Y ning hamma qiymatini chiqarish uchun 4 ta xona ajratilishi, ulardan ikkitasi kasr qismi uzunligini anglatadi (bunda sonning butun va kasr qismini ajratuvchi vergul (nuqta) ham hisobga olinishi zarur).

Butun sonlarni chiqarishda kasr qismi formati ko'rsatilmaydi.

Aytaylik, N=111 butun sonli qiymatni chiqarish kerak bo'lsin. Chiqarish operatori buning uchun quyidagi ko'rinishda bo'lishi mumkin:

WRITE ('N=', N:3)

Paskal tilida boshqa chiqarish operatorlari ham ishlatiladi.

Parametrlarsiz chiqarish operatori

WRITELN - displey ekranida yangi satrga o'tishni ta'minlaydi.

3)WRITELN(a1,a2,...,an);

- chiqarish operatori oldin a1,a2,...,an larning qiymatlarini chiqaradi, so'ng yangi qatorga o'tishni ta'minlaydi. Shunday qilib, bu ham quyidagi ikki operatorga ekvivalent:

WRITE(a1,a2,...,an); WRITELN;

Quyida oddiy masalalarni dasturlash namunalari ko'rsatilgan.

1)Uchburchakning ikkita kateti berilgan $a=3$ sm va $b=5$ sm. gipotenuza uzunligini topish dasturini tuzish.

Program gipot;

Const a=3; b= 5;

Var c: real;

Begin

c:=sqrt (sqr(a)+sqr(b));

Write('c=',c)

End.

2) Arrenius qonuniyati asosida reaktsiya tezligi konstantasini temperaturaga bog'liqligini ifodalovchi formula uchun dastur tuzish.

Program arren;

```

Var K, k0,E,R,T:real;
Begin
  Write('o'zgaruvchilarni kiriting');
  Read(k0,E,R,T);
  K:= k0* exp(-E/R*T);
  Write ('K=',K)
End.

```

VI. TARMOQLANUVCHI HISOBLASH JARAYONLARINI ALGORITMLASH VA DASTURLASH

1. Shartsiz o'tish operatori
2. Shartli o'tish operatori
3. Tanlash operatori

Ko'pgina masalalarni echishda ba'zi bir jarayonlar ma'lum shart yoki shartlarning qo'yilishiga nisbatan bajariladi. Bunday jarayonlar tarmoqlanuvchi jarayonlar deb yuritiladi.

Tarmoqlanuvchi hisoblash jarayonlari oddiy va murakkab bo'lishi mumkin. Bu esa jarayondagi tarmoqlar soniga bog'liq. Ma'lum bir tarmoqlanuvchi jarayon tarkibida yana tarmoqlanishlar bo'lishi mumkin. Bunday tarmoqlanishlari bor bo'lgan hisoblash jarayonlari murakkab tarmoqlanuvchi hisoblash jarayonlari deb ataladi.

Paskal tilida tarmoqlanuvchi jarayonlarni dasturlash uchun shartsiz, shartli o'tish va tanlash operatorlaridan foydalaniladi.

1. Shartsiz o'tish operatori

Dasturda ba'zi bir hollarda boshqaruvni to'g'ridan-to'g'ri biron-bir operatorga uzatishga, ya'ni dasturning bajarilish ketma-ketligini buzishga to'g'ri keladi. Bu jarayon shartsiz o'tish operatori yordamida bajariladi. Shartsiz o'tish operatorining umumiy ko'rinishi quyidagicha:

GOTO < operator belgisi>;

Bu erda operator belgisi boshqaruv uzatiladigan operator belgisidir. Belgi sifatida 0-9999 oraliqdagi natural sonlar va CHAR turidagi belgilar ishlatiladi. Belgi tavsiflash bo'limining LABEL bo'limida albatta tavsiflangan bo'lishi shart.

Misol: GOTO 32;

25: WRITE('y=',Y);

2. Shartli o'tish operatori

Dasturda boshqaruvni ma'lum shart asosida u yoki bu tarmoqqa uzatish shartli o'tish operatori yordamida amalga oshiriladi. Shartli o'tish operatori ikki xil ko'rinishda ishlatilishi mumkin: to'liq va qisqa.

Shartli o'tish operatorining to'liq ko'rinishi:

IF <mantiqiy ifoda> THEN S1 ELSE S2;

Bu erda IF (agar), THEN (u holda) va ELSE (aks holda) degan xizmatchi so'zlar, S1 va S2 ixtiyoriy operatorlar.

Operatoridagi mantiqiy ifoda boshqaruvni uzatish shartini belgilaydi.

Operatorning ishlash tartibi quyidagicha: Agar keltirilgan mantiqiy ifoda TRUE (rost) qiymatni qabul qilsa, ya'ni qo'yilgan shart bajarilsa, THEN - xizmatchi so'zidan keyingi operator bajariladi, aks holda ELSE xizmatchi so'zidan keyingi operator bajariladi.

Mantiqiy ifodalarda munosabat amallari, mantiqiy amallar ishlatilishi mumkin.

Masalan,

$A > 5$, $A = B$, $X < 1.5$ va h.k.

Shartlar oddiy va murakkab bo'lishi mumkin. Agar mantiqiy ifodada bitta munosabat amali berilgan bo'lsa, «oddiy shart» ni ifodalaydi.

Kattaliklar orasidagi shartlar HAM, YOKI, EMAS (Paskal tilida AND, OR, NOT) mantiq amallari belgilari orqali bog'lanuvchi bir necha munosabatlardan iborat bo'lsa „murakkab shartlar" deb ataladi.

Masalan,

Matematik yozilishi Algoritmik tilda yozilishi

1) $2 \leq X < 5$ ($X \geq 2$) AND ($X < 5$)

AND amalining natijasi uning ikkala argumenti ham rost bo'lsa rost bo'ladi.

OR amalining natijasi rost bo'lishi uchun argumentlardan birining rost bo'lishi etarli.

NOT amalining natijasi argumentning inkor qiymatiga teng, ya'ni argument rost bo'lsa - natija yolg'on, argument yolg'on bo'lsa - natija rost bo'ladi.

Masalan:

$(4 < 5) \text{ AND } (5 < 100)$ - mantiqiy ifoda TRUE (rost),

$(\text{SIN}(X) > 1) \text{ AND } (5 \text{ DIV } 2 = 0)$ ifoda FALSE (yolg'on) qiymatga teng.

Shuni ta'kidlab o'tish kerakki, agar mantiqiy ifodalar, biz yuqorida aytganimizdek, mantiqiy amallar yordamida (AND, OR, NOT) murakkab ko'rinishga ega bo'lsa, ular qavslarga olib yoziladi.

Shartli o'tish operatorining ishlatilishini misollarda ko'rib chiqamiz.

1) IF $x > 0$ THEN $y := \text{SQRT}(x)$ ELSE $z := \text{sqr}(x)$;

Operatorning bajarilishi natijasida $x > 0$ bo'lsa, u holda $y := \text{sqrt}(x)$ operatori, aks holda $z := \text{sqr}(x)$ operatori bajariladi.

Ayrim algoritmlarda ba'zan shunday xol uchrashi mumkinki, bunda hisoblash jarayonida ayrim amallar ba'zi bir shartlar bajarilgandagina hisoblanadi, aks holda, hech qanday amal bajarilmaydi. Bu holda shartli o'tish operatorini qisqa ko'rinishda ifodalash mumkin.

IF <mantiqiy ifoda> THEN <operator>;

Misol:

IF $X < 1$ THEN $Y := \text{sqr}(X)$;

Shartli o'tishda operator o'rnida, o'z navbatida, yana shartli o'tish operatorining to'la va qisqa ko'rinishlari ishlatilishi mumkin. Masalan:

1) IF B1 THEN IF B2 THEN A;

Bu erda B1, B2 - mantiqiy ifoda, A - operator.

Bu operatorning bajarilishi natijasida B1 mantiqiy ifoda tekshiriladi, agar TRUE qiymat qabul qilsa, B2 mantiqiy ifoda tekshiriladi, u ham rost bo'lsa (TRUE), A operator bajariladi.

Agar B1 yoki B2 mantiqiy ifodalar yolg'on bo'lsa (FALSE), shartli o'tish operatoridan keyingi operator bajariladi.

Agar shartli o'tish operatorida THEN yoki ELSE dan keyin bir necha operator guruhi bajarilsa, ular tarkibiy operator ko'rinishida yozilishi kerak, ya'ni operatorlar qavsi - BEGIN va END lar orasida yoziladi.

3. Tanlash operatori

Juda ko'p tarmoqlanish jarayonlarida tarmoqlanish ikki yoki undan ortiq tarmoqqa ajraladi. Umuman olganda, buni bizga tanish shartli o'tish operatori yordamida amalga oshirish mumkin:

IF B1 THEN A1 ELSE

IF B2 THEN A2 ELSE

IF BK THEN AK ;

Lekin bu hollarda shartli o'tish operatorlarining yozilishi noqulay.

Ko'p hollarda dasturchi uchun shartli operatorning umumiydashgan ko'rinishi - tanlash (variant) operatorini ishlatish qulay.

Tanlash operatorining metaformulasi quyidagicha yoziladi:

< tanlash operatori>::= CASE <operator selektori> OF < tanlash ruyxati elementi>; END

bunda:

Tanlash operatorining umumiy ko'rinishi:

CASE S OF

M1 : A1;

M2: A2;

Mp: An

END;

Bu erda CASE (tanlash) -xizmatchi so'z, OF (dan), S - selektor, Mi – operatorlar belgilari, Ai -operatorlar (i=1 dan n gacha).

CASE operatori tarmoqlanish jarayonini berilgan bir necha operatorlardan birini tanlash yo'li bilan amalga oshiradi. Tanlash operatorida barcha operatorlar, shu jumladan bajarilishi uchun tanlangan operator ham aniq ravishda keltiriladi (berilgan operatorlar ketma-ketligi chegaralangan).

Bajarilishi kerak bo'lgan operator yoki operatorlar ketma-ketligi operator selektorining qiymatiga ko'ra aniqlanadi. Operator selektori sifatida haqiqiy bo'lmagan, skalyar ko'rinishdagi har qanday ifoda yoki o'zgaruvchi ishlatilishi mumkin.

Operatorning ishlashida uning tarkibidagi har bir operator tanlash belgisi deb ataluvchi belgi bilan ta'minlanadi. Bu belgi operatorning bajarilishi uchun zarur bo'lgan selektorning maxsus qiymatini qabul qiladigan selektorning tavsifiga mos konstantadir. Operator bir necha mavjud qiymatlar bilan ishlashi uchun, unda tanlash belgilari ro'yxati keltirilishi kerak.

Tanlash operatoridagi belgili operatorlar oddiy belgiga ham ega bo'lishlari mumkin. Bu holda oldin tanlash belgilari, so'ngra oddiy belgilar yoziladi.

Shuni ham inobatga olish lozimki, tanlash operatoriga faqat CASE xizmatchi so'z orqali kirish mumkin, ya'ni tanlash operatoridan tashqaridagi o'tish operatori orqali bu operatorga murojaat qilish mumkin emas.

Tanlash operatorining bajarilishi uning tarkibidagi operatorlar ketma-ketligidagi bitta operatorning bajarilishiga olib keladi. Shuning uchun ularning biridan biriga GOTO operatori yordamida o'tish xato demakdir.

Shartli o'tish operatorining quyidagi

IF B THEN A1 ELSE A2

ko'rinishi tanlash operatorining quyidagi ko'rinishiga ekvivalentdir:

CASE B OF

TRUE: A1;

FALSE:A2;

END;

qisqa ko'rinishdagi shartli o'tish operatorining IF B THEN A ko'rinishi tanlash operatorining quyidagi ko'rinishga ekvivalentdir:

CASE B OF

TRUE: A;

FALSE

END;

Misol:

CASE T OF

'*', '/': R:=1;

'+', '-': R:=2

End;

Bu operatorning bajarilishi natijasida, agar T-belgili o'zgaruvchi "+" yoki "-" belgi qiymatlarni qabul qilsa, R o'zgaruvchi 2 qiymatni, agar T o'zgaruvchi "*" yoki "/" belgini qabul qilsa, R o'zgaruvchi 1 qiymatni qabul qiladi.

Misollar.

1) Ikkita son berilgan bo'lsin. Agar birinchi son absolyut qiymat jihatidan ikkinchi sondan katta bo'lsa, u holda birinchi sonni besh marta kamaytirish, aks holda sonlarni o'zgarishsiz qoldirish dasturi tuzilsin.

Program misol;

Var x,y; real;

Begin

Read(x,y);

If abs(x) > abs (y) then x:=x / 5;

Write (x,y)

End.

2) Agar temperatura T 30^0 dan oshib ketsa, u holda bosim P $10T^2$ ga , 30^0 dan kamayib ketsa, u holda $15 T^3$ ga, 30^0 ga teng bo'lsa u holda bosim $10 T^3 + \sin^2 T$ ga teng bo'ladi. Shunday holatni ifodalovchi dastur tuzish.

Program temp;

Var P,T: real;

Begin

Read(T);

If $T > 30$ then $P := 10 * \text{sqr}(T)$;

If $T < 30$ then $P := 15 * \exp(3 * \ln(T))$ else $P := 10 * \exp(3 * \ln(T)) + \text{sqr}(\sin(T))$;

Write ('P= ', P)

End.

Mavzuni mustaxkamlash uchun topshiriqlar

1) $\arcsin(1 + \ln x) = a$ tenglama ildizi mavjud bo'lsa, u x o'zgaruvchiga berilsin.

2) x va y o'zgaruvchilarning berilgan qiymatlari shunday qayta taqsimlansinki, x ushbu qiymatlarning kattasini, y esa kichigini olsun.

3) Berilgan to'rtta sonlardan uchta o'zaro teng va bittasi ulardan farqli. Ushbu element nomerini o'zgaruvchiga o'zlashtirilsin.

4) a, b, c o'zgaruvchilar qiymatlarini shunday almashtirilsinki, natijada quyidagi tengsizlik o'rinli bo'lsin: $a \geq b \geq c$.

5) x va y o'zgaruvchilar mos ravishda

a) $x = y = 1$; b) $x = 1; y = -1$; c) $x = -1; y = 1$

qiymatlarni qabul qilganda va quyidagi amallar bajarilganda z o'zgaruvchi qiymati xisoblansin:

$z := 0$ if $x > 0$ then $z := 1$ else $z := 2$

6) Xatolar ko'rsatilsin:

a) if $1 < x < 2$ then $x := x + 1; y := 0$; else $x := 0; y := y + 1$;

b) if $1 < x$ and $x < 2$ then begin $x := x + 1; y := 0$ end;

else begin $x := 0; y := y + 1$ end

7) Agar x va y sonlari teng bo'lsa, b mantiqiy o'zgaruvchiga true, aks xolda false qiymati berilsin.

8) Agar $n = 3$ bo'lsa, quyidagi tarkibiy operatorning bajarilishi natijasida f o'zgaruvchi qanday qiymat qabul qiladi?

Begin

$f := 1; i := 2$;

1: if $i > n$ goto 9;

$f := f * i; i := i + 1$; goto 1;

9: end.

9) Quyidagi savollarga javob bering:

a) belgi sifatida ixtiyoriy musbat butun son dan foydalanish mumkinmi?

b) Dasturda foydalanilgan belgilarning xammasini e'lon qilish shartmi?

c) Dasturda belgilarni o'sish tartibida e'lon qilinishi to'g'rimi?

10) Dasturdagi xatolarni toping:

program hato;

const l = 18;

```

label 18,8;
var   x,y: integer;
begin
    y := 0;
8: read(x); if x < 0 then goto 1;
    y := y + x; goto 18;
1: writeln(y)
end.

```

11) Quyidagi operator yozuvidagi xatolarni toping:

- a) begin 77: if $|x| < 1$ then goto 5
begin $x := x/2$; goto 77; 5: $y := \sin(x)$ end end;
- b) 34: if $0 < x < 2$ then if $y < 1$ then goto 34 else goto 15
else 15: $y := \text{sqr}(y)$

12) Quyidagi operatorlar bajarilgandan keyin p va d o'zgaruvchilarning qiymatlari nimaga teng bo'lishi aniqlansin.

```

P := true; d := 1;
Case k mod 10 of
    3, 2, 7, 5: d := k;
1: ;
    4, 8: begin p := false; d := 2 end;
    9, 6: begin p := false; d := 3 end;
end {of case}

```

13) Quyidagi tanlash operatoridagi xatolarni toping:

```

var a: char;
    d: real;
...
case a of
    1, 2, 3: d := d + 1;
    '1', '2', '3': d := d + 1; write('d +', d);
end {of case}

```

...

14) Berilgan figura uchburchak, to'rtburchak yoki doira ekanligiga qarab mos ravishda ularning yuzalarini xisoblash dasturini tuzing.

15) Quyidagi operatorlarni bajarilish natijasini hisoblang:

$Z := 0;$

if $X > 0$ then if $y > 0$ then $z := 1$ else $z := 2$

A) $x = y = 1$; B) $x = 1; y = -1$; B) $x = -1; y = 1$.

16) Quyidagi operatorlarni yozilishidagi xatoni toping.

a) IF $1 < x < 2$ then $x := x + 1; y := 0;$

else $x := 0; y := y + 1;$

b) if $1 < x$ end $x < 2$ then begin $x := x + 1;$

$y := 0$ end; else begin $x := 0; y := y + 1$ end

17) x, u, z qiymatlari berilgan bo'lsin. Quyidagi ifodalarning dasturi tuzilsin:

A) $\max(x + y + z, xyz)$

B) $\min(x^2 + y^2 + z^2)$

18) Agar $n = 3$ bo'lsa, quyidagi operatorlarni bajarilishi natijasida f o'zgaruvchi nimaga teng bo'ladi.

BEGIN $f := 1; i := 2;$

1) if $i > n$ then goto 9

$f := f * i; i := i + 1; goto 2;$

9 : end.

19) Quyidagi operatorlar yozilishining xatosini toping:

A) BEGIN 100 : if $(x) < 1$ then goto 10

BEGIN $x := x / 2; goto 100;$

10: $y := \sin(x)$ end

end;

b) 5 if $0 < x < 2$ then if $y < 2$ then goto 5

else goto 15 else.

15: $y := \sqrt{y};$

20) CASE tanlash operatorini ishlatib, yilning oylari nomerlari bo'yicha chiqarish dasturini tuzing.

VII. TAKRORLANUVCHI HISOBLASH JARAYONLARINI ALGORITMLASH VA DASTURLASH

1. Takrorlanish jarayonlarini tashkil qilish
2. Sharti avval tekshiriladigan takrorlanish jarayoni
3. Sharti keyin tekshiriladigan takrorlanish jarayoni
4. Parametrli takrorlanish jarayoni
5. Murakkab takrorlanish jarayonlari

1. Takrorlanish jarayonlarini tashkil qilish

Amaliyotda murakkab jarayonlarni dasturlashda ma'lum buyruqlar ketma-ketligini ma'lum shartlar asosida qayta-qayta bajarish zaruriyati tug'iladi. Ma'lum bir o'zgaruvchining turli qiymatlarida ma'lum buyruqlar tizimining biron-bir qonuniyatga asosan qayta-qayta bajarilishi takrorlanuvchi hisoblash jarayoni - sikl deb ataladi.

Takrorlanuvchi hisoblash jarayonining takror-takror hisoblanadigan qismini takrorlanishning tanasi (jismi) deb ataladi.

Takrorlanish ichida qiymatlari o'zgarib boradigan o'zgaruvchini takrorlanish o'zgaruvchisi yoki takrorlanishni boshqaruvchi o'zgaruvchisi (sikl parametri) deb yuritiladi.

Takrorlanuvchi jarayonning algoritmi umumiy holda quyidagilarni o'z ichiga olishi kerak:

1. Takrorlanishni tayyorlash - takrorlanishni boshlashdan oddin takrorlanishda qatnashadigan o'zgaruvchilarning boshlang'ich qiymatlari yoki takrorlanish o'zgaruvchisining boshlang'ich qiymati o'rnatiladi, takrorlanish o'zgaruvchisining o'zgarish qadami belgilanadi.

2. Takrorlanish tanasi - takrorlanish o'zgaruvchilarining turli qiymatlari uchun takror bajariladigan amallar ketma-ketligi ko'rsatiladi.

3. Takrorlanish o'zgaruvchisiga yangi qiymat berish - har bir takrorlanishdan avval o'zgaruvchiga o'zgarish qadamiga mos ravishda yangi qiymat beriladi.

4. Takrorlanishni boshqarish - takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga o'tish ko'rsatiladi.

Paskal algoritmik tilida uch xil ko'rinishda takrorlanuvchi hisoblash jarayonini tashkil qilish mumkin va bu jarayonlarni dasturlash uchun maxsus operatorlar belgilangan:

- sharti avval tekshiriladigan takrorlanish jarayoni, bu maxsus WHILE operatori orqali amalga oshiriladi;

- sharti keyin tekshiriladigan takrorlanish jarayoni, bu jarayonni maxsus REPEAT operatori orqali amalga oshiriladi;

- parametrli takrorlanish jarayoni, bu jarayon maxsus FOR operatori yordamida amalga oshiriladi.

2. Sharti avval tekshiriladigan takrorlanish jarayoni

Takrorlanuvchi jarayonning bu ko'rinishi takrorlanish soni oldindan noma'lum bo'lgan hollarda, ya'ni takrorlanishdan chiqish ma'lum shartga bog'liq bo'lgan hollarda ishlatiladi. Takrorlanishning bu jarayonida takrorlanishdan chiqish sharti takrorlanish tanasini bajarishdan oldin tekshiriladi.

Ushbu operatorning umumiy ko'rinishi quyidagicha:

WHILE L DO M ;

bu erda, **WHILE** - toki, do - bajarish ma'nosini anglatuvchi xizmatchi so'zlar, L - mantiqiy ifoda, M - operatorlar yoki operatorlar guruhi, u takrorlanish tanasini belgilaydi. Takrorlanish tanasida bitta yoki bir necha operatorlar guruhi bo'lishi mumkin.

Bunda operatorlar guruhi, albatta, Begin va End orasida yozilishi kerak.

Operatorning bajarilishi quyidagicha: L mantiqiy ifodaning qiymati hisoblanadi. Agar L mantiqiy ifoda rost qiymatga ega bo'lsa, M operatori bajariladi va bu operator L mantiqiy ifodaning qiymati yolg'on bo'lgungacha qayta-qayta bajariladi.

Agar L mantiqiy ifodaning qiymati birinchi tekshirishdayoq yolg'on bo'lsa, M operatori biron marta ham bajarilmaydi va boshqaruv WHILE operatoridan keyingi operatorga uzatiladi.

Misollar.

1) N! ni hisoblash dasturi tuzilsin.

Program fact;

Var n, i, p: integer;

Begin

Read(n); P:=1; I:=0;

While I<n do

Begin

I:=I+1;P:=P*I;

End;

Write ('n! =', P)

End.

2) Sonni o'nlik sanoq tizimidan ikkilikka o'tkazish dasturi tuzilsin

PROGRAM sanoq;

TYPE massiv=array [1..50] of integer;

var a:massiv;

n,i:integer;

begin write('введите число:');

readln(n);

i:=1;

while n>=2 do

begin a[i]:= n mod 2;

i:=i+1;

n:= n div 2;

end;

i:=i-1;

write(n);

```

while i<>0 do
begin write(' ',a[i]);
i:=i-1;
end;
writeln
end.

```

3) Quyidagi dasturning bajaradigan funktsiyasini tushuntiring

```

program pasc;
const a = pi / 4;
b = pi/2;
h = pi/60;
var x,fx :real;
begin x:=a;
writeln(' x y');
while x<=b do
begin fx:=sin(x)+0.5*cos(x);
writeln ('f (' ,x:4:2,')=',fx:4:2);
x:=x+h;
end;
end.

```

3. Sharti keyin tekshiriladigan takrorlanish jarayoni

Takrorlanish jarayonining bu ko'rinishi ham takrorlanish soni oldindan ma'lum bo'lmagan hollarda ishlatiladi. Bunday jarayonda biror buyruq yoki buyruqlar tizimi berilgan shart bajarilgunga qadar takror va takror bajariladi. Bu jarayonni avvalgisidan farqi shundaki, bunda takrorlanishning tanasi hech bo'lmaganda bir marta bo'lsa ham bajariladi, chunki takrorlanishdan chiqish sharti takrorlanishning tanasi bajarilgandan keyin tekshiriladi.

Bunday takrorlanish jarayonini dasturlashda maxsus REPEAT operatoridan foydalaniladi.

Ushbu operatorning umumiy ko'rinishi quyidagicha:

REPEAT M UNTIL L;

bu erda

REPEAT - takrorlamoq, UNTIL - «gacha» ma'nosini anglatuvchi xizmatchi so'zlar;

M - takrorlanish tanasi, uning tarkibida bitta operator yoki operatorlar guruhi bo'lishi mumkin. Operatorlar guruhini Begin va End orasiga olish shart emas.

Operator quyidagi tartibda bajariladi:

- takrorlanish tanasiga kirgan operatorlar birin-ketin bajariladi. So'ng L mantiqiy ifodaning qiymati topiladi, ya'ni shart tekshiriladi. Agar bu shart bajarilsa (L ning qiymati TRUE (rost) bo'lsa) boshqaruv takrorlashdan tashqariga, UNTIL so'zidan keyingi operatorga uzatiladi. Aks holda takrorlanish davom etadi.

Takrorlanish jarayonining bunday ko'rinishi yuqorida keltirilgan takrorlanishdan shu bilan farq qiladiki, bunda takrorlanish tanasi hech bo'lmaganda bir marta bajariladi.

M i s o l.

1) $S=1+1/2 +1/3+...+1/n$ yig'indini hisoblash dasturi tuzilsin.

program prep(input,output);

var i,n : integer; s: real;

begin

read(n);

s:=0;I:=1;

Repeat

S:=S+1/ I;

Write (s);

I:=I+1;

Until I>n

End.

4. Parametrli takrorlanish jarayoni

Yuqorida keltirilgan takrorlanish operatorlarini, odatda, takrorlanish soni noma'lum bo'lgan hollarda ishlatish maqsadga muvofiqdir. Takrorlanish jarayonining

takrorlanish soni uni bajarilishidan oldin ma'lum bo'lsa hamda takrorlanish o'zgaruvchisining qiymatlari orasidagi farq (o'zgarish qadami) 1 ga yoki -1 ga teng bo'lsa, parametrli takrorlanish operatoridan foydalanish qulay.

Bu buyruq quyidagi ko'rinishga egadir:

FOR V:=L1 TO (DOWNTO) L2 DO M

Bu erda: FOR operatorning nomi; V- takrorlanish o'zgaruvchisining nomi; L1, L2 - mos ravishda V ning boshlang'ich va oxirgi qiymati; TO - "gacha" ma'nosini anglatuvchi xizmatchi so'z; DOWNTO - V ning qiymati -1 ga o'zgarganda TO ning o'rnida ishlatiladi; DO - bajarilsin degan ma'noni anglatadi; M - takrorlanish tanasi.

Tanada bir necha operator bo'lsa, ular Begin va End xizmatchi so'zlari orasida yoziladi.

Takrorlanish o'zgaruvchisining toifasi REAL va cheklanmagan INTEGER toifasidan tashqari, barcha oddiy toifalarda berilishi mumkin va uning qiymatlari orasidagi farq bitta kattalikka teng. L1 va L2 V ning toifasidagi kattaliklar bo'lib, uning o'rnida o'zgaruvchi nomlarini, ifodalarni va o'zgarmaslarni ishlatish mumkin.

Misol. 1 dan 100 gacha bo'lgan musbat sonlar yig'indisini hisoblash dasturini tuzish.

```
Program sum;  
Var s:real; I: integer;  
Begin  
S:=0;  
For I:=1 to 100 do  
Begin  
S:=S+I;  
End;  
Write('s=',s)  
End.
```

5.Murakkab takrorlanish jarayonlari

Yuqorida keltirilgan operatorlar asosida murakkab takrorlanishlar tashkil qilish mumkin. Agar takrorlanish jarayonlarining tanasi takrorlanish strukturasidan tashkil

topgan bo'lsa, u holda bunday takrorlanish ichma-ich joylashgan yoki murakkab deb ataladi, ya'ni boshqacha qilib aytganda, bitta takrorlanish ichiga bir yoki bir necha boshqa takrorlanishlar kirsa murakkab tarkibli takrorlanishli dasturlar hosil bo'ladi.

Boshqa takrorlanish jarayonlarini o'z ichiga olgan takrorlanish tashqi takrorlanish deb ataladi.

Takrorlanish jarayonlarini ichida joylashgan takrorlanish ichki takrorlanish deb ataladi.

Misol.

1) $\cos(I^3) \sin(I \cdot n)$, $I=1, \dots, n$ ni hisoblash asosida E dan ($E=0.001$) kichik bo'lgan sonlar bo'lsa, «bor», bo'lmasa «yo'q» degan natijani olish dasturi tuzilsin.

Program mur;

Label 1;

Var I,n : integer;

Begin

Read(n);

For I:= 1 to n do

If $\cos(I \cdot I \cdot I) \cdot \sin(I \cdot n) < 0.001$ then

Begin

Write('bor');

Goto 1

End;

Write ('yo'q')

1: end.

VIII. MALUMOTLARNING QAYD QILINGAN, CHEGARALANGAN VA MUNTAZAM TOIFALARI

1. Ma'lumotlarning qayd qilingan toifasi.
2. Ma'lumotlarning chegaralangan toifasi
3. Ma'lumotlarning murakkab toifasi. Muntazam toifa

1. Ma'lumotlarning qayd qilingan toifasi.

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 echilayotgan masalaning mazmuniga ko'ra belgilanadi.

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

TYPE bo'limida:

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 turlar va ularning qiymatlar ro'yxati keltirildi. 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:

Masalan:

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

SEZ, D4: (SUMMY, 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 erda 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);

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

2. Ma'lumotlarning chegaralangan toifasi

Ba'zi bir hollarda o'zgaruvchi o'zi aniqlangan toifaning hamma qiymatlarini emas, balki ma'lum chegara (diapazon) dagi 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 tur nomi>q1-konstanta..2-konstanta;

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

Bu erda 1-konstanta va 2-konstanta mos ravishda baza turining birinchi va oxirgi qiymatlari.

Chegaralangan turdagi o'zgaruvchi baza turining 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 C: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.

3. Ma'lumotlarning murakkab toifasi. Muntazam toifa

Massiv tushunchasini A-umumiy nomga ega va bir toifadagi qo'zg'almas kattaliklar ketma-ketligidan iborat bo'lgan sonli vektor $A(4) = (a_1, a_2, a_3, a_4)$ misolida tushuntirish mumkin.

Bu erda $a_1, a_2, \dots$ - 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 birmuncha pastda ko'rsatilar edi, masalan $A(1)$ yoki A_1 , umumiy holda esa $A(I)$, bu erda $I=1,2,3,\dots,n$.

Paskal tilida o'zgaruvchining ko'rsatkichi to'rtburchak qavslar ichiga olinib yoziladi, ya'ni $A[1]=11.5$, $A[2]=10.5$, $A[3]=-65.7$, $A[4]=7.3$.

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

VAR <massiv nomi>:

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

Bu erda <massiv nomi>-ixtiyoriy identifikator, ARRAY (massiv) va OF (-dan) - xizmatchi so'zlar, < ko'rsatkich toifasi> - ko'rsatkichli ifoda, bu toifa qiymatlari 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 vektorini massiv ko'rinishda dasturda quyidagicha tavsiflash mumkin:

```
VAR A : ARRAY [1..4] OF REAL;
```

Indeks sifatida faqat aniq son emas, qiymati massiv elementining tartib raqamini belgilovchi ifoda ham ishlatilishi mumkin, masalan, A[I+3], B[I DIV (j+3)], C[n1 OR n2], YEAR[2004] va h.k.

Ko'rsatkichli ifoda qiymatining toifasi massiv elementa ko'rsatkichning 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, Turbo Paskal tilida INTEGER va REAL toifasidagi ma'lumotlar to'plami cheklanmagan, REAL toifa esa, shuningdek, 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 [4] OF REAL;
```

```
C: 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, 50 ta haqiqiy elementlardan iborat massiv quyidagi ko'rinishda tavsiflanishi mumkin:

```
VAR A: ARRAY [1..50] OF REAL;
```

Bu erda 1 dan 50 gacha chegaralangan butun toifa A massiv elementlari soni ellikta ekanligi va ularning tartiblanganligini (birinchidan elliginchigacha) bildiradi.

To'g'ri tavsiflangan massivlarga misollar:

VAR MASSIV: ARRAY [-645..-1] OF REAL; YEAR:ARRAY [1999.. 2004]
OF CHAR; L:ARRAY [BOOLEAN] OF CHAR; A,B,C:ARRAY [1..50] OF
REAL;

Ko'rsatkich va ko'rsatkich toifasi tushunchalari o'rtasida o'zaro farq mavjud bo'lib, ko'rsatkich toifasi massiv elementlari soni va ular o'zaro 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(4) massiv uchun A o'zgaruvchi to'liq o'zgaruvchi (massiv nomi), A[1] - ko'rsatkichli o'zgaruvchi bo'lib, u A massivining 1 elementini ifodalaydi.

Massiv elementlarining operatorlar bo'limida ishlatilishiga misollar:

B[4]:= B[3] + 1;

SUM:= SUM - ROUND(C[K]);

P1:=SQRT(A[2*I+1]);

Massivlarning o'lchami ularni kompyuterda qayta ishlashga hech qanday to'sqinlik ko'rsatmaydi, chunki massiv elementlari ularning o'lchamidan qat'iy nazar kompyuter xotirasida chiziqli ketma-ketlik sifatida saqlanadi.

Ko'p o'lchamli massivlar dasturda quyidagicha tavsiflanadi.

Masalan B o'zgaruvchini to'rtta satr va beshta ustundan iborat (4,5) ikki o'lchamli massiv sifatida quyidagicha tavsiflash mumkin

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

Massiv elementlariga qiymat berishda malumotlarni kiritish yoki o'zlashtirish operatorlaridan foydalanish mumkin.

Misol 1.

Matematik kutilma $M = \frac{1}{N} \sum_{i=1}^N y_i$ va dispersiya $D = \frac{1}{N-1} \sum_{i=1}^N (y_i - M)^2$ ni hisoblash dasturi tuzilsin.

Program takr;

```

Const N=10;
Var M, D:real; i,N: integer; Y: array [1..N] of real;
Begin
    M:=0; For i:=1 to N do
begin
    Read (y[i] );
    M:=M+y[i]; M:=M/N ;
End;
    D:=0;
    For i:=1 to N do
Begin
    D:=D+sqr(y[i]-M); D:=D/ (N-1);
End;
    Write ('matematik kutilma M=',M, 'dispersiya D=',D)
End.

```

Misol 2.

Regressiya tenglamasidagi noma`lum koeffitsientni topish dasturi tuzilsin.

$$K = \frac{\sum_{i=1}^N y_i x_i}{\sum_{i=1}^N x_i^2} \quad N=8$$

```

program koef;
const N=8;
var K,S,F : real; I,N: integer; x, y: array [1..N] of real;
begin
    S:=0; F:=0;
    for I:=1 to 8 do begin
    read(y[I],x[I]); S:=S+y[I]*x[I];
    F:=F+sqr(x[I]); end;
    K:=S/ F;
    Write('K ning qiymati =', K)
End.

```


IX .FUNKTSIYA VA PROTSEDURALAR TUSHUNCHASI

1. Turbo Paskal tilida protsedura konstruktsiyasi.
2. Funktsiya qism dasturi haqida
3. Protsedura va funktsiya qism dasturiga murojaat
4. Protsedura va funktsiya qism dasturlarining parametrlari

1. Turbo Paskal tilida protsedura konstruktsiyasi.

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 qismlar vujudga keladi.

Kompyuter xotirasini va dastur tuzuvchining vaqtini tejash maqsadida ushbu qismdagi o'xshash operatorlar bir marta asosiy dasturdan ajratib yoziladi va unga asosiy dastur bajarilishi jarayonida murojaat qilinadi.

Dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatish mumkin bo'lgan bunday operatorlar guruhiga qism dastur deb ataladi va u asosiy dastur bilan bir butunni tashkil etadi.

Qism dasturlarni ishlatish dasturning hajmini kichraytiradi va uning ko'rinishini, o'qilishini hamda xatolar sonining kamayishiga olib keladi.

Turbo Paskal tilida qism dasturlar protsedura va funktsiya qism dasturi ko'rinishida bo'ladi.

Protsedura qism dasturi qism dasturda bir necha natija hosil qilish kerak bo'lgan hollarda ishlatiladi va u asosiy dasturda struktura bo'yicha qism dastur bo'limida, ya'ni o'zgaruvchilar va operatorlar bloki bo'limlari orasida tasvirlanadi.

Har bir protsedurani tasvirlash sarlavhadan boshlanadi. Bunda protseduraning nomi va parametrlarning ro'yxatlari ularning turlari bilan birga beriladi.

Protsedura parametrsiz ham bo'lishi mumkin. Bu holda protsedura sarlavhasida faqat uning nomi ko'rsatiladi. Parametrlar yordamida belgilarni protseduraga berish hamda hosil bo'lgan natijani protsedurani chaqirayotgan dasturga berish global parametrlar yordamida amalga oshiriladi.

Protsedura sarlavhasining umumiy ko'rinishi quyidagicha bo'ladi:

PROCEDURE < nom > (< sohta parametrlar ro'yxati >);

bu erda PROCEDURE - xizmatchi so'z; <nom>- protseduraning nomi; <sohta parametrlar ro'yxati>- turlari ko'rsatilgan berilganlarni va natijalarni ifodalaydigan nomlar turlari bilan sanab o'giladigan qator.

Sohta parametrlar ro'yxatida parametr-qiymat, parametr-o'zgaruvchilar (bularning oldida VAR xizmatchi so'zi turishi kerak), parametr-protseduralar (bularning oldida PROCEDURE xizmatchi so'zi turishi kerak) va parametr-funksiyalar (bularning oldida FUNCTION xizmatchi so'zi turishi kerak) bo'lishi mumkin.

Protsedura sarlavhasidan keyin keladigan bo'limlar asosiy dasturda qanday tartibda kelsa, protsedurada ham shu tartibda keladi.

Protsedura bloki nuqta vergul (;) bilan tugaydi.

Misol: Berilgan x va y sonlari uchun $U = \text{MAX}(x + y, x * y)$;

$V = \text{MAX}(0.5, U)$ topilsin.

Program max;

Var x, y, u, v, s: real;

Prosedure *prosed* (p1, p2: real);

Begin

If p1 > p2 **then** s := p1 **else** s := p2

End;

Begin

Write ('x = '); **readln** (x);

Write ('y ='); **readln** (y);

Prosed (x + y, x * y); u := s;

Prosed (0.5, u); v := s;

Writeln (u:3:1, v);

End.

2. Funktsiya qism dasturi haqida

Funktsiya bu shunday qism dasturki, bunda bitta natija chiqadi va u funktsiya qism dasturining nomida hosil bo'ladi. Shuning uchun funktsiya qism dasturi protsedura qism dasturining xususiy holi bo'lib, undan farqi shundaki, birinchidan, funktsiya qism dasturining natijasi bitta qiymatdan iborat bo'ladi.

Protsedura qism dasturida natija esa bitta yoki bir necha bo'lishi mumkin. Funktsiya qism dasturining natijasi asosiy dasturga funktsiyaning natijasi sifatida beriladi.

Protsedura qism dasturi natijalari esa uning parametrlari qiymati sifatida hosil bo'ladi.

Funktsiya qism dasturini aniqlanishi protsedura qism dasturini aniqlanishiga o'xshaydi va u ham sarlavha va blokdan iborat bo'ladi.

Funktsiya qism dasturining umumiy ko'rinishi quyidagicha bo'ladi:

Function <nom> (<sohta parametrlar ro'yxati>): <tur>;

bu erda function - xizmatchi so'z, <nom> - funktsiyaning nomi, <sohta parametrlar ro'yxati> - kiritiladigan sohta parametrlar sanab o'tiladi, <tur> - funktsiya natijasining turi.

Bunda natija funktsiya nomida hosil bo'ladi. Bundan tashqari, funktsiyaning parametrsiz ko'rinishini ham yozish mumkin:

Function <nom>: <tur>;

Har ikkala holda ham funktsiya natijasi funktsiya qism dasturi tanasida funktsiya nomiga hech bo'lmaganda bir marta o'zlashtirilishi shart.

Misol: Ikki o'zgaruvchi faktorialning ayirmasini xisoblash dasturi funktsiya qism dasturi yordamida amalga oshirilsin.

$F = m! - n!$

```
program factorial;  
var f,m,k,:integer;  
function fakt (n: integer): integer;  
var p,i:integer;  
begin
```

```

p:=1;
for i : =2 to n do
p:=p*i;
fact := p;
end;

begin

readln (m,k);
f := fact(m)-fact(k);
writlen ('f='; f);

end .

```

3. Protsedura va funktsiya qism dasturiga murojaat

Protsedurani bajarish uchun asosiy dasturning ixtiyoriy (kerakli) joyidan unga murojaat qilish kerak.

Protseduraga murojaat va uni bajarish quyidagi operator yordamida amalga oshiriladi:

```
<protsedura nomi>(<aniq parametrlar ro'yxati>);
```

Bunda qism dastur tasvirida keltirilgan protseduraga murojaat qilishda sohta va aniq parametrlar orasida to'liq moslik bo'lishi shart, ya'ni sohta va aniq parametrlar soni bir xil bo'lishi, kelish tartiblari mos tushishlari va ularning turlari bir xil bo'lishi kerak.

Protsedura chaqirilganda sohta parametrlar o'rniga aniq parametrlar mos qo'yiladi.

Bunda parametr-qiymatlar o'rniga kiritiladigan malumotlar beriladi. Natija esa parametr-o'zgaruvchilarda hosil bo'ladi.

Protsedura chaqirilganda sarlavhada ko'rsatilgan sohta parametrlar aniq parametrlar bilan ularning kelishi tartibida almashtiriladi, ya'ni birinchi sohta parametrqa birinchi aniq parametr, ikkinchi sohta parametrqa ikkinchi aniq parametr va h.k. mos qo'yiladi.

Funktsiya qism dasturiga murojaat uning nomi orqali amalga oshiriladi. Funktsiyaga murojaat qilinadigan joyda funktsiyaning nomi yoziladi va undan keyin qavs ichida aniq parametrlar yoziladi.

Agar funktsiya qism dasturi parametrsiz ko'rinishda bo'lsa, u holda faqat uning nomi yoziladi.

4. Protsedura va funktsiya qism dasturlarining parametrlari

Protsedura va funktsiya qism dasturlari aniqlanayotganda ularning sohta parametrlar ro'yxatida parametrlarning to'rt xil ko'rinishi ko'rsatilishi mumkin:

- 1) parametr- qiymatlar;
- 2) parametr- o'zgaruvchilar;
- 3) parametr- protseduralar;
- 4) parametr- funktsiyalar.

Sohta va aniq parametrlar protsedura va funktsiya qism dasturlarida bir xil qayta ishlanadi.

Har bir parametr turi ko'rsatilgan holda ro'yxatga kiritiladi.

Masalan:

```
PROCEDURE SUMMA(X,Y,Z: REAL);  
PROCEDURE SUMMA1(A,B: REAL; N: INTEGER);  
PROCEDURE TH(VAR A,R: REAL);  
FUNCTION SS1(B: REAL; I,J: INTEGER): REAL;
```

Agar ikkita yoki undan ko'p sohta parametrlar ro'yxatda bir xil turda bo'lsa, u holda ularni bitta tasvirlashga kiritish mumkin. Sohta parametrlarni ro'yxatda ixtiyoriy tartibda ko'rsatish mumkin. Protsedura va funktsiya qism dasturlariga murojaat qilinayotganda sohta parametrlar qanday tartibda kelsa, aniq parametrlar ham shu tartibda kelishi kerak.

Protsedura va funktsiya qism dasturlarini o'z ichiga olgan asosiy dasturda nom tanlanayotganda lokal va global parametrlarning bir xil bo'lmagani maqul.

Parametr-qiymatlar parametrlari berilganlarni qism dasturlar (protsedura va funktsiya) ga uzatish uchun ishlatiladi.

Bunda yuqorida aytilganidek, ular sohta parametrlar ro'yxatida turlari ko'rsatilgan holda, bir-birlaridan vergul bilan ajratilgan holda sanab o'tiladi.

Mavzuni mustaxkamlash uchun topshiriqlar

1. S, t berilgan $S(a,b,c)=\frac{2a-\sin C}{5+(C)}$ yordamida $S(t,-25,1.17)+(2.2,t,S-t)$ funktsiya

xisoblansin.

2. S, t berilgan $D(a,b)=\frac{a^2+b^2}{a^2+2ab+3b^2+4}$ yordamida $D(1,2,5)+d(t,s)-d(2.5,St)$

funktsiya xisoblansin.

3. y berilgan. $T(X)=\sum_{k=0}^{10} \frac{x^{2k+1}}{(2k+1)}$ yordamida $\frac{1.7t(0,25)+2t(1+y)}{6-t(y^2-1)}$ funktsiya

xisoblansin.

4. a,v,c berilgan. $\frac{\max(a,a+b)+\max(a,b+c)}{1+\max(a+bc,1,15)}$ xisoblansin.

5. S, t berilgan. $H(a,b)=\frac{a}{1+b^2}+\frac{b}{1+a^2}-(a-b)^3$ yordamida

A). $H(S,t)+\max(h^2(S-t,S,t))$ xisoblansin.

B). $H^2(S-t,S+t)+h(1,1)$ xisoblansin.

6. a, b berilgan. $N=\min(a,b)$ $v=\min(a,b,a+b)$ $S=\min(n+v^2,3.14)$ xisoblansin

7. a, b berilgan. $N=f(0,5,a)+f(a,b)+f(a+b)$ xisoblansin.

bu erda $f(x,y)=\frac{x^2+xy-y^2}{1+x^2+y^2}$

8. Butun tipli m va xaqiqiy tipli x,y,z [1; m] vektorlar berilgan.

$n=\sum_{i=1}^m x_i$ $v=\sum_{i=1}^m y_i$ $W=\sum_{i=1}^m Z_i$ ni xisoblash dasturini tuzing.

9. n, m sonlar berilgan

fact = 1*****2 N funktsiyadan foydalanib $\frac{m!+n!}{(m+n)}$ ni xisoblash dasturini tuzing.

10. $M=\frac{1}{n}*\sum_{i=1}^n a_i$ - matematik kutilma va $D=\frac{1}{n-1}\sum_{i=1}^n (a_i-M)^2$ - dispersiyani xisoblash

protsedurasini yozing.

11. Quyidagi qism lavhada:

Type Msat =1 .. K;

Var m, n : Msat;

```

Function EKB (a,b:Mast) : Mast;
Begin write a< > b do
If a>b then a := a-b
Else b := b - a
EKB:= a end;
M:= 8; n:=6 ; WriteLn (EKB (m,n),m,n) ni toping.

```

12. Quyidagi dasturlar lavhalarida hatto toping:

a) Function a (κ : integer) ;

Var i, s :

Begin S:= 0;

For i:= 1 to κ do s:=s + sqr (i) end;

b) Function b (x : integer) :integer;

Begin b (x):= (sqr(x)+x)/2 end

13. Quyidagi dastur lavhasida xato toping:

Procedute Msat (z,y,a,x) : real;

Begin

Z :=Sin (sqr(exp(x+x+a)));

Y := Ln (sqrt(abs(1-z*cos(2))))

End

X. MA'LUMOTLARNING ARALASH TOIFASI. YOZUVLAR.

1. Dasturda yozuvlarni tavsiflash
2. Yozuv elementlari va ular ustida bajariladigan amallar
3. Birlashtirish operatori

Amaliyotda iqtisod va axborotni qayta ishlash masalalarini echishda ma'lum turdagi xujjatlar, kataloglar, ro`yxatlar, vedomostlar ishlatiladi.

Masalan, talabalarning anketa ma'lumotlari: familiyasi, ismi, otasining ismi, turar joyi, tug`ilgan yili, mutaxassisligi, gurux raqami va xokazo. Bu xollarda turli toifadagi ma'lumotlarni bir guruxga birlashtirish zaruriyati tug`iladi.

Bizning misolimizda bu ma'lumotlarni talaba guruxiga birlashtirish mumkin. Ko`rinib turibdiki, bu guruxdagi ma'lumotlarning toifalari turlicha: familiya, ism-qator(so`z), tug`ilgan yili, gurux 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.

1. 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 aloxida-aloxida beriladi.

Element nomi va uning tavsifi keltirilgan qator yozuv maydonini tashkil etadi.

Demak, yozuvlar bir necha maydondan tarkib topgan bo`lishi mumkin.

Yozuvlarni tavsiflashda quyidagi qoidalardan foydalaniladi:

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

`<maydonlar ro`yxati> ::= <yozuvlar bo`limi>;`

`<yozuvlar bo`limi> ::= <maydon nomi> { , <maydon nomi> } : <toifa>;`

Xar bir ob'ektga nom berilgani kabi yozuvlarga va uning elementlariga xam 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; p-element nomi : n-element toifasi END; VAR <yozuv nomi, ya'ni aralash toifadagi o`zgaruvchi>: toifa nomi;

Masalan:

TYPE ZAPIS=RECORD

FAM: string[15]; GODROG: integer; GODPOST: integer; NOMGR: integer
END;

VAR STUD : ZAPIS;

Bu erda STUD nomli yozuv 4 ta elementdan iborat: FAM qator toifadagi, GODROG, GODPOST, NOMGR butun toifadagi qiymatlarni qabul qilali.

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; p-element nomi : n-element toifasi END;

Masalan:

VAR STUD : RECORD

FAM: string[15];

NOMGR: integer;

BAHO : array[1..3] of integer

END;

2. Yozuv elementlari va ular ustida bajariladigan amallar

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

Dasturda yozuv elementlari quyidagicha ifodalanadi:

yo'zuv nomi. element nomi.

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

Bu erda STUD - yozuv toifasidagi o'zgaruvchi nomi, NOMGR, 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 xolatda 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 elementi yozuvdan iborat.

Tavsif asosida quyidagi o'zlashtirish operatorini yozish mumkin:

```
ISM.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.

M a s a l a. Talabalar xaqida quyidagi ma'lumotlar berilgan: Talabaning familiyasi, tug'ilgan yili, dorilfununga kirgan yili, gurux raqami. Shu ma'lumotlar asosida 51-guruxda o'qiydigan talabalar ro'yxati bosmaga chiqarilsin.

Masalani echish dasturi quyidagicha:

```
PROGRAM RAN;
TYPE ZAPIS=RECORD
FAM:STRING[15]; TUGYIL:INTEGER; KIRYIL:INTEGER;
GRTAR:INTEGER END; VAR STUD: ZAPIS; GR:ARRAY[1..10] OF
ZAPIS;
I,J:INTEGER;
BEGIN WRITELN('TALABALAR SONINI KIRITING'); READLN(N);
FOR I:=1 TO N DO BEGIN
```

```

READLN (STUD.FAM);
READLN ( STUD.TUGYIL,STUD.KIRYIL,STUD.GRTAR); GR[I]:=STUD;
END;
WRITELN(' 51- GUPUH TALABALARI'); FOR I:=1 TO N DO
IF GR [I].GRTAR=51 THEN WRITELN (GR[I].FAM,' % GR[I].GRTAR)
END.

```

3. 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`lyapti. Buning evaziga dastur xajmi xam oshib ketadi. Shu noqulayliklarni bartaraf qilish maqsadida Paskal tilida maxsus operator - birlashtirish operatori kiritilgan.

Birlashtirish operatori yordamida yozuv nomi sarlavxaga chiqariladi va elementga murojaat qilishda faqat elementning nomi ko`rsatiladi, xolos.

Operatorning sintaksis tavsifi quyidagicha:

<Birlashtirish operatori>::=<sarlavxa><operator>;

<sarlavxa>::= WITH <o`zgaruvchilar ro`yxati> DO

<o`zgaruvchilar ro`yxati>::=<yozuv toifasidagi o`zgaruvchi>{, <yozuv toifadagi o`zgaruvchi>} yoki operatorning umumiy ko`rinishi quyidagicha:

WITH F DO O

Bu erda 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 lavxasini 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 xaqida quyidagi ma'lumotlar berilgan: Talabaning familiyasi va matematika, informatika, fizika fanlaridan olgan baxolari.

Shu ma'lumotlar asosida qarzidorlar ro'yxati tayyorlansin. Qarzdor talaba deb xech bo'lmaganda bitta fandan 2 baxo olgan talaba xisoblansin.

Masalani echish dasturi:

```
PROGRAM FF; TYPE ST = RECORD
FAM: STRING[15]; BAHO= RECORD
MAT,INF,FIZ : INTEGER END END;
VAR GRUP: ARRAY[1..5] OF ST; I,J: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;
write ( 'qarzdorlar ro'yxati');
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.

Mavzuni mustaxkamlash uchun topshiriqlar

1. Quyidagi tushunchalarni anglatadigan aralash turlar yozilsin.

- a) narx – so'm va tiyinda;
- b) vaqt - soat, minut va sekundlarda;
- v) sana - son, oy, yil;
- g) manzil - shaxar, ko'cha, uy, xona;
- d) seminar - fan, o'qituvchi, gurux nomeri, hafta kuni, mashg'ulot soati, auditoriya;

2. type rasm= (piki, trefi,bubni, chervi);

nom=(oiti, etti, sakkiz, to`qqiz, on, valet ,dama, qirol, tuz);

karta= record m: rasm; d: nom end;

KR rasmlı (rasm) karta (karta) zot bo`lganda K1 karta K2 kartani urishi yoki yo`qligini aniqlovchi mantiqiy uradi (K1,K2,KM) protsedura-funktsiyasi tuzilsin.

3. tupe choqqi= record nomi: string (5);

balandligi: 1000..9999

end;

ruyxat= array / 1..30 / of choqqi;

S ruyxatdagi (ruyxat) eng baland cho`qqining nomini chop qiluvchi eng baland (S) protsedurasi tuzilsin.

4. Quyidagi savollarga javob berilsin.

a) Yozuvni barcha maydonlari xar xil turda bo`lishi shartmi?

b) Nima uchun turni aniqlayotganda uning maydoni ixtiyoriy tartibda sanab o`tiladi?

c) Yozuvdagi maydon nomi o`zgaruvchi, o`zgarmas va programmadagi boshqa ob'ektlar nomi bilan ustma-ust tushishi, biroq boshqa yozuvdagi maydon nomi bilan ustma-ust tushmasligi to`grimi?

5. Quyidagi tushunchalarni jadval kattalik yoki yozuv ko`rinishida, agar mumkin bo`lsa, ikkala ko`rinishda xam yozilsin.

a) shaxmat taxtasi maydonining belgilanishi (a5,h8 va x.k.);

b) kompleks son;

c) 50 o`lchovli fazodagi nuqtani.

6. tupe manzil= record shahar, kucha: string|8|;

uy, hona: 1..999 end;

var manzil _1, manzil_2: manzil;

Birlashtirish operatoridan foydalanib, manzil_1 o`zgaruvchiga «Toshkent, Do`stlik ko`chasi, 1 uy, 5-xona» manzil qiymati berilsin. Undan tashqari, manzil\_1 qiymatining, faqat uy nomerini 17 soniga o`zgartirilgani manzil_2 o`zgaruvchiga o`zlashtirilsin.

7. Maktab o'quvchisi xaqidagi ma'lumotni o'z ichiga olgan aralash tur e'lon qilinsin. Unda o'quvchining familiyasi, ismi va sharifi, yoshi, maktab nomeri, nechanchi sinfda o'qishi va qandaydir beshta fandan olgan baxolari berilgan bo'lsin.

Mazkur turdagi o'zgaruvchi e'lon qilinib, unga mos ravishda quyidagi qiymatlar berilsin: Qodirov Hasan Olimovich, 16 yosh, 111-maktab, 9-sinf, baxolari-5, 3, 4, 5, 2.

```
8. Tupe complex= record re, im: real end;
point= record x,y : real end;
var z,w: complex; p:point; re: real;
z,w,p va re o'zgaruvchilari quyidagi operatorlar ketma-ketligi bajarilgandan so'ng
qanday qiymatlar qabul qilishi aniqlansin.
with z do begin re:=0; im:= 1 end;
w:=z; re:=2;
with z do re:=1;
with z,w do im:=-im;
with p do begin x:=re; y:=2 end;
```

XI. TO`PLAMLAR. TO`PLAM USTIDA BAJARILADIGAN AMALLAR

1. Turbo Paskal tilida to`plam tushunchasi.
2. To`plamlarni tavsiflash
3. To`plam elementlari ustida bajariladigan amallar

1. Turbo Paskal tilida to`plam tushunchasi.

To`plam deganda, bir necha elementlarning majmuasi tushuniladi. Bu elementlar bir xil toifali, lekin tartiblanmagan bo'ladi.

Masalan, butun sonlar to`plami, shakllar to`plami, radiodetallar to`plami va xokazo.

Turbo Paskal tilida to`plam deb bir toifadagi cheklangan soxadagi ma'lumotlarning betartib majmuasiga aytiladi. Xar bir to`plamga nom beriladi. To`plamga kirgan ma'lumotlar to`plam elementlari deb yuritiladi. Elementlar turi baza toifasi deyiladi. Baza toifasi sifatida REAL va cheklanmagan INTEGER

toifasidan tashqari barcha standart, qayd qilingan va chegaralangan toifalarni olish mumkin.

XII. MA'LUMOTLARNING FAYLLI TOIFASI. TOIFALASHGAN VA TOIFALASHMAGAN FAYLLAR

1. Turbo Paskal algoritmik tilida fayl tushunchasi
2. Toifalashgan fayllar va ular bilan ishlash
3. Toifalashmagan fayllar

1. Turbo Paskal algoritmik tilida fayl tushunchasi

Sizga ma'lumki, fayl deb xotiraning nomlangan soxasiga aytiladi. Faylda turli ma'lumotlar saqlanadi. Xar bir fayl bilan fayl ko'rsatkichi degan tushuncha biriktirilgan. Fayl bir necha elementlardan iborat bo'lib, foydalanuvchi faqat faylning ko'rsatkichi ko'rsatayotgan ma'lumotga murojaat qilishi mumkin.

Demak, fizik jixatdan biz faqat ketma-ket fayllarga egamiz. Ya'ni biz oldin birinchi, keyin ikkinchi, uchinchi va x.k. ma'lumotlarni o'qishimiz mumkin.

Fayl o'z nomiga ega. Masalan, d:tp \ myfile.dat

Turbo Paskal tili dasturiy vositalari yordamida, ya'ni dasturda xam fayllar tashkil qilish va undagi ma'lumotlarni qayta ishlash mumkin. Shu paytga qadar, Paskal dasturiy tilida bir necha o'zgaruvchilarning toifalari bilan ishlab keldik. Bular skalyar, oddiy va murakkab tarkiblashgan toifalardir. Bu toifadagi ma'lumotlar yordamida masalalarni echishda boshlang'ich ma'lumotlar klaviaturadan operativ xotiraga kiritiladi va natija ekranga chiqariladi. Ulardan boshqa dasturlarda foydalanib bo'lmaydi, chunki ular tizimidan chiqilgandan so'ng xech qaerda saqlanmaydi. Bu ma'lumotlarni xotirada saqlash uchun Turbo Paskal tilida ma'lumotlarning faylli toifasi belgilangan. Fayl toifasi aloxida o'rin egallaydi.

Fayl toifasi bilan ishlashda ma'lum tushunchalarni o'zlashtirish talab qilinadi. Birinchidan, fayllar toifasi nega va qachon qo'llaniladi? Maqsad nima? Zaruriyat nimadan kelib chiqyapti? Ikkinchidan, boshqa toifalardan nega katta farqi bor? Bu savollarga faqat foydalanuvchining nuqtai nazaridan qaragan xolda javob bera olamiz:

1. Juda ko'p o'zgaruvchilardan foydalanganda ularning qiymatlarini xar doim klaviaturadan 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 (masalan, natijalar), bu kattaliklarni faylga yozish maqsadga muvofiq.

3.Xech qanday toifalar tashqi qurilmalarga murojaat qilib, ular bilan ishlashga imkon yaratmaydi (dasturiy til muxitida).

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

Masalan:

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

bu toifalar ichida fayl toifasini ishlatish mumkin emas.

Fayllarning turlari. 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 xam mashina kodlari bilan yozilgan bo'lib baytlar to'plamini tashkil qiladi.

Matnli (tekst) fayllar ASCII kodlardan tashkil topgan va qatorlarga ajratilgan bo'ladi. Matnli fayllarda nafaqat faylning yakunida fayl oxiri belgisi, balki xar qatorning yakunida maxsus qator oxiri belgisi qo'yiladi.

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

Fayl turi uchun arifmetik amallar belgilanmagan. Xatto fayllarni solishtirish va bir faylning qiymatini ikkinchi faylga o`zlashtirish amallari xam aniqlanmagan.

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

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

Bu aloxida protsedura yordamida amalga oshiriladi:

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

Bu erda 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 murojaat 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`rsatkishi faylning boshiga o`rnatiladi. Lekin bu protsedurani extiyotkorlik 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. Buning uchun quyidagi protsedura ishlatiladi:

Reset (<fayl o`zgaruvchisi>);

Reset protsedurasi bajarilganda Assign protsedurasida ko`rsatilgan fayl undagi ma'lumotlarni o`qish uchun tayyorlanadi, ya'ny fayl ko`rsatkishi 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 erda kattalik o`rnida o`zgaruvchi yoki ifoda ishlatilishi mumkin. Protsedura bajarilganda kattalikning qiymati fayl o`zgaruvchisi bilan bog`langan faylda fayl ko`rsatkishi o`rnatilgan joyga yoziladi. So`ngra fayl ko`rsatkishi 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`rsatkishi 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`rsatkishi faylning oxiriga o`rnatilganda True qiymatiga ega bo`ladi, aks xolda uning qiymati False ga teng. 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.

2. 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 turi>::= <toifa>

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

Toifalashgan fayllarni xar doimgidek, Ture 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 xam, masalan, yozuvlarni ishlatish mumkin:

```
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 qatorida yana qo'shimcha protsedura va funktsiyalarni ishlatish mumkin. Ba'zi bir protseduralar faylni bevosita murojaat fayli sifatida ishlatish imkonini beradi.

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

2. FilePos(<fayl o'zgaruvchisi>):integer; -bu funktsiya joriy elementning fayldagi o'rnini aniqlab beradi, joriy element deb fayl ko'rsatkishi o'rnatilgan elementga aytiladi.

3. Seek(<fayl o'zgaruvchisi>,<elementning n-tartib raqami>); protsedurasi fayl ko'rsatkishini 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 Fl ; var f: file of char; ch: char;

i: integer ;

begin

assign(f,'myfile.dot');

rewrite(f);

for i:= 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.dot 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; I:integer; Procedure SdF;

begin

Reset(f);

for I:=1 to FileSize(f) do

```

begin
read(f,ch) ; write (ch,',')
end;
Close(f);
End;
Begin Assign(f,'Chfile. txt');
Rewrite(f);
FOR I:=1 to 10 do begin
Read(c); Write(f,c) end; Sdf; writeln('fayl') Close(f);
End.

```

Matnli fayllar va ular uchun mo`ljallangan protsedura va funktsiyalar.

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

- a) ma'lumot matn shaklida ASCII kodlar jadvalining simvollarida tasvirlanadi;
- b) ma'lumotlar qatorlarga bo`linishi mumkin;
- c) sonlar, mantiqiy qiymatlar, qatorlar Char toifasidagi ma'lumotlarga aylantiriladi va mashina kodlariga yoziladi.

Dasturda matnli fayllar Text xizmatchi so`zi yordamida beriladi:

<fayl o`zgaruvchisi>: Text;

Turbo-Paskal muxitida ikkita matnli fayl o`zgaruvchilari standart ravishda aniqlangan. Bu INPUT va OUTPUT o`zgaruvchilaridir. 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 qatorini o`qish protsedurasi. Bu protsedura bajarilganda fayl ko`rsatkichi o`rnatilgan qator o`zgaruvchiga o`zlashtiriladi, faylda qator «qator oxiri belgisi» yordamida ajratib olinadi.

2. `Writeln(<fayl o`zgaruvchisi>,<qator>);` - faylga simvollar qatorini yozish protsedurasi. Protsedura bajarilganda fayl ko`rsatkichi o`rnatilgan joyga <qator> yoziladi.

3. `Append(<fayl o`zgaruvchisi>);` protsedurasi. Bu protsedura faylni unga qo`shimcha 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 qatorning oxirini aniqlaydi. Funktsiyaning toifasi Boolean ( mantiqiy) bo`lib, fayl ko`rsatkichi qator oxiriga o`rnatilganda funktsiyaning qiymati True ga, aks xolda False ga teng bo`ladi.

5. `SeekEoln(<fayl o`zgaruvchisi>);` funktsiyasi qator yakuniga erishilganini aniqlaydi.

6. `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.

M i s o l. Matnli faylning qatorlarini bosmaga chiqarish.

```
Program f5;  
var ftl:text; fln:string;  
Begin Assign(ftl,'d:\myfile.txt');  
Reset (ftl);  
While NOT eof(ft1) do  
Begin readln(ftl,fln); writeln(fl n)  
End;  
Close (ftl)  
End.
```

3. Toifalashmagan fayllar

Turbo-Paskal dasturiy tilida aloxida 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, 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. Shundan kelib chiqqan xolda bu ko'rinishdagi fayllarga nisbatan quyidagi protsedura va funktsiyalarni ishlatish mumkin:

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

2. BlockRead (F,V,N); protsedurasi, bu erda 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 lavxada F faylini ochib unga uchta blok ma'lumotlarni yozishga yordam beradi:

Assign (F,'ABC.dat'); Rewrite (F,size); BlockWrite(f,a,3); Close(F);

Bu ma'lumotlarni fayldan o'qish quyidagi lavxa 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.

Namunaviy dastur:

1) program mas;

uses CRT;

var f: Text; S: string; R: extended; i,j,k,m,N: integer; X,Y: array[1..1000] of extended;

BEGIN

ClrScr;

ReadLn(S); if S="" then begin ReadLn(R); ReadLn(N);

for i:=1 to N do begin

ReadLn(X[i]); ReadLn(Y[i]); end; end else begin Assign(f , S+ '.pas'); Reset(f);
ReadLn(f); ReadLn(R); i:= 0; while not eof(f) do begin inc(i); ReadLn(f, x[i],y[i]);
end; N:= i; Close(f); end; WriteLn('N=', N , ' R=', R:0:0); m:=0; for i:= 1 to N-2 do
for j:= i+1 to N-1 do for k:= j+1 to N do if (SQRT(sqr(X[i]) + sqr(Y[i])) <= R) and
(SQRT(sqr(X[j]) + sqr(Y[j])) <= R)and(SQRT(sqr(X[k]) + sqr(Y[k])) <= R) then
begin inc(m); WriteLn(m,' : [('X[i]:0:0,',Y[i]:0:0,') , '+'('X[j]:0:0,',Y[j]:0:0,') , '+
'('X[k]:0:0,',Y[k]:0:0,')]'); end;WriteLn('< OK >'); ReadKey; END.

XIII. MA'LUMOTLARNING MUROJAAT TURI. KO`RSATKICHLAR

1. Ko`rsatkichlarni tavsiflash

Biz shu vaqtgacha ishlatib kelgan o`zgaruvchilar, ya'ni tavsiflash bo`limining Var bo`limida tavsiflangan barcha o`zgaruvchilar statik o`zgaruvchilar bo`lib, bu o`zgaruvchilarga dastur bajarilishiga qadar, kompilyatsiya vaqtida ularning turiga 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 echishda, xamda kompyuterning tovush va grafik imkoniyatlarini ishlatishda xotira xajmi etmasligi mumkin.

Bunday xollarda dinamik xotiradan foydalanish maqsadga muvofiqdir. Dinamik xotira bu shaxsiy kompyuterning dasturga ma'lumotlar segmentidan tashqari yuklatilgan tezkor xotirasidir. Bu xotira taxminan 200-300 kbaytni tashkil qiladi. Shuning uchun katta xajmdagi ma'lumotlarni dinamik xotiraga joylashtirish yaxshi samara beradi.

Dinamik xotiradan foydalanish uchun ma'lumotlarning maxsus turi- murojaat turi 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 statik o`zgaruvchilar singari biror nom bilan murojaat qilib bo`lmaydi.

Bundan tashqari, bu turdagi o`zgaruvchilar dasturning bajarilishi davomida ko`rsatilishi va yo`qotilishi mumkin.

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

1. Amaliyot tizimi va qobiqdar.
2. Paskal kutubxonalari.
3. Interfeys, taxrirlagich, kompilyatorlar.
4. Xatolar xaqida ma'lumot beruvchi fayl.
5. Boshlang`ich modul.
6. Ob'ekt kodi.
7. Dinamik xotira.
8. Rekursiv stek.
9. Markaziy protsessor steki.

Kompyuter xotirasining 1-6 qismlari avtomatik ravishda shakllandi va amaliyot tizimini, kutubxona, funktsiya va modullarini, taxrirllovchi, 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 steki maksimal qo`llash mumkin bo`lgan adres bilan boshlanadi. Ko`rsatkich doimo adres kamayib borish tomoniga siljib boradi. Dinamik xotiraning boshlang`ich nuqtasi «adres» Heap org deb, oxiri - Heap end deb ataladi.

Yuqorida ta'kidlab o`tganimizdek, ko`rsatkich ma'lum turdagi 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 turdagi ma'lumotlar yoki turli turdagi ma'lumotlar yozilgan adreslarga murojaat qilishi mumkin. Shunga ko`ra ular turli va tursiz ko`rsatkichlarga bo`linadi.

Turli ko`rsatkichlar quyidagicha tavsiflanadi:

<Ko`rsatkich> : ^<tur>;

bu erda :^ - ko`rsatkich belgisi, <tur>- ko`rsatkich murojaat qilayotgan tur nomi yoki turning 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 erga dasturning bajarilishi davomida mos turning ixtiyoriy qiymatini joylashtirish mumkin.

Ko`rsatkichlar tursiz bo`lishi xam mumkin. Bu xolda ular yordamida strukturasi oldindan noma'lum bo`lgan va dastur bajarilishi davomida o`zgaradigan kattaliklarni xotiraga dinamik joylashtirish mumkin.

Shunday qilib, Turbo-Paskalda ko`rsatkichni aniq bir turga bog`lamasdan e'lon qilish mumkin. Bu esa maxsus - Pointer so`zi yordamida amalga oshiriladi:

<Ko`rsatkich>: Pointer;

Masalan:

Var k: Pointer;

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

Toifali ko`rsatkichlar ustida quyidagi amallarni bajarish mumkin:

1) NEW(<Ko`rcatkich>) protsedurasi - yangi turlashgan 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 murojaat 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 xolda 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.

```
type a=(a1,a2);  
rec=record  
h:integer;  
case:a of  
al(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 etarli bo`ladi, xaqiqiy o`zgaruvchi uchun esa ikkita.

Masalan, NEW(p)-p[^] dinamik o`zgaruvchi uchun 2002 ta so`z ajratadi:

NEW(p,a1)-p[^] uchun 3 ta so`z, NEW(p,a2)-p<sup>^</sup> uchun 2002 ta so`z.

2) DISPOSE(<ko`rsatkich>) protsedurasida - turlashgan 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,

R: DISPOSE(p)

NEW protsedurasida singari DISPOSE protsedurasida xam variant qismli parametrdan iborat bo`lishi mumkin.

Bu xolda 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.

```

Tursiz ko`rsatkichlar uchun qo`llaniladigan protsedura va funktsiyalar.

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

Bu erda 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 xajmda dinamik xotira ajratiladi.

Masalan,

GETMEM (p:1000) dinamik o`zgaruvchi uchun 1000 bayt joy ajratilgan;

2) FREEMEM (p:pointer; size:word);

Bu protsedura SIZE xajmdagi 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 xajmini aniqlaydi;

5) MAXAVAIL: Longint funktsiyasi eng katta bo`sh maydonni tanlab, uning xajmini ko`rsatadi. Ikki turdagi turlar uchun NIL konstantasi belgilangan. Const NIL tursizdir, uning qiymatini va so`zini ixtiyoriy ko`rsatkichga berish mumkin.

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

2. Ko`rsatkichlar ustida taqqoslash amallari - (=, < , > , < >) ni qo`llash mumkin.

Amalning ikki chetidagi operandlar bir xil turda 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 echish imkonini beradi. Ko`rsatkichlar xali tavsiflanmagan kattaliklar turiga murojaat qilishi mumkin:

Type ppointer = PRecord

PRecord=record

Name: string;

Job: string;

Next: ppointer

end;

Var p:ppointer;

Kompyuterda barcha turdagi ko`rsatkichlar bir xil ko`rinishga ega, ya'ni bular kompyuter xotirasining adresidir. Shuning uchun kompyuterda fizik nuqtai nazardan ko`rsatkichlar tursiz deb xisoblanadi. Turli ko`rsatkichlarni qo`llash ko`rsatkichlarni noto`gri ishlatish natijasida yuzaga keladigan xatolarni aniqlashda yordam beradi.

2. Dinamik o`zgaruvchilarni ishlatish

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

Ro`yxatlar, o`z navbatida, dinamik strukturaning quyidagi turlariga bo`linadi:

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

Ro`yxat - bu tartiblangan shunday strukturaki, uning xar 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 xar hil turli o'zgaruvchilar bo'lishi mumkin. Qo'shimcha qismda ro'yxatning keyingi yozuviga murojaat 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 qiymat beriladi. Ro'yxatni tugatish uchun xam ro'yxat oxiridagi elementning qo'shimcha qismiga NIL qiymat berish kerak.

XIV. TURBO - PASKAL MODULLARI. GRAPH MODULI

1. Foydalanuvchi modullarini tashkil qilish va ishlatish
2. Turbo - Paskal standart modullari
3. Graph modulining protseduralari

1. Foydalanuvchi modullarini tashkil qilish va ishlatish

Turbo-Paskalning shunday imkoniyatlari mavjudki, foydalanuvchilar, ya'ni biz o'zimizga yoqqan protsedura va funktsiyalar to'plamini ixtiyoriy dasturlarda ishlatishimiz mumkin.

Bu to'plamni esa maxsus fayl ko'rinishida xotirada saqlab qo'yib, unga ixtiyoriy dasturda murojaat qilish mumkin.

Turbo- Paskalda bu to'plam modul deb yuritiladi. Modulga yuqorida ta'kidlab o'tilgan protseduralar, funktsiyalar, kattaliklar ma'lum qonun-qoida asosida kiritiladi. Modul kompilyatsiyadan o'tkazilgandan keyin, undan dasturlarda foydalanish mumkin.

Modullar standart va foydalanuvchi modullarga bo'linadi.

Bu modullarga murojat dasturning e'lonlar bo'limidagi USES maxsus so'zidan keyin amalga oshiriladi:

USES crt, graph, ...;

Bu erda USES – paskal algoritmik tilining maxsus so'zi, crt, graph, ... - dasturda murojat etiladigan modullar ketma-ketligi.

2. Turbo - Paskal standart modullari

Quyida paskal algoritmik tili modullarining ayrimlari bilan tanishib o`tamiz.

SYSTEM moduli

Bu modul asosan matnlar (string turi) ustida turli xil amallarni bajaruvchi funktsiyalardan tashkil topgan bo`lib, ular quyidagilardan iborat:

Concat – matnlar ketma-ketligini qo`shish amalini bajaradi. Dasturda u quyidagicha qo`llaniladi:

Concat (s1, s2, ..., sN: string): string;

Bu erda s1, s2, ..., sN – string turidagi o`zgaruvchilar nomi.

Copy – matnlarning kerakli qismining qo`shimcha nusxasini oladi.

Qo`llanilishi:

Copy (s: string; n, m: integer): string;

Bu erda s – boshlang`ich matn, n – matnda ajratib olinayotgan qismdagi birinchi belgining tartib nomeri, m – ajratib olinayotgan qismdagi belgilar soni.

Delete – matnlarning ayrim qismlarini o`chirish uchun ishlatiladi. Qo`llanilishi:

Delete (var s: string; n: integer; m: integer);

Bu erda s – boshlang`ich matn, n – matnda ajratib olinayotgan qismdagi birinchi belgining tartib nomeri, m – ajratib olinayotgan qismdagi belgilar soni.

Length – matnning uzunligini aniqlash uchun ishlatiladi.

Qo`llanilishi:

Length (s: string): integer;

Bu erda s- o`zgaruvchi nomi.

Quyida yuqorida ko`rsatilgan funktsiyalarga doir misolni ko`rib o`tamiz:

Program misol;

Uses system

Var a, b, c, d, e: string;

i: integer;

Begin

a := 'salom,';

b := 'navro`z,';

c := 'bahor bayrami';

d := concat (a, b, c,);

e := copy (a, 1, 5);

i := length (c);

Writeln (d, ' ', e);

Writeln (I);

End.

CRT moduli

Bu modul asosan ekran, klaviatura va dinamikaning ishlashini muvofiqlashtirish uchun ishlatiladi. Bu modulga murojat qilish bilan Turbo Paskal kutubxonasining quyidagi funktsiyalaridan foydalanish mumkin:

Clrscr – ekranni tozalaydi va kursorni ekranning yuqori chap burchagiga joylashtiradi.

Textcolor (color) – matnning rangini aniqlaydi;

Textbackground (color) – ekranda xosil bo`luvchi rangni aniqlaydi;

Gotoxy (i, j) – ekrandagi i qator va j ustundagi nuqtaga kursorni joylashtiradi;

Inline – kursor turgan qatorni belgilardan bo`shatadi;

Clreol – kursor turgan qatorni o`chiradi;

Delay (n) – dastur bajarilishini n millisekundga to`xtatib turadi;

Sound (m) – dinamikani oldindan berilgan m chastotada ishlashini ta'minlaydi;

Window (x1, y1, x2, y2) – kompyuter ekranida matn uchun oyna xosil qiladi.

Bu erda x1, y1 – oynaning yuqori chap burchagi, x2, y2 – oynaning pastki o`ng burchagi koordinatalari.

Quyida yuqorida ko`rsatilgan funktsiyalarga doir misolni ko`rib o`tamiz:

Program misol;

Uses crt;

Begin

Textbackground (black); ekranni qora rangga bo`yash

Clrscr; ekranni tozalash

Textbackground (brown); xosil qilinayotgan oynaning rangini tanlash

Window (36,11,44,14); oyna o`lchamini aniqlash
Clrscr; oyna ichini tozalash
Gotoxy (1,2); kursorni joyini tanlash
Textcolor (blink + blue); yozilishi kerak bo`lgan tekstning rangini tanlash
Write (* *); ko`zni xosil qilamiz
Textcolor (white); tekstning rangini o`zgartirish
Gotoxy (5,3); kursorning joyini tanlash
Write (^); burunni xosil qilamiz
Gotoxy (1,4); kursorni joyini tanlash
Write (#####); og`izni xosil qilamiz
End.

GRAPH moduli

Bu modul o`z ichiga kompyuterda turli xil chizmalar va tasvirlar xosil qiluvchi funktsiyalarni olgan.

Ekranda tasvirlarni xosil qilish uchun grafik rejimiga o`tish zarur bo`ladi. Bu rejimda matn rejimidan farqli ekran rang-barang nuqtalar – pixcellar to`plamidan tashkil topgan to`rtburchakdan iborat bo`ladi.

Ekranning xolatini adapterlar belgilaydi va ular bir-biridan ekranga mumkin qadar ko`p nuqtalar joylashtira olish va rang-baranglikni oshirish imkoniyati bilan farqlanadi.

Kompyuter odatda matn xolatida ishlaydi. Ekranni grafik xolatiga o`tkazish uchun quyidagi funktsiyadan foydalaniladi:

Initgraph (gd, gm, path);

Bu erda gd – drayver nomeri, gm – xolat nomeri, path – drayverni saqlab turuvchi faylga o`tish yo`li.

Agar path o`rnida “ belgi turgan bo`lsa, drayver joriy katalogdan axtariladi. Agar 0 ga teng bo`lsa, bu drayver uchun grafik rejimning eng maqbul varianti kompyuter tomonidan avtomatik tarzda tanlanadi.

Grafik rejimdan chiqish uchun Closegraph funktsiyasi ishlatiladi.

Ekranda turli xil nuqta va chiziqlarni xosil qilish uchun quyidagi funktsiyalar ishlatiladi:

Putpixel (x,y, Color) – ekranda x,y koordinatali nuqtani Color rangida xosil qiladi;

Getpixel (x,y) – ekrandagi x,y koordinatali nuqtaning rangini aniqlaydi;

SetColor (Color) – chiziqlarning rangini belgilaydi. Color ga 0 dan 15 gacha qiymat berilib, 16 xil rangni xosil qilish mumkin. Bu erda 0 – black (qora), 15 – white (oq), ya'ni raqamlarning o`rniga ranglarning inglizcha nomlari yozilishi mumkin;

Line (x1,y1,x2,y2) - x1,y1 nuqtadan x2,y2 nuqttagacha kesma xosil qiladi;

Circle (x,y,r) - x,y markazga ega r radiusli aylana xosil qilish;

Rectangle (x1,y1,x2,y2) - x1,y1 yuqori chap va x2,y2 pastki o`ng uchlarga ega bo`lgan to`gri to`rtburchakni xosil qilish;

Drawpoly (numpointes, polypointes) - ko`pburchak xosil qilish. Bu erda numpointes – ko`pburchak uchlari soni, polypointes - ko`pburchak uchlari koordinatalari berilgan massiv.

Quyida yuqorida ko`rsatilgan funktsiyalarga doir misolni ko`rib o`tamiz:

1-misol. Uchlarining koordinatalari (10, 20) va (10, 100) bo`lib OX o`qiga perpendikulyar kesmani yasang.

Program misol1;

Uses crt, graph;

Var gd, gm: integer;

Begin

Gd := detect; initgraph (gd,gm, ”);

SetColor (red);

Line (10,20,10,100);

Readln;

Closegraph;

End.

Natijada ekranda talab qilingan - OX abtsissalar o`qiga perpendikulyar kesma xosil bo`ladi.

2-misol. Dioganalining koordinatalari (50,150) va (200,50) bo`lgan to`g`ri to`rtburchak yasang.

```
Program misol2;  
Uses crt, graph;  
Var gd, gm: integer;  
Begin  
  Gd := detect; initgraph (gd,gm,"");  
  Rectangle (50,150,200,50);  
  Readln;  
  Closegraph;  
End.
```

Natijada ekranda talab qilingan to`g`ri to`rtburchak xosil bo`ladi.

3-misol. Markazi (80,80), radiusi 60 bo`lgan aylanani chizing.

```
Program misol3;  
Uses graph;  
Var gd, gm: integer;  
Begin  
  Gd := detect; initgraph (gd,gm,"");  
  circle (80,80,60);  
  Readln;  
  Closegraph;  
End.
```

Natijada ekranda talab qilingan aylana xosil bo`ladi.

3. Graph modulining protseduralari

Graph modulining yana shunday protseduralari mavjudki, ular yordamida chizmalarni ekran bo`ylab harakatga keltirish mumkin. Statik chizmalarni harakatga

keltirishning bir necha usullari bor. Ulardan biri harakatlantirishni takrorlanish buyrug'i orqali tashkil qilishdir.

Ikkinchi usul ekranda chizilgan tasvir joylashgan sohani massiv ko'rinishida eslab qolib, uni maxsus protseduralar yordamida ekranning kerakli nuqtasiga ko'chirishdir.

Quyida bu protseduralar keltirilgan:

ImageSize (X1,Y1,X2,Y2) - ekranning chap yuqori nuqtasi (X1,Y1), o'ng pastki nuqtasi (X2,Y2) koordinatadagi to'g'ri to'rtburchakli sohasini saqlash uchun kerak bo'lgan xotiraning o'lchamini baytlarda olish.

GetImage (X1 ,Y1 ,X2,Y2,Area) - xotiraning berilgan Area maydonida to'g'ri to'rtburchakli tasvirni saqlaydi. Bu protsedurada Area-tasvir saqlanadigan joyning adresini saqlaydigan o'zgaruvchi.

PutImage (X,Y,Area,mode) - ekranning berilgan joyiga tasvirni chiqaradi.

Bu protsedurada (X,Y) - xotiraning Area maydonidagi tasvirdan nusxa ko'chiriladigan ekran maydonining chap yuqori burchagi.

mode - tasvirni ekranga chiqarish rejimi;

Quyidagi dastur GetImage, PutImage protseduralari va ImageSize funktsiyasini ishlatgan holda tasvir chizilgan to'g'ri to'rtburchakli sohani eslab qoladi va ekranning berilgan joyiga berilgan rejimda chiqaradi.

```
uses Crt,Graph;
```

```
var Gd,Gm,Size:integer;
```

```
P:pointer;
```

```
begin
```

```
Gd:=Detect;
```

```
InitGraph(Gd,Gm,' ');
```

```
{ekranning (0,0,40,40) sohasida tasvir chiziladi}
```

```
SetFillStyle(10, LightGreen);
```

```
Bar(0,0,40,40);
```

```
Rectangle(0,0,40,40);
```

{Size o'zgaruvchisi ekranning(0,0,40,40) sohasini saqlash uchun kerak bo'lgan xotiraning o'lchamini baytlarda qabul qiladi}

Size:=ImageSize(0,0,40,40);

{P ko'rsatkichli o'zgaruvchi xotiraning Size da ajratilgan sohasiga murojaat qiladi}

GetMem(P,Size);

{ ekranning (0,0,40,40) sohasi P o'zgaruvchi ko'rsatayotgan xotiraning sohasida saqlanadi}

GetImage(0,0,40,40,P^);

{saqlangan tasvir ixtiyoriy klavish bosilmaguncha tasodifiy koordinatadagi nuqtalarga chiqadi}

repeat

PutImage(Randome(GetMaxX),Random(GetMaxY),P^,NormalPut);

until keypressed;

Readln;

CloseGraph;

end.

Graph modulining quyidagi protseduralari ekran va oynalarni boshqarish imkonini beradi:

ClearDevice - grafik ekranni tozalash;

ClearViewPort - grafik oynani tozalash;

SetActivePage - videoxotiraning aktiv qismini o'rnatish;

SetViewPort - oynaning o'lchamini o'rnatish;

SetVisvalPage - tasvir etilayotgan videoxotiraning aktiv qismini o'rnatish.

Xulosa qilib shuni aytish mumkinki, Turbo-Paskalning keyingi, ya'ni yangi versiyalarida yuqorida keltirilgan modullarning imkoniyatlari qo'shimcha protsedura va funktsiyalar yordamida kengaytirilgan bo'lishi mumkin.

Hatolar kodlari va ular to`g`risida ma`lumotlar

№	Xatoning nomi	Ruscha	Lotincha
1	Out of memory	Выход за границы памяти	Xotira chegarasidan chiqish
2	Identifier expected	Не указан идентификатор	Identifikator ko`rsatilmagan
3	Unknown identifier	Неизвестный идентификатор	Noma`lum identifikator
4	Duplicate identifier	Двойной идентификатор	Identifikator qaytarilmoqda
5	Syntax error	Синтаксическая ошибка	Sintaksis xato
6	Error in real constant	Ошибка в вещественной константе.	haqiqiy o`zgarmasda xato
7	Error in integer constant	Ошибка в целой константе	Butun o`zgarmasda xato
8	String constant exceeds line	Строковая константа превышает допустимые размеры	O`zgarmas qatori berilgan o`lchovdan oshib ketgan
9	Too many nested files	Слишком много вложенных файлов	Ichma-ich qo`yilgan fayllar ko`p
10	Unexpected end of file	Не найден конец файла	Fayl oxiri topilmadi
11	Line too long	Слишком длинная строка	Juda uzun qator
12	Type identifier expected	Здесь нужен идентификатор типа	Identifikator turi kerak
13	Too many open files	Слишком много открытых файлов	Ochilgan fayllar ko`p
14	Invalid file name	Неверное имя файла	Fayl nomi to`g`ri emas
15	File not found	Файл не найден	Fayl topilmadi
16	Disk full	Диск заполнен	Disk to`lgan
17	Invalid compiler directive	Неправильная директива компилятора	Kompilyator direktivasi to`g`

			ri emas
18	Too many files	Слишком много файлов	Fayllar juda ko`p
19	Undefined type in pointer definition	Неопределенный тип в объявлении указателя	Tur e`lon qilinmagan
20	Variable identifier expected	Отсутствует идентифи-катор переменной	O`zgaruvchini identifikatori ko`rsatilmagan
21	Error in type	Ошибка в объявлении типа	Turni e`lon qilishda xato
22	Structure too large	Слишком большая структура	Struktura juda katta
23	Set base type of range	Базовый тип множества нарушает границы	To`plamni bazaviy turi chega rani buzayapti
24	File components may not be files	Компонентами файла не могут быть файлы	Faylni komponentasi fayl bo`lmaydi
25	Invalid string length	Неверная длина строки	qator uzunligi noto`g`ri
26	Type mismatch	Несоответствие типов	Turlarning mos emasligi
27	Invalid subrange base type	Неправильный базовый тип для типа-диапазона	Diapazon uchun noto`g`ri turi ko`rsatilyapti
28	Lower bound greater than upper bound	Нижняя граница больше верхней	quyi chegara yuqorisidan katta
29	Ordinal type expected	Нужен порядковый тип	Tartibli tur lozim
30	Integer constant expected	Нужна целая константа	Butun o`zgarmas kerak

31	Constant expected	Нужна константа	O`zgarmas kerak
32	Integer or real constant expected	Нужна целая или вещественная константа	Butun yoki haqiqiy o`zgar mas kerak
33	Type identifier expected	Нужен идентификатор типа	Identifikator turi kerak
34	Invalid function result type	Неправильный тип результата функции	Funktsiya qiymati turi noto`g`ri
35	Label identifier expected	Нужен идентификатор метки	Belgi identifikatori kerak
36	BEGIN expected	Нужен BEGIN	BEGIN kerak
37	END expected	Нужен END	END kerak
38	Integer expression expected	Нужно выражение типа	INTEGER turidagi ifoda kerak
39	Ordinal expression expected	Нужно выражение перечисляемого типа	Keltirilayotgan turlarni ifodasi kerak
40	Boolean expression expected	Нужно выражение типа BOOLEAN	BOOLEAN turidagi ifoda kerak
41	Operand types do not match operator	Типы операндов не соответствуют операции	Bajarilayotgan amallar mos tushmayapti
42	Error in expression	Ошибка в выражении	Ifodada xato
43	Illegal assignment	Неверное присваивание	Noto`g`ri o`zlashtirish
44	Field identifier expected	Нужен идентификатор поля	Maydon identifikatori kerak
45	Object file too large	Объектный файл слишком большой	Ob'ektli fayl juda katta
46	Undefined external	Неопределенная внешняя процедура	Aniqlanmagan tashqi protse dura

47	Invalid object file record	Неправильная запись объектного файла	Ob'ektni fayl noto`g`ri yozilgan
48	Code segment too large	Сегмент кода слишком большой	Kod segmenti juda katta
49	Data segment too large	Сегмент данных слишком велик	Ma'lumotlar segmenti katta
50	DO expected	Нужен оператор DO	DO operatori kerak
51	Invalid PUBLIC definition	Неверное PUBLIC-определение	PUBLIC-noto`g`ri
52	Invalid EXTRN definition	Неправильное EXTRN-определение	EXTRN-noto`g`ri
53	Too many EXTRN definition	Слишком много EXTRN-определений	EXTRN- juda ko`p
54	OF expected	Требуется OF	OF kerak
55	INTERFACE expected	Требуется интерфейсная секция	Interfeys kerak
56	Invalid relocatable reference	Неправильная перемещаемая ссылка	Noto`g`ri ko`rsatish
57	THEN expected	Требуется THEN	THEN kerak
58	TO or DOWNT0 expected	Требуется TO или DOWNT0	TO yoki DOWNT0 kerak
59	Undefined forward	Неопределенное опережающее описание	Aniqlanmagan shoshilinch yozuv
60	Too many procedures	Слишком много процедур	Protseduralar ko`p
61	Invalid typecast	Неверное преобразование типа	Turlarni noto`g`ri qayta o`zgartirish

62	Division by zero	Деление на ноль	Nolga bo'lish
63	Invalid file type.	Неверный файловый тип	Fayl turi noto'g'ri
64	Cannot Read or Write variables of this type	Нет возможности считать или записать переменные данного типа	O'zgaruvchini berilgan turi bo'yicha yozish imkoniyati yo'q
65	Pointer variable expected	Нужно использовать переменную-указатель	O'zgaruvchi–ko'rsatkichni ishlatish lozim
66	String variable expected	Нужна строковая переменная	qatorli o'zgaruvchi kerak
67	String expression expected	Нужно выражение строкового типа	qatorli turni ifodasi kerak
68	Circular unit reference	Перекрестная ссылка модулей	Modullar kesishishi
69	Unit name mismatch	Несоответствие имен программных модулей	Dasturli modullarni nomlari mos emas
70	Unit version mismatch	Несоответствие версий модулей	Modullarni versiyalari mos emas
71	Duplicate unit name	Повторное имя программного модуля	Dasturli modul nomi qaytarilyapti
72	Unit file format error	Ошибка формата файла модуля	Modul faylini formati xato
73	IMPLEMENTATION expected	Отсутствует исполняемая часть модуля	Modulni bajaruvchi bo'limi qatnashmayapti

74	Constant and case types do not match	Типы констант и тип выражения оператора CASE не соответствуют друг другу	CASE operatorini o'zgarmas va ifoda turlari mos tushmayapti
75	Record variable expected	Нужна переменная типа запись	Yozuv turini o'zgaruvchisi kerak
76	Constant out of range	Константа нарушает границы	O'zgarmas chegarani buza yapti
77	File variable expected	Нужна файловая переменная	Faylli o'zgaruvchi kerak
78	Pointer expression expected	Нужно выражение типа указатель	Ko'rsatkich turini ifodasi kerak
79	Integer or real expression expected	Нужно выражение вещественного или целого типа	haqiqiy yoki butun turdagi ifoda kerak
80	Label not within current block	Метка не находится внутри текущего блока	Belgi joriy blok ichida joylash magan
81	Label already defined	Метка уже определена	Belgi aniqlab bo'lingan
82	Undefined label in processing statement part	Неопределенная метка в предшествующем разделе операторов	Operatorlar bo'limida aniq lanmagan belgi
83	Invalid @ argument	Неправильный аргумент операции @	@ amalining noto'g'ri argumenti
84	Unit expected	Нужно кодовое слово UNIT	Kodli so'z UNIT kerak
85	”;” expected	Нужно указать”;”	”;” ni ko'rsatish kerak

86	”:” expected	Нужно указать”:”	”:” ni ko`rsatish kerak
87	”,” expected	Нужно указать”,”	”,” ni ko`rsatish kerak
88	”(” expected	Нужно указать ”(”	”(” ni ko`rsatish kerak
89	”)” expected	Нужно указать”)”	”)” ni ko`rsatish kerak
90	”=” expected	Нужно указать”=”	”=” ni ko`rsatish kerak
91	”:=” expected	Нужно указать”:=”	”:=” ni ko`rsatish kerak
92	”[” or ”(.” expected	Нужно указать ”[” или ”(.”	”[” yoki ”(.” ni ko`rsatish kerak
93	”]” or ”.)” expected	Нужно указать”]” или ”.)”	”]” yoki ”.)” ni ko`rsatish kerak
94	”.” expected	Нужно указать”.”	”.” ni ko`rsatish kerak
95	”..” expected	Нужно указать”..”	”..” ni ko`rsatish kerak
96	Too many variables	Слишком много переменных	O`zgaruvchilar juda ko`p
97	Invalid FOR control variable	Неправильный параметр цикла оператора FOR	FOR operatori noto`g`ri
98	Integer variable expected	Нужна переменная целого типа	Butun turdagi o`zgaruvchi kerak
99	File and procedure types are not allowed here	Здесь не могут использоваться файлы или процедурные типы	Fayllar yoki protsedura turlari ishlatilishi mumkin emas
100	String length	Несоответствие длины	qator uzunligi mos emas

	mismatch	строки	
101	Invalid ordering of fields	Неверный порядок полей	Maydon tartibi noto`g`ri
102	String constant expected	Нужна константа строкового типа	qator turidagi o`zgarmas kerak
103	Integer or real variable expected	Нужна переменная типа INTEGER или REAL	INTEGER yoki REAL turidagi o`zgaruvchilar kerak
104	Ordinal variable expected	Нужна переменная порядкового типа	Tartibli turdagi o`zgaruvchilar kerak
105	INLINE error	Ошибка в операторе INLINE	INLINE operatorida xato
106	Character expression expected	Предшествующее выражение должно иметь символьный тип	Oldingi ifoda simvolli turda bo`lishi kerak
107	Too many relocation items	Слишком много перемещаемых элементов	O`tkaziladigan elementlar juda ko`p
108	Overflow in arithmetic operator	Переполнение при выполнении арифметического оператора	Arifmetik operatorni bajarish da xotira to`lib qolishi
109	No enclosing FOR, WHILE or REPEAT statement	Нет операторов, заканчивающих операторы FOR, WHILE или REPEAT	FOR, WHILE yoki REPEAT operatorlarni tugatuvchi operatorlar yo`q
110	Debug information table overflow	Переполнение информационной таблицы	Tuzatish ma'lumotlar jadvali

		отладки	ning to`lib qolishi
111	N/A		
112	CASE constant out of range	Константа CASE нарушает допустимые границы	CASE o`zgarmasi mumkin bo`lgan chegarani buzayapti
113	Error in statement	Ошибка в операторе	Operatorda xato
114	Cannot call an interrupt procedure	Невозможно вызвать процедуру прерывания	Uzilish protsedurasini chaqi rish imkoniyati yo`q
115	N/A		
116	Must be in 8087 mode to compile this	Для компиляции необходим режим 8087	Kompilyatsiya uchun 8087 rejimi kerak
117	Target address not found	Указанный адрес не найден	Ko`rsatilgan adres topilmadi
118	Include files are not allowed here	Здесь не допускаются включаемые файлы	qo`shiluvchi fayllar mumkin emas
119	No inherited methods are accessible here	В этом месте программы нет унаследованных методов	Dasturni bu joyida ajdodiy usullar yo`q
120	N/A		
121	Invalid qualifier	Неверный квалификатор	Noto`g`ri kvalifikator
122	Invalid variable reference	Недействительная ссылка на переменную	O`zgaruvchiga aniq bo`lma-gan ko`rsatish

123	Too many symbols	Слишком много символов	Juda ko`p simvollar
124	Statement part too large	Слишком большой раздел операторов	Juda katta operatorlar bo`limi
125	N/A		
126	Files must be var parameters	Файлы должны передаваться как параметры-переменные	Fayllar parametr-o`zgaruvchilar kabi uzatilishi lozim
127	Too many conditional symbols	Слишком много условных символов	Shartli simvollar juda ko`p
128	Misplaced conditional directive	Пропущена условная директива	Shartli direktiva tushirib qoldirilgan
129	ENDIF directive missing	Пропущена директива ENDIF	ENDIF direktivasi tushirib qoldirilgan
130	Error in initial conditional defines	Ошибка в условных определениях	Shartli aniqlashda xato
131	Header does not match previous definition	Заголовок не соответствует предыдущему определению	Mavzu nomi oldingi aniqlashga mos emas
132	Critical disk error	Критическая ошибка диска	Diskda xatolik
133	Cannot evaluate this expression	Нельзя вычислить данное выражение	Ifodani hisoblash mumkin emas
134	Expression incorrectly germinated	Некорректное завершение выражения	Ifodani noaniq yakunlanishi
135	Invalid format specifier	Неверный спецификатор формата	Formatni noto`g`ri spetsifika

			tori
136	Invalid indirect reference	Недопустимая косвенная ссылка	Mumkin bo`lmagan mavhum ko`rsatish
137	Structured variable are not allowed here	Здесь нельзя использовать переменную структурного типа	Bu erda o`zgaruvchining strukturali turini ishlatish mumkin emas
138	Cannot evaluate without System unit	Нельзя вычислить выражение без модуля SYSTEM	Ifodani SYSTEM modulisiz hisoblash mumkin emas
139	Cannot access this symbol	Нет доступа к данному символу	Berilgan simvolga ruxsat yo`q
140	Invalid floating-point operation	Недопустимая операция с плавающей запятой	Suzuvchi vergul bilan mumkin bo`lmagan amal
141	Cannot compile overlay to memory	Нельзя выполнить компиляцию оверлейных модулей в память	Xotirada overlay modulini kompilyatsiya qilish mumkin emas
142	Procedure or function variable expected	Должна использоваться переменная процедурного типа	Protsedura turidagi o`zgaruvchini ishlatish kerak
143	Invalid procedure or function reference.	Недопустимая ссылка на процедуру или функцию	Protsedura yoki funktsiyaga mumkin bo`lmagan ko`rsatish
144	Cannot overlay this	Этот модуль не может	Bu modulni overlay sifatida

	unit	использоваться в качестве оверлейного	ishlatish mumkin emas
145	Too many nested scopes	Слишком много вложений	Ichma-ichlik ko`p
146	File access denied	Отказано в доступе к файлу	Faylga kirish bekor qilinadi
147	Object type expected	Здесь должен быть тип OBJECT	Bu erda OBJECT turi bo`li shi kerak
148	Object types are not allowed	Нельзя объявлять локальные объекты	Lokal ob'ektlarni e`lon qilish mumkin emas
149	VIRTUAL expected	Пропущено слово VIRTUAL	VIRTUAL so`zi tushirib qoldirilgan
150	Method identifier expected	Пропущен идентификатор инкапсулированного правила	Inkapsulirovan qoidali identifikator tushirib qoldi rilgan
151	Virtual constructor are not allowed	Конструктор не может быть виртуальным	Konstruktor virtual bo`lishi mumkin emas
152			
153	Destructor identifier expected	Пропущен идентификатор деструктора	Destruktor identifikatori tushirib qoldirilgan
154	Fail only allowed within constructor	Обращение к стандартной процедуре FAIL может содержаться только в конструкторе	FAIL standart protsedurasi ga murojaat faqat konstruktorda joylashgan

155	Invalid combination of opcode and operands	Недопустимая комбинация кода команды и операндов	Buyruqlar va amallar kodlarining mumkin bo'lmagan kombinatsiyasi
156	Memory reference expected	Отсутствует адрес	Adres ko'rsatilmagan
157	Cannot add or subtract relocatable symbols	Нельзя складывать или вычитать перемещаемые символы	Aralash simvollarni qo'shish va ayirish mumkin emas
158	Invalid register combination	Недопустимая комбинация регистров	Registrlarni mumkin bo'lmagan kombinatsiyalari
159	286/287 instructions are not enabled	Недоступен набор команд микропроцессоров 286/287	286/287 mikroprotsessorlarni buyruqlari mumkin emas
160	Invalid symbol reference	Недопустимая ссылка на символ	Simvolga mumkin bo'lmagan ko'rsatish
161	Code generation error	Ошибка генерации кода	Kodni generatsiyasida xato
162	ASM expected	Отсутствует зарезервированное слово ASM	Rezervdagi ASM so'zi qatnashmayapti

GLOSSARIY

Algoritm - masalani echish uchun bajarilishi lozim bo'lgan amallar ketma-ketligi

FORTTRAN-FORMylar TRANslator -formulalar translyatori

ALGOL -ALGORITMIC Language -algoritmik til

KOBOL- Common Business Oriented Language - “savdo-sotiq masalalariga mo'ljallangan til”

PL/1 - Programming Language/ 1 - 1-tartib raqamli dasturlash tili

BEYSIK -BASIC - Beginner's All Purpose Symbolic Instruction Code - boshlovchilar uchun ko'p maqsadli dasturlash tili

LIFO - «oxirida kirib birinchi chiqish»

Funktsiya - shunday qism dasturki, bunda bitta natija chiqadi

Protsedura- qism dasturida natija esa bitta yoki bir necha bo'lishi mumkin.

Fayl - xotiraning nomlangan soxasi

Yozuv- turli toifadagi ma'lumotlarning cheklangan to'plami.

Toifalashgan fayllar- bir xil toifali elementlardan tashkil topadi.

Matnli fayl- qatorlardan tashkil topgan fayldir.

Stek - bu shunday ro'yxatki, uning elementiga faqat bir nuqtadan murojaat qilish mumkin.

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

Tayanch so'zlar lug'ati

inglizcha	ruscha	lotincha
Abort (eybort)	Прервать выполнение программы	Dastur bajarilishini uzish
add (edd)	Добавить	Qo'shmoq
Align (elayn)	Выравнивать	Tekislamoq
all (oll)	весь, все	Barchasi
Apply (eplay)	Зафиксировать, применить	Qo'llamoq
Arrange (arreyndj)	Упорядочить	Tartiblamq
Arrow (errou)	Стрелка	Yo'nalish
Background (bekgraund)	Фон	fon

bar (bar)	полоса, строка, зона	Satr, zona, yo`lak
black (blek)	Черный	Qora
bold (bold)	Полужирный (о шрифтах)	Qalinroq
Border (border)	рамка, граница	chegara, ramka
Break (breyk)	Разрыв	uzilish (dastur)
Browse (brouse)	Просмотреть	ko`rmoq (faylni)
Bullet (bullet)	Символ списка	ro`yxat belgisi
Button (batton)	Кнопка	Tugma
Calculator (kalkyuleyter)	Калькулятор	xisoblovchi, kalkulyator
Cancel (kansel)	Отменить, аннулировать	yo`q qilmoq (oxirgi xarakatni)
cell (sel)	ячейка (таблица)	Yacheyka, jadval
Change (chaynj)	Изменить	O`zgartirmoq (direktoriyni)
Check (chek)	Проверка управления	boshqarishni tekshirish
Choose (chuz)	Выбрать	Tanlamoq
Clear (kleir)	Очистить	Tozalamoq
click (klik)	Щелкнуть	bosmoq (qisqa vaqt sichqonchada)
Clipboard (klipbord)	буфер обмена	Almashtirish buferi (bo`lak, oraliq)
Close (klouz)	Заккрыть	Yopmoq
Color (kalor)	Цвет	Rang
Column (kolumn)	Столбец, колонка	ustun, kolonka
Compare (kompeyr)	Сравнить	Solishtirmoq
Continue (kontinyu)	Продолжать	Davom ettirmoq
copy (kopi)	Копировать	Nusxa olmoq
Create (krieyt)	Создать	Yaratmoq

cut (kat)	Вырезать	Kesmoq
date (deyt)	Дата	Sana
Default (defolt)	по умолчанию	Oshkormas
Delete (delit)	Удалить	Chetlashtirmoq
Desktop (desktop)	Рабочий стол	Ishchi stoli
Device (divays)	Устройство	Qurilma
Directory (direktori)	Каталог, директория	Katalog, direktoriya
down (daun)	Вниз	Pastga
Draft (draft)	Черновой (о качестве печати)	qora yozma
draw (dro)	Рисовать	Chizmoq
Drive (drayv)	Дисковод, логический диск	Disk yuritgichi, mantiqiy disk
Disk	Диск	Tashqi xotira qurilmasi
edit (edit)	Редактировать, править	Taxrirlash, o`zgartirish
enter (enter)	ввести, войти	Kiritish, kirish
Erase (irayz)	Стереть, уничтожить	O`chirish, yo`q qilish
error (error)	Ошибка	Xato
Exist (ekzist)	Существовать	Mavjud bo`lmoq
exit (eksit)	выход, выходить	chiqish, chiqmoq
Extension (ekstenshn)	Расширение	Kengaytma
False	Ложь	Yolg`on
fail (feyl)	Потерпеть неудачу	Noqulaylikka uchrash
field (fild)	Поле	Maydon
file (fayl)	файл	Fayl
find (faynd)	Найти	Topmoq
Folder (folder)	Папка	Papka

font (font)	Шрифт	Shrift
Foreground (forgraunt)	Передний план	Oldidan ko`rinishi
get (get)	Получить	Olmoq
go to (gou tu)	Перейти к.. .	.. . ga o`tish
Information	Информация	Axborot
Insert (insert)	Вставка, вставить	Orasiga qo`yish, orasiga qo`ymoq
Italic (italik)	курсив (о шрифте)	Kursiv (shrift turi)
item (item)	Элемент	Element
Justify (djastifay)	Выравнивать (по ширине)	Tekislaroq (eni bo`ylab)
key (key)	Клавиша или ключ (в базах данных)	Klavisha yoki kalit (berilganlar bazasida)
Layout (leaut)	Разметка	Belgilash
left (left)	Левый	Chap
line (layn)	строка (документа, программы)	satr, yo`l (xujjatda, programmada)
link (link)	Связывать	Bog`lamoq (kompyuterlarni)
list (list)	Список	Ro`yxat
main (meyn)	Главный	bosh (menyu, programma)
make (meyk)	Сделать, создать	qilmoq, yaratmoq (fayllarni)
Margin (merdjn)	Граница	Chegara
Memory (memori)	Память	Xotira (kompyuter)
Message (messedj)	Сообщение	Bildirish
Modify (modifay)	Модифицировать, изменить	O`zgartirmoq
move (muv)	Переслать, переместить,	Jo`natmoq, siljitmoq, xarakatlantirmoq

	передвинуть	
name (neym)	Имя	Nom
new (nyu)	Новый	yangi (faylni yaratish)
Normal (normal)	Обычный, нормальный (о стиле)	Oddiy
Option (opshn)	режим, опция	Rejim, bo`lak
Outline (autlayn)	контур, план, схема, структура	Kontur, reja, tizma, tuzilish
Overwrite (overrayt)	Переписать	qayta yozmoq
Page (peydj)	Страница	Saxifa
Palette (pallet)	Палитра	Palitra (rangga oid)
Paper (peyper)	Бумага	Qog`oz
Paragraph (peragraf)	Абзац	Abzats (xat boshi)
paste (pest)	Вставить	Joyiga qo`ymoq
path (pet, pes)	Путь	yo`l (fayl)
pick (pik)	Преобразовать	O`zgartirmoq
Picture (pikcher)	Рисунок, изображение, картина	Surat, tasvir, rasm
Point (point)	Точка, указать	nuqta, ko`rsatmoq
pop up (pop ap)	Всплывающее (о меню)	xolatli menyu
Portrait (porteyt)	Вертикальный, книжный	Vertikal, kitobli
press (press)	Нажать	Bosmoq
print (print)	Печатать	chop qilmoq
Program group (programm grup)	Программная группа	Programma guruhi
Program item (program item)	Программный элемент	programma elementi

Prompt (prompt)	Приглашение (о DOS)	taklifnoma
proof (pruf)	Стандартный (о качестве печати)	standart (pechat sifati)
pull down (pul down)	Ниспадающее, нисходящее (о меню)	pastga tushuvchi, pastga yuruvchi (pastga tushgan menyu)
push button (pash batton)	Командная кнопка	buyruq tugmasi
Put (pat)	выдать, поместить	bermoq, joylashtirmoq
Quality (kualiti)	Качество	sifat
Quit (kuit)	Выход	chiqish

ADABIYOTLAR RO'YXATI

1. Holmatov T.X., Toyloqov N.I. Amaliy matematika, dasturlash va kompyuterning dasturiy ta'minoti. T. Mexnat, 2000 .
2. Kadirova N.R. Polatov A.M. Programmirovaniye na yasike Paskal T. 2004
3. Aripov M., Xaydarov A. Informatika asoslari T. "O`qituvchi" 2002 .
4. Alyaev Yu.A. i dr. Praktikum po algoritmizatsii i programmirovaniyu na yazik Paskal: Ucheb. posob. – M.: FIS, 2004. – 528s.
5. Bartenev O.V. Predpriyatie: programmirovaniye dlya vsekh bazovye ob'ekti i rascheti na odnoy diskete. – M.: Dialog – MIFI, 2003. – 464 s.
6. Faronov V.V. Turbo Paskal 7.0. Uchebnoe posobie. M.: Nolidj., 2002g.
7. Stavrovskiy A.B. Turbo Paskal 7.0. Kiev: BHV., 2000g.
8. Gurova L.I. Osnovi programmirovaniya. M., Fis, 1990.

9. Dokukina T.K. Programmirovaniye i algoritmicheskie yaziki. Uchebnik.M., Mashinostroenie, 1988.
- 10.Chernov B.I. Programmirovaniye na algoritmicheskix yazikax. M., Prosveshenie, 1991.
- 11.Donald Alkol. Yazik Paskal v illyustratsiyax. M., Mir, 1991.
- 12.Faronov V.V. Turbo Paskal (v 3x knigax). M.: MVTU-FESTO DIDAKTIK., 2000.
- 13.Fedorov A. Osobennosti programmirovaniya na Borland Pascal. Kiev:Dialektika, 1999.
- 14.Vasyukova N.D., Tyulyaeva V.V. Praktikum po osnovam programmirovaniya yazik Paskal. M.: Visshaya shkola., 1996.
- 15.Zavarikin V.M. Texnika vichisleniy i algoritmizatsiya. Uchebnoe posobie M., Prosveshenie, 1987