

O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI

FARG'ONA POLITEKNIKA INSTITUTI

ENERGETIKA FAKULTETI

“INFORMATIKA VA AXBOROT
TEKNOLOGIYALARI”

kafedrası

“Informatika va axborot texnologiyalari” fanidan

na'munaviy

KURS ISHI

MAVZU:

“O'quvchilarning karra jadvalini qay darajada o'zlashtirganini aniqlovchi dastur
tuzish”

Tuzuvchilar:

“I va AT” kafedrası dotsenti

f-m.f.n. M.X.Oxunov

“I va AT” kafedrası assistenti

S.I.Zokirov

Farg'ona - 2013

Topshiriq varaqasi

Talaba: _____

(Familiyasi, ismi, guruxi)

Mavzu: _____

(Mavzu nomi)

Vazifalar:

- 1.
- 2.
- 3.

Bajarish grafigi:

№	Ish nomi	Muddat Xaftada
1.	Kurs ishi mavzusini olish.	1-3
2.	Kurs ishini topshiriq varag'ini tuzish.	1-3
3.	Mavzu bo'yicha adabiyotlar bilan tanishish va tushuntirish xatining kirish qismini yozish.	3-5
4.	Nazariy qismni tayyorlash.	6-9
5.	Tushuntirish xatini tayyorlash.	9-13
6.	Mavzuda berilgan qurilma yoki dasturiy ta'minot ishini ko'rsatuvchi sxematik chizma tayyorlash	14-15
7.	Himoya qilish.	16-17

Mavzuni oldim: _____

(sana, talaba, imzosi)

Rahbar: _____

(imzo, sana, familiyasi, ismi, lavozimi)

MUNDARIJA

KIRISH.	4
NAZARIY QISM.	6
Turbo Paskal dasturlash tili muhiti.....	6
Paskal dasturlash tili va uning alifbosi.....	7
O‘zgarmas va o‘zgaruvchi miqdorlar	8
Standart funksiyalar va algebraik ifodalar	11
O‘zlashtirish va ma’lumotlarni ekranga chiqarish operatorlari	11
Ma’lumotlarni hotiraga muloqot usulida kiritish operatorlari	12
O‘tish va tarmoqlanish operatorlari	12
Parametrli takrorlash operatorlari	13
Shart bo‘yicha takrorlash operatorlari.....	14
AMALIY QISM.	16
Masalaning qo‘yilishi va uning yechish algoritmi.....	16
Dasturning to‘liq kodi.....	21
FOYDALANILGAN ADABIYOTLAR.	23

KIRISH.

O'zbekiston Respublikasida chuqur keng ko'lamli isloxotlar amalga oshirilgan, uzluksiz iqtisodiy ta'lim tizimini shakllantirishga birinchi darajali ahamiyat berilmoqda. Davlatimiz rahbari I.A.Karimov ta'kidlaganlaridek: "Biz kiyin sharoitda kelajagimiz poydevorini qurib oldik. Bunga o'rnimizga keladigan yosh avlod qanday baxo beradi, loyixalarimizni taroziga solib ko'rib nima deydi?- Xar birimiz ana shu savolni dilga tugib mehnat qilishimiz, yuqori natijalarga erishishimiz lozim. Mening ishonchim komilki, O'zbekiston xalqi farovon turmush kechiradigan buyuk davlat bo'ladi".

Kelajagi buyuk davlatni qurish tafakkuri, dunyoqarashi o'zgargan xodimlarimiz, mutaxassislarimizga ko'p jihatdan bog'liqdir. Yangicha fikrlaydigan, bozor munosabatlarida muvaffaqiyatli xo'jalik yuritadigan yuksak malakali, chuqur bilimli mutaxasislarni tayyorlash davr talabi bo'lib qoldi. Mamlakatimiz rivojlangan davlatlar qatoridan mustaxkam o'rin egallashi uchun zamonaviy kompyuter texnologiyalarini xayotimizning barcha jabxalariga, ayniqsa iqtisodiyotga keng joriy etish zarur.

O'zbekiston Respublikasida informatikani rivojlantirish, tamoyillari quyidagilardan iborat:

Birinchidan, zamonaviy axborot texnologiyasini rivojlantirish, davlat muassasalari va xo'jalik sub'ektlari, muassasalari va xo'jalik sub'ektlari, muassasa va tashkilotlar, xususiy shaxslar uchun axborot xizmatini yo'lga qo'yish:

Ikkinchidan, iqtisodiyot, fan, ta'lim, ijtimoiy soxada axborot tizimlarini shakllantirishga ko'maklashish;

Uchinchidan, Respublikaning jaxon axborot tizimlari va xalqaro tarmoqlariga ulanishini ta'minlash;

Ana shu uch asosiy tamoyillardan kelib chiqan xolda quyidagi asosiy vazifalar xalq etilishi talab kilinmokda:

A) axborotga tovar sifatida karashni ta'minlovchi va rag'batlantiruvchi xukukiy va iqtisodiy me'yorlarni joriy etish;

B) axborotni takdim etish, saqlash va uzatishga oid xalqaro standartlarga rioya kilish.

V) Informatika industriyasini shakllantirish va rivojlantirish, yangi axborot texnologiyalari, kompyuter va rivojlangan telekommunikatsiya texnikasi, dasturiy vositalar ishlab chiqarishni o'zlashtirish, ma'lumotlar bazasi va eksport ta'minoti xizmati bozorini shakllantirish;

G) ma'lumotlarni uzatish milliy axborot hisoblash tarmog'i, davlat xizmati va tuzilmalari idoraviy tarmog'i, vazirlik va idoralar, korxona va tashkilotlar, xususiy va tijorat lokal axborot tarmoqlarini, ilmiy-texnikaviy fanlar, avtomatlashgan ish o'rinlarini yaratish va rivojlantirish;

V) axborot tizimlari eng yangi modellarini yaratish soxasidan fundamental va amaliy tadqiqotlarni rag'batlantirish va qo'llab-quvvatlash,

E) informatika vositalaridan foydalanuvchilarni tayyorlashning uzluksiz tizimini takomillashtirish.

Informatikaning inson faoliyatining mustaqil soxasi sifatida ajralib chiqishi birinchi navbatda kompyuter texnikasining rivojlanishi bilan bog'liq. Bunda asosiy xizmat mikroprotsessor texnikasiga to'g'ri proporsional keladi, uning paydo bulishi 70-yillar o'rtalarida ikkinchi elektron inqilobni boshlab berdi. SHu davrdan boshlab xisoblash mashinalarining element negizini integral chizma va mikroprotsessorlar tashkil etdi. Informatika atamasi nafakat kompyuter texnikasi yutuklarini aks ettirish va foydalanish, balki axborotni uzatish va qayta ishlash jaraenlari bilan xam bog'lanadi.

Informatika axborotni qayta ishlash, ularni qo'llash va ijtimoiy amaliyotning turli soxalariga ta'sirini EXM tizimlariga asoslangan xolda ishlab chiqish, loyixalash, yaratish, ilmiy va muxandislik Fani soxasidir.

Informatika bu jixatdan axborot modellarini ko'rishning umumiy metadologik tamoyillari ishlab chiqishga yo'naltirilgan. SHu bois axborot uslublari ob'ekt, xodisa, jarayon va xokazolarni axborot modellari yordamida bayon etish imkoniyatiga egadir.

Kurs ishining maqsadi «Informatika» fanidan olingan nazariy va amaliy bilimlarini umumlashtirish, chukurlashtirish va amaliyda uchraydigan masalalarni ijodiy yendashgan xolda mustakil xal etish malakasini shakllantirishdan iboratdir.

O'zbekiston Respublikasi mustaqillikka erishganidan sung, rivojlanishning yukori boskichiga kutarildi. SHu borada kompyuter texnologiyalarining ishlab chiqarishga, iktosodiyotga va biznesga tadbiki kengayib bormokda. 1997 yilda O'zbekiston Respublikasi Oliy Majlisining tukkizinchi sessiyasida qabul qilingan «Ta'lim to'g'risida » va «Kadrlar tayerlash buyicha Milliy dastur to'g'risida» to'g'risida qonunlarida, O'zbekiston Respublikasi Oliy va O'rta iaxsus ta'lim vazirligining karorlari va buyruqlarida (jumladan, 11.4-sonli xa'at karori.

1996 yil 27 noyabr yuqori texnologiyalarni, xususan informatsion texnologiyalarni va axborotni qayta ishlash tizimlarini xaytga tadbik etishga katta e'tibor berilgan.

Raqobatbardosh iqtisodiet asosan yuqori malakali kadrlar tayyorlashda ta'lim jarayonining informatsiyalashtirilganligi va kompyuterlashtirilganligi muxim axamiyatga ega ekanligi ko'rsatilgan.

Informatikaning inson faoliyatining mustakil soxasi sifatida ajralib chiqishi birinchi navbatda kompyuter texnikasining rivojlanishi bilan boglik. Bunda asosiy xizmat mikroprotsessor texnikasiga to'g'ri keladi, uning paydo bulishi 70-yillar urtalarida ikkinchi elektoron inkilobini boshlab berdi.

SHu davrdan boshlab xisoblash iashinalarining element negizini integral chizma va mikroprotsessorlar tashkil etdi. Informatika atamasi nafakat kompyuter texnikasi yutuklarini aks ettirish va foydalanish, balki axborotni uzatish va qayta ishlash jaraenlari bilan xam boglanadi.

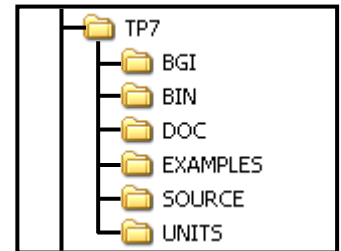
Informatika axborotni qayta ishlash, ularni kullash va ijtimoiy amaliyotning turli soxalariga ta'sirini EXM tizimlariga asoslangan xolda ishlab chiqish, loyixalash, yaratish, baxolash, ishlashning turli jixatlarini urganuvchi kompleks ilmiy va muxandislik fani soxasidir.

Informatika-kompyuterlar yordami va ularni qo'llash muxiti vositasida axborotni yangilash jaraenlari bilan bog'liq inson faoliyati soxasidir.

NAZARIY QISM.

Turbo Paskal dasturlash tili muhiti

Hozirgi kunda qo'llanilayotgan Turbo Paskalning "Turbo Pascal 7.0" naqli diskning odatda "TP7" nomli katalogiga joylashtiriladi (*umuman olganda boshqa katalogga joylashtirish ham mumkin*). Turbo Pascal 7.0 dasturlash tizimi yuzdan ortiq fayldan iborat bo'lib, ular vazifalariga qarab bir nechta kataloglarga joylashtirilgan (1-rasm). Turbo Paskalning asosiy fayllari "BIN" katalogida joylashgan. Qolgan kataloglarda asosan yordamchi fayllar hamda Turbo Paskalning imkoniyatlarini namoyish etuvchi dasturlar joylashgan. Ulardan asosiylari quyidagilar:



1 – rasm

BIN katalogida quyidagi asosiy fayllari joylashadi:

TURBO.EXE – dastur tuzish uchun integrallashgan muhit;

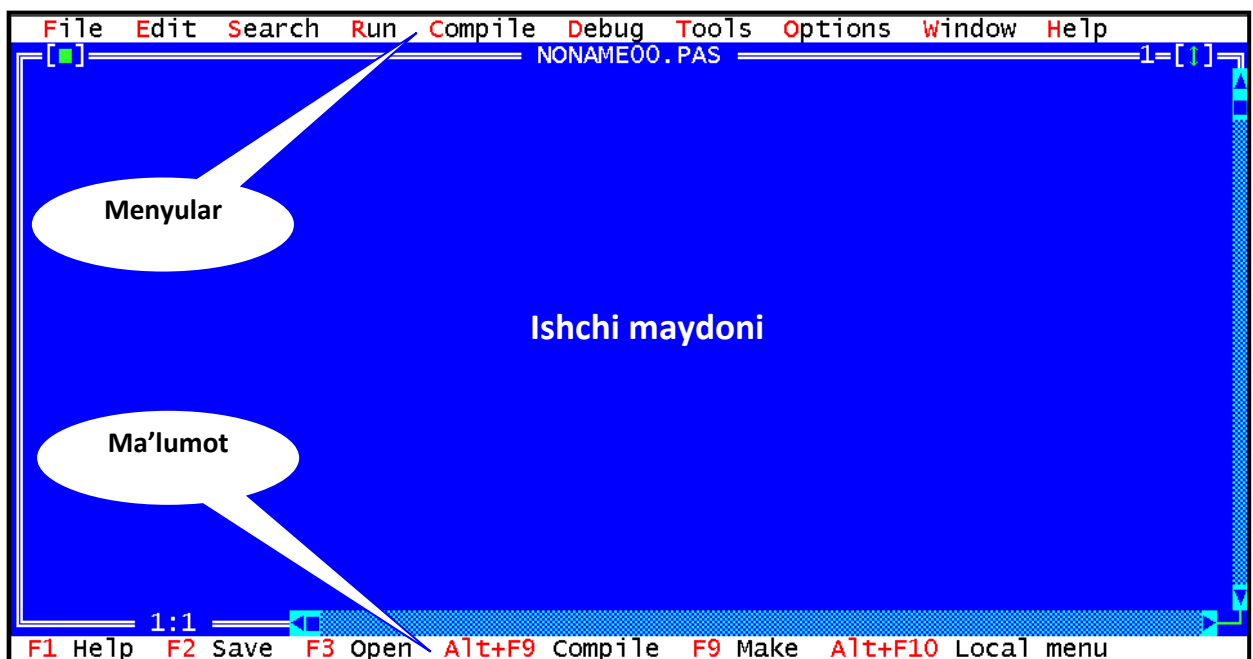
TURBO.HLP – dasturchiga yordam uchun ma'lumotlar fayli;

TURBO.TP – tizimning konfiguratsiyasi fayli.

BGI katalogida grafik holatda ishlash uchun zarur fayllar, **UNITS** katalogida esa Turbo Paskalning modullari joylashgan.

Turbo Paskalning qolgan katalog va fayllaridan ushbu darslik doirasida yoritilmaydi. *Ular haqida darslikda keltirilgan qo'shimcha adabiyotlar ro'yhatidagi qo'llanmalardan yetarlicha ma'lumot olishingiz mumkin.*

Turbo Paskal integrallashgan muhitiga kirish uchun **turbo.exe** fayli ishga tushiriladi. Natijada ekranda Turbo Paskal integrallashgan muhiti interfeysi



2 – rasm. Turbo Pascal 7.0 interfeysi ko'rinishi

ochiladi. U menyular satri, ishchi maydoni va ma'lumot satridan iborat (2-rasm).

Ko‘rib turganingizdek, Turbo Paskal integrallashgan muhiti interfeysi matn muharririni interfeysiga o‘xshaydi. Uning ishchi maydoniga dastur matni matn muharriridagi kabi kiritiladi. Ishchi maydoni “oyna” ko‘rinishida bo‘lib, u “matn kiritish oynasi” yoki qisqacha “oyna” deb ham yuritiladi. MS Worddagi kabi Turbo Paskalda ham bir nechta oyna ochib, ularning har biri bilan alohida ishlash imkoniyati mavjud. Bu bir vaqtda bir nechta dastur bilan ishlash imkonini beradi. Joriy vaqtda ishlanayotgan oyna **aktiv oyna** deyiladi.

Menyular satriga o‘tish uchun **F10** klavishi bosiladi. So‘ngra chapga yoki o‘ngga yo‘nalish klavishalari yordamida kerakli menyuni tanlab **ENTER** klavishi bosiladi. Kerakli menyuni sichqoncha yordamida ham tanlash mumkin.

Run menyusi dasturni ishga tushirish, **Compile** menyusi dasturni kompilyatsiya qilish (dasturni “mashina tili”ga o‘girib “EXE” kengaytmali fayl ko‘rinishida saqlash) amallarini o‘z ichiga oladi. Qolgan menyulardan dastur tuzishda deyarli foydalanilmaydi.

Menyular tarkibidagi amallarni ma’lum klavishalarni bosish orqali ham bajarish mumkin. Quyida asosiy amallarni bajarishga mo‘ljallangan klavishalar keltirilgan:

F3	ochish (tashqi hotiradagi faylni yuklash)
F2	saqlash
Alt+F3	aktiv oynani yopish
F6	bir oynadan ikkinchi oynaga o‘tish
Ctrl+F9	dasturni ishga tushirish
Alt+F5	dastur natijasini ekranda ko‘rish
Alt+F9	dasturni kompilyatsiya qilish
Alt + x	chiqish

Paskal dasturlash tili va uning alifbosi

Har qanday dasturlash tili kabi Paskal dasturlash tili ham o‘zining alifbosiga ega bo‘lib, u quyidagilardan iborat:

Lotin alifbosining 26 ta harfi : Aa, Bb, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk, Ll, Mm, Nn, Oo, Pp, Qq, Rr, Ss, Tt, Uu, Vv, Ww, Xx, Yy, Zz

(izohlar va matnlar yozish uchun kirill harflarini ham qo‘llash mumkin);

O‘nta arab raqami: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ;

Arifmetik amal belgilari: +(qo‘shish), - (ayirish), * (ko‘paytirish), /(bo‘lish);

Munosabat belgilari: =(teng), <>(teng emas), <(kichik), <=(katta emas), >(katta), >=(kichik emas);

Maxsus belgilar: . (nuqta), , (vergul), ; (nuqtali vergul), ‘ (apostrof), “ (qo‘sh tirnoq), ! (undov), ? (so‘roq), % (foiz), \$ (dollar belgisi), @ (tijorat belgisi), & (ampersand), (bo‘shliq, ekranda tasvirlanmaydi), (,), {, }, [,] (turli qavslar);

Mantiqiy amallar : **AND** ("VA" – *mantiqiy ko'paytirish amali*),
OR ("YOKI" – *mantiqiy qo'shish amali*),
NOT ("EMAS" – *mantiqiy inkor amali*).

Barcha dasturlash tillari singari Paskal dasturlash tili ham o'zining imlosi, qonun va qoidalariga ega bo'lib, ular asosida yuqorida keltirilgan harflar, belgilar va amallar yordamida ko'rsatma va buyruqlar tuziladi. Har bir ko'rsatma yoki buyruq “;” (*nuqtali vergul*) belgisi bilan yakunlanadi.

Dastur yozishda quyidagilar qo'llaniladi:

Konstantalar (o'zgarmaslar) – dastur ishlashi davomida qiymati o'zgarmaydigan miqdorlar;

O'zgaruvchilar – dastur ishlashi davomida qiymati o'zgaradigan miqdorlar;

Algebraik ifodalar – arifmetik amallar bilan bog'langan o'zgarmaslar, o'zgaruvchilar va funksiyalar;

Operatorlar – dasturlash tilining biror tugallangan amalini berish uchun mo'ljallangan buyrug'i;

Funksiya va protseduralar – o'z nomiga ega bo'lgan alohida dastur qismlari (bloklari). Ularga asosiy dasturdan murogaat etiladi;

Nishonlar – dasturda boshqarish uzatilayotgan operatorni ko'rsatadi.

Paskalning buyruq va ko'rsatmalari **modul** deb ataluvchi **TPU** kengaytmali maxsus fayllarda joylashgan. Ularga misol qilib **system** (*sistem*), **crt** (*sierti*), **graph** (*graf*) modullarini keltirish mumkin. Modullarning har biri ma'lum yo'nalishdagi buyruq va ko'rsatmalarni o'z ichiga olgan. Masalan **crt** moduli ekran va klaviatura bilan ishlashga (ekranni tozalash, ekranda bir nechta alohida oyna hosil qilish,...), **graph** moduli esa grafik holatda ishlashga mo'ljallangan buyruq va ko'rsatmalarni o'z ichiga oladi.

Label, Const, Var, Begin, End – paskal tilining maxsus so'zlari bo'lib, *label* – *nishon*, *const* (*constant* – *konstanta*) – *o'zgarmas miqdor*, *var* (*variable*) – *o'zgaruvchilarni tavsiflash*, *begin* – *boshlanish*, *end* – *tamomlash* ma'nolarini anglatadi.

O'zgarmas va o'zgaruvchi miqdorlar

Paskalda asosan uch xil: o'zgarmas, o'zgaruvchi va massiv (jadval) ko'rinishidagi miqdorlar ishlatiladi. Ular belgili, satrli, mantiqiy va sonli turdagi qiymatlarni qabul qilishi mumkin.

O'zgarmas miqdorlar.

Belgili o'zgarmaslar apostrof ichiga olingan bitta belgi – harf, raqam yoki maxsus belgidan iborat. Masalan:

'a'; 'B'; '9'; '-' va hokazo.

Satrli o'zgarmaslar uzunligi 255ta belgidan oshmagan va apostrof ichiga olingan harf, raqam va maxsus belgilar ketma-ketligidan ibopat. Masalan:

'Toshkent'; 'A 549'; '***.'; '37%'; 'A ='; '..-...-' va hokazo.

Mantiqiy o'zgarmaslar faqat True (rost) yoki False (yolg'on) qiymatlardan birini qabul qiladi.

Sonli o'zgarmlar ikki turda – **butun** yoki **haqiqiy** bo'lishi mumkin. Haqiqiy sonlar o'z navbatida qo'zg'almas nuqtali va qo'zg'aluvchi nuqtali sonlarga bo'linadi.

Qo'zg'almas nuqtali sonlar – o'nli kasr ko'rinishidagi sonlardir.

Masalan: - 2.753; 283.45; 0.517; - 0.0013.

Qo'zg'aluvchi nuqtali sonlar – eksponensial ko'rinishda ifodalangan sonlardir. Sonlarni bu usulda yozish juda kichik yoki juda katta sonlarni ifodalashda qo'l keladi.

O'zgaruvchi miqdorlar.

Dasturning bajarilish jarayonida qiymati o'zgaradigan miqdorlar **o'zgaruvchi miqdorlar** yoki qisqacha **o'zgaruvchilar** deyiladi. O'zgaruvchilar 63 ta belgidan oshmaydigan o'z nomiga ega bo'ladi. Paskalda o'zgaruvchi nomi lotin harfi yoki tagchiziq (_) belgisidan boshlanadi. O'zgaruvchi nomida faqatgina raqamlar, lotin harflari va tagchiziq belgisi ishlatilishi mumkin. Masalan:

abc; _son; mening_ismim; sinf_9.

Paskalda o'zgaruvchilarning nomida ishlatilgan katta va kichik lotin harflari farqlanmaydi. Masalan, *karra*, *Karra*, *kArRa* nomlar bitta o'zgaruvchini bildiradi. Bunday yozish Paskalning xizmatchi so'zlari uchun ham o'rinlidir. Chunki, Paskal translyatori dasturni kompilyatsiya qilish (dasturni mashina tiliga o'girish) vaqtida o'zgaruvchilarning nomlari va Paskalning xizmatchi so'zlaridagi barcha katta harflarni kichik harflarga almashtirib oladi.

O'zgaruvchilar dasturning tavsiflash qismida albatta tavsiflanishi, yani ularning turi ko'rsatilgan bo'ishi lozim. Dasturda o'zgaruvchilarni tavsiflash Paskalning **Var** xizmatchi so'zi bilan boshlanadi:

Var

o'zgaruvchi : turi;

o'zgaruvchi : turi;

...

Agar bir nechta o'zgaruvchining turi bir xil bo'lsa, ularni alohida tavsiflamasdan, birgalikda tavsiflash ham mumkin:

Var

1-o'zgaruvchi, 2-o'zgaruvchi, ..., n-o'zgaruvchi : turi;

Butun sonli qiymatlar qabul qiladigan o'zgaruvchilar **butun sonli o'zgaruvchilar** deyiladi. Ular 5 turga bo'linib, bir-biridan qabul qiladigan qiymatlarining chegarasi (diapozoni) va kompyuter hotirasidan egallaydigan joyi (hajmi) bilan farqlanadi. Quyidagi jadvalda butun sonli o'zgaruvchilarni tavsiflash uchun Paskalning maxsus so'zlari, ularga mos qiymatlar chegarasi va egallaydigan hotira hajmi keltirilgan:

Turi	Qiymatlar chegarasi	Egallaydigan hotira hajmi
ShortInt	-128 ... 127	8 bit
Integer	-32768 ... 32767	16 bit
LongInt	-2147483648 ... 2147483647	32 bit

Byte	0 ... 255	8 bit
Word	0 ... 65535	16 bit

Misol:

Var

i, j : Integer;

tartib_raqam : Byte;

Haqiqiy sonli qiymatlar qabul qiladigan o'zgaruvchilar **haqiqiy sonli o'zgaruvchilar** deyiladi. Ularning turlari quyidagi jadvalda keltirilgan:

Turi	Qiymatlar chegarasi	Razryadi	Egallaydigan hotira hajmi
Real	$-2,9 \cdot 10^{39} \dots 1,7 \cdot 10^{38}$	11-12	6 bayt
Single	$-1,5 \cdot 10^{45} \dots 3,4 \cdot 10^{38}$	7-8	4 bayt
Double	$-5,0 \cdot 10^{324} \dots 1,7 \cdot 10^{308}$	15-16	8 bayt
Extended	$-3,4 \cdot 10^{4932} \dots 1,1 \cdot 10^{4932}$	19-20	10 bayt
Comp	$-9,2 \cdot 10^{18} \dots 9,2 \cdot 10^{18}$	19-20	8 bayt

Jadvaldagi "Razryadi" sonning aniq raqamlari sonini bildiradi. Juda ko'p hollarda **real** turli o'zgaruvchilardan foydalanish yetarli bo'ladi.

Misol:

var

burchak, yoy_uzunligi : Real;

daraja : Single;

Belgili o'zgaruvchilar qiymatini qabul qiluvchi o'zgaruvchilar **belgili o'zgaruvchilar** deyiladi. Ular Paskalning **Char** maxsus so'zi yordamida tavsiflanadi.

Misol:

var

harf, belgi : char;

Satrlı o'zgaruvchilarni tavsiflash uchun Paskalning **String** maxsus so'zi qo'llaniladi. Bunday o'zgaruvchilar uchun kompyuter hotirasidan 256 bayt (256 ta belgi uchun) joy ajratiladi. Agar satrlı o'zgaruvchi qabul qiladigan satrdagi belgilar soni dastur ishlashi davomida ma'lum miqdordan, masalan, 10 ta belgidan oshmasa, kompyuter hotirasini tejash maqsadida, uni String[10] orqali tavsiflash maqsadga muvofiq.

Misol:

var

qator : String; {qator nomli o'zgaruvchiga hotiradan 256 bayt ajratildi}

_satr : String[24]; {_satr nomli o'zgaruvchiga hotiradan 24 bayt ajratildi}

Mantiqiy o'zgaruvchilar Paskalning **Boolean** maxsus so'zi orqali tavsiflanadi.
Misol:

var

natija : Boolean;

katta, kichik : Boolean;

Paskalda tilida tuzilgan dasturda faqat tavsiflangan o'zgaruvchilar ishtirok etishi mumkin. Paskal translyatori tavsiflanmagan o'zgaruvchilarni dasturda ishlatishga yo'l qo'ymaydi, ya'ni dastur bajarilmaydi va ekranga "Unknown identifier" (noma'lum **identifikator** ya'ni noma'lum o'zgaruvchi) yozuvi chiqadi. Shuni yodda tutish zarurki, o'zgaruvchilarga faqat tavsifda ko'rsatilgan turdagi qiymatlarnigina berish mumkin bo'ladi.

Standart funksiyalar va algebraik ifodalar

Dastur translyatori ta'minotiga kiritilgan funksiyalar **standart funksiyalar** deb yuritiladi. Paskal dasturlash tilida standart funksiyalar bilan birga **standart protseduralar** ham foydalaniladi. Bu ikkalasining farqi shundaki, standart funksiya biror qiymatni hisoblasa, standart protsedura ma'lum bir amalni bajaradi.

Quyida Paskalning ba'zi standart funksiyalari va protseduralarini izoh bilan keltiramiz:

Funksiya nomi	Argument turi	Qiymat turi	Izoh
Matematik protseduralar			
Inc(x)	butun	butun	"x" ni bittaga orttiradi ($x=x+1$)
Dec(x)	butun	butun	"x" ni bittaga kamaytiradi ($x=x-1$)
O'zgaruvchilar turini o'zgartiruvchi funksiyalar			
Chr(x)	Byte	Char	"x" ASCII kodga mos belgi
Ord(c)	Char	Byte	"c" belgining ASCII kodi

O'zlashtirish va ma'lumotlarni ekranga chiqarish operatorlari

O'zlashtirish operatori. O'zlashtirish operatori o'zgaruvchilarga qiymat berish uchun ishlatiladi. U **:=** belgi bilan ifodalanadi. O'zlashtirish operatorning umumiy ko'rinishi quyidagicha:

o'zgaruvchi := algebraik ifoda;

Bu operator bajarilganda algebraik ifoda hisoblanib, uning qiymati o'zgaruvchiga beriladi.

Ma'lumotlarni kompyuter ekraniga chiqarish uchun **chiqarish operatoridan** foydalaniladi. Paskalda chiqarish operatori quyidagi ikki xil ko'rinishga ega:

Write(chiqarish ro'yxati)

WriteLn(chiqarish ro'yxati),

bu yerda *Write* (ing. – yozish) - Paskalning xizmatchi soʻzi; chiqarish roʻyxati – oʻzaro vergul bilan ajratilgan va ekranga chiqarilishi kerak boʻlgan ifoda, oʻzgaruvchi yoki oʻzgarmaslar ketma-ketligi. Chiqarish roʻyxatida ifoda qatnashsa, avval u hisoblanib, hosil boʻlgan natija ekranga chiqariladi. Chiqarish roʻyxatidagi oʻzgarmaslar belgisi yoki satrli boʻlsa, albatta apostrof ichiga olinishi shart.

Write va *Writeln* operatorlarining farqi shundaki, *Write* operatori yordamida maʼlumotlar ekranga chiqarilgach yurgich ekranning shu satrida qoladi, yaʼni ekranga keyingi chiqariladigan maʼlumotlar shu satrga (yurgich turgan joydan) chiqariladi. *Writeln* operatorida esa maʼlumotlar ekranga chiqarilgach yurgich keyingi satr boshiga oʻtadi.

Maʼlumotlarni hotiraga muloqot usulida kiritish operatori

Paskalda oʻzgaruvchilarga qiymat berishning oʻzlashtirish operatoridan tashqari boshqa usullari ham bor. Ulardan biri maʼlumotlarni hotiraga muloqot usulida kiritish deyiladi va **kiritish operatori** yordamida amalga oshiriladi. Kiritish operatoridan oʻzgaruvchilarga dastur bajarilishi davomida kompyuter klaviaturasidan qiymat berish uchun qoʻllaniladi. Mazkur operator quyidagi ikki xil koʻrinishga ega:

Read(kiritish roʻyhati)

ReadLn(kiritish roʻyhati),

bu yerda ***Read*** va ***ReadLn*** – Paskalning xizmatchi soʻzlari (*read* (ing.) – oʻqish), kiritish roʻyhati esa bitta yoki oʻzaro vergul bilan ajratilgan bir nechta oʻzgaruvchidan iborat. Masalan,

Read(a); Read(alfa,betta); ReadLn(_name);

Kiritish operatori dastur ishlashini toʻxtatadi va roʻyhatdagi oʻzgaruvchilarga klaviatura orqali qiymat berilishini kutadi. Agar roʻyhatda bir nechta oʻzgaruvchi boʻlsa, ularning qiymatlari oʻzaro probel (boʻsh joy) bilan ajratib kiritiladi. Barcha qiymatlar kiritib boʻlingach <ENTER> klavishi bosiladi.

Read va *Readln* operatorlarining farqi shundaki, *Read* operatori yordamida maʼlumotlar kiritilgach yurgich ekranning shu satrida qoladi, yaʼni kiritiladigan maʼlumotlar shu satrda (yurgich turgan joyda) kiritiladi. *Readln* operatorida esa maʼlumotlar kiritilib boʻlingach yurgich keyingi satr boshiga oʻtadi.

Oʻtish va tarmoqlanish operatorlari

Biz hozirgacha chiziqli, yaʼni buyruqlari ketma-ket bajariladigan dasturlar bilan tanishdik. Lekin, koʻpincha berilgan masalani hal qilishda operatorlarning bajarilish tartibini buzishga, yaʼni boshqarishni dastur boʻyicha orqaga yoki oldinga oʻtkazish zarur boʻladi. Buning uchun dasturning boshqarish uzatilayotgan operatoriga **nishon** qoʻyiladi. Nishon oʻzgaruvchining nomi singari lotin harflari va raqamlar yordamida hosil qilinadi. Masalan, 7, N1, nishon2 va hokazo.

Dasturda qoʻllaniladigan nishonlar dasturning tavsif qismida **Label** xizmatchi soʻzi yordamida koʻrsatilishi shart.

Nishonlardan dasturda **o'tish operatori** qo'llanilsagina foydalaniladi. O'tish operatori quyidagi ko'rinishga ega:

GOTO <nishon> .

Bu holda **GOTO** (ing. – “o'tilsin”) operatori boshqarishni dasturning **nishon** qo'yilgan operatoriga uzatadi.

O'tish operatorida hech qanday shart tekshirilmasdan boshqarish ko'rsatilgan nishonli operatorga uzatiladi. Lekin aksariyat masalalarni hal etishda biror shartning bajarilishiga qarab u yoki bu amallar ketma-ketligini bajarish kerak bo'ladi. Masalan, kvadrat tenglamani yechishda natijani hisoblash uchun diskriminantning ishorasiga qarab uch yo'nalishdan biri tanlanadi. Shu kabi masalalarni hal qilish uchun Paskalda **tarmoqlanish operatori** qo'llaniladi.

Tarmoqlanish operatori. Bu operatorning umumiy shakli quyidagicha:

If <shart> **Then** <operator yoki operatorlar> **Else** <operator yoki operatorlar>;

Bu yerda **if**, **then** va **else** Paskalning hizmatchi so'zlari bo'lib, quyida ularning o'qilishi va ma'nosi keltirilgan:

If (if) – “agar”, **Then** (zen) – “u holda”, **Else** (elz) – “aks holda”;

<shart> - rost yoki yolg'on qiymatlardan birini qabul qiluvchi mantiqiy ifoda;

<operator yoki operatorlar> - Paskalning ixtiyoriy operatori yoki operatorlari ketma-ketligi bo'lishi mumkin. Ularni bitta satrga yozish shart emas. Agar operatorlar ketma-ketligi yozilgan bo'lsa, bu ketma-ketlik albatta *begin* hizmatchi so'zi bilan boshlanib, *end* hizmatchi so'zi bilan tugallanishi lozim. **Else** hizmatchi so'zidan avval yozilgan operatoridan keyin “;” (nuqtali vergul) qo'yilmaydi.

Tarmoqlanish operatori quyidagicha ishlaydi:

Avval **shart** tekshiriladi, agar uning qiymati **rost** bo'lsa **then** dan keyingi operator yoki operatorlar ketma-ketligi, aks holda **else** dan keyingi operator yoki operatorlar ketma-ketligi bajariladi.

Tarmoqlanish operatorining **Else** qismi zaruratga qarab ishlatiladi. Ya'ni, tarmoqlanish operatorini quyidagi shaklda ham ishlatish mumkin:

If <shart> **Then** <operator yoki operatorlar>

Bu tarmoqlanish operatorining qisqa shakli deyiladi. Bu holda **shart** ning qiymati rost bo'lsa, **Then** dan keyingi operator yoki operatorlar ketma-ketligi bajariladi, aks holda boshqarish navbatdagi (tarmoqlanish operatoridan keyingi) operatorga o'tadi.

Parametrli takrorlash operatori

Takrorlanuvchi dasturlar tuzish uchun takrorlash operatorlaridan foydalaniladi. Paskalda ular uchta bo'lib, bu darsda ulardan bittasi – **parametrli takrorlash operatori** bilan tanishamiz. Mazkur operatorning umumiy ko'rinishi quyidagicha:

For I: = N1 To N2 Do <operator>;

bu yerda **For** (uchun), **To** (gacha) va **Do** Paskalning hizmatchi so'zlari; **I** – butun turli ixtiyoriy o'zgaruvch bo'lib, u **takrorlash parametri** deyiladi; **N1** – takrorlash parametrining qabul qiladigan boshlang'ich qiymati; **N2** – takrorlash

parametrining qabul qiladigan oxirgi qiymati; **<operator>** – takrorlanishi lozim bo‘lgan operator yoki operatorlar ketma-ketligi bo‘lib, u **takrorlash tanasi** deyiladi. Takrorlash tanasini operatorlar ketma-ketligi tashkil etgan bo‘lsa, ular albatta *begin* ko‘rsatmasi bilan boshlanib, *end*; ko‘rsatmasi bilan tugallanadi. Takrorlash parametrining boshlang‘ich va oxirgi qiymatlari o‘zgarmas, o‘zgaruvchi yoki ifoda ko‘rinishida bo‘lishi mumkin.

Mazkur operator quyidagicha ishlaydi:

avval takrorlash parametri boshlang‘ich qiymatni qabul qiladi;

agar takrorlash parametrining qiymati oxirgi qiymatdan katta bo‘lmasa, takrorlash tanasini tashkil etuvchi operatorlar bajariladi, aks holda takrorlanish to‘xtatiladi va boshqarish navbatdagi operatorga uzatiladi;

takrorlash parametrining qiymati bittaga ortadi (unga 1 qo‘shiladi) va 2-bo‘limga o‘tiladi.

For operatoridan asosan takrorlanishlar soni avvaldan ma’lum bo‘lganda qo‘llaniladi.

Shart bo‘yicha takrorlash operatorlari

Avvalgi darsda ko‘rgan misollarimizda takrorlanishlar soni aniq edi. Ammo shunday masalalar ham uchraydiki, unda biror amallar ketma-ketligini ma’lum bir shart bajarilmaguncha takrorlash kerak bo‘ladi. Bunda takrorlanishlar sonini avvaldan aytib bo‘lmaydi. Bunday hollarda **shart bo‘yicha takrorlash** operatorlaridan foydalaniladi. Paskalda bunday operatorlardan ikkita: **While** va **Repeat** operatorlari.

While operatori quyidagi umumiy ko‘rinishga ega:

While <shart> Do <operator>;

bu yerda **While** va **Do** Paskalning hizmatchi so‘zlari; **<shart>** - oddiy yoki murakkab mantiqiy ifoda; **<operator>** – takrorlash tanasini tashkil etuvchi operator yoki operatorlar ketma-ketligi. Agar takrorlash tanasida operatorlar ketma-ketligi yozilsa, ular **begin** bilan boshlanib, **end**; bilan yakunlanadi.

Mazkur takrorlanish operatori quyidagicha ishlaydi:

Avval **<shart>** tekshiriladi. Agar uning qiymati “rost” bo‘lsa, takrorlash tanasini tashkil etuvchi operatorlar ishlaydi, va yana **<shart>** teksiriladi. Bu jarayon shart “yolg‘on” qiymat qabul qilgunga qadar davom etadi.

Repeat operatori ham shart bo‘yicha takrorlashni amalga oshiradi. Uning umumiy ko‘rinishi quyidagicha:

Repeat

<operatorlar>

Until <shart>

bu yerda *repeat* va *until* Paskalning xizmatchi so‘zlari bo‘lib, *Repeat* – takrorlash boshi, *Until* – takrorlash oxirini bildiradi; **<shart>** – mantiqiy ifoda; **<operatorlar>** - ixtiyoriy operatorlar bo‘lib, takrorlash tanasini tashkil etadi. Ular **<shart>** bajarilmaguncha (rost qiymat qabul qilmaguncha) takrorlanaveradi.

Repeat operatorining While operatoridan farqi shundaki, While operatorida shart takrorlash boshida tekshirilsa, Repeat operatorida shart takrorlash oxirida

tekshiriladi. Shu sababli While operatorida takrorlash tanasini tashkil etuvchi operatorlar bir marta ham bajarilmasligi mumkin (shart avvaldan bajarilmasa), Repeat operatorida esa hech bo'lmaganda bir marta bajariladi. Bundan tashqari While operatorida takrorlash sharti bajarilmaganda ("yolg'on" qiymat qabul qilganda) yakunlansa, Repeat operatorida shart bajarilganda ("rost" qiymat qabul qilganda) yakunlanadi.

AMALIY QISM.

Masalaning qo'yilishi va uning yechish algoritmi

Masala: O'quvchilarning karra jadvalini qay darajada o'zlashtirganini aniqlovchi dastur tuzish.

Ishni boshlashdan avval dasturning interfeysini tasavvur qilib olaylik:

- Dasturning nomi e'lon qilinsin, masalan: "Karra sinovi";
- Sinovlar soni aniqlansin;
- Sinov jarayonida javob berishga urinish 3 martagacha amalga oshirilsin. Foydalanuvchi imkoniyatdan foydalana olmagan taqdirda sinov to'xtatilsin;
- Sinov tugagach natija e'lon qilinsin (urinishlar soni, samaradorlik);
- Sinovni davom ettirish haqida taklif berilsin.

Endi esa bevosita dasturlashga o'tamiz.

Avvalo dasturda qo'llaniladigan o'zgaruvchilarni e'lon qilib olaylik. Bizga sikl, sinovlar soni, birinchi ko'paytuvchi, ikkinchi ko'paytuvchi, kiritilgan javob, urinishlar soni va javob berish imkoniyati uchun bittadan integer tipidagi, foydalanuvchi bilan muloqot qilishda bosilgan tugmani aniqlash uchun o'zgaruvchi kerak bo'ladi:

```
var
  i, ss, k, j, l, urin, imk: integer;
  d: char;
```

Har safar sinov yangidan boshlanishidan avval ekran tozalab olinishi kerak. Buning uchun Clrscr operatoridan foydalanamiz. Ushbu operator ishlashi uchun **crt** moduli talab qilingani sababli modullar e'lon qilish bo'limi (**Uses**) da mazkur modul e'lon qilinadi:

```
Uses crt;
Begin
  Clrscr;
End.
```

Shundan so'ng ekranga dastur nomini chiqarib, uni dasturning ishchi qismidan to'g'ri chiziq va bo'sh qator yordamida ajratib qo'yamiz:

```
...
writeln('Karra sinovi:');
writeln('_____');
writeln;
```

Dasturning ishchi qismi sinovlar sonini so'rash va kiritilgan qiymatni sinovlar soni (ss) sifatida qabul qilish bilan boshlanadi:

```
...
Writeln('Sinovlar sonini kiriting:');
Readln(ss);
```

Sinovlar soni musbat sonlarda ifodalangani uchun uning qiymati tekshiriladi, agar 0 dan kichik son bo'lsa boshqaruv dastur boshiga uzatiladi. Buning uchun metkalar bo'limi (label) da metka e'lon qilinadi:

```
uses crt;
label 3;
var
...

```

Endi esa tekshirish amalga oshiriladi:

```
3: clrscr;
   writeln('Karra sinovi:');
   writeln('_____');
   writeln;
   writeln('Sinovlar sonini kiriting:');
   readln(ss);
   if ss<=0 then goto 3; // ss ning qiymati 0 dan kichik bo'lsa, boshqaruv 3
ga uzatiladi
...

```

Kiritilgan qiymat to'g'ri bo'lgan holda dastur davom ettiriladi va ko'paytuvchilar tasodifiy sonlardan tashkil topishi uchun tasodifiylik generatori faollashtiriladi.

```
...
randomize;
```

Beriladigan topshiriqlar soni ss ga bog'liq bo'lgani uchun sikl ochiladi:

```
...
for i:=1 to ss do
begin
end;
```

Va shu sikl tarkibida ko'paytuvchilar aniqlanadi. Ko'paytuvchilar (k, l) tasodifiy sonlardan tashkil topishi uchun **random** operatoridan foydalanamiz:

```
...
k:=random(11);
l:=random(11);
```

Bunda k va l ning qiymati 0 dan 10 gacha bo'lgan tasodifiy sonlardan tashkil topadi. Hali topshiriq berilmagani va javob berishga urinish amalga oshirilmagani uchun o'zgarishlar qiymatini 0 ga tenglaymiz:

```
...  
imk:=0;
```

Endi esa ekranda topshiriq tartib raqami bilan birgalikda e'lon qilinadi va javob qabul qilinadi:

```
...  
write(i, ' ', k, '*', l, '=');
```

Javob to'g'ri yoki noto'g'ri bo'lishidan qat'iy nazar urinishlar sonini bittaga oshiriladi:

```
...  
inc(urin);
```

Kiritilgan javob to'g'ri yoki noto'g'ri ekanligi tekshiriladi. Agar javob noto'g'ri bo'lgan taqdirda xato haqida habar berilib, javob berish imkoniyatining qiymati bittadan oshirib boriladi:

```
if j<>k*l then  
    begin  
        inc(imk);  
        writeln('Xato');  
    end;
```

Agar javob berish imkoniyati 3 taga yetib qolgan holda bu haqida habar beriladi aks holda javobni boshqatdan kiritish uchun boshqaruv javob kiritiladigan qatorga uzatiladi:

```
...  
if imk=3 then  
    begin  
        writeln;  
        writeln('_____');  
        Writeln('Uch imkoniyatdan foydalanib bo"ldingiz!');  
        writeln('Sinov tugadi');  
        writeln('_____');  
    end  
else  
    goto 1;
```

goto 1 ifoda ishlashi uchun metkalar bo'limi va javob kiritish qatoriga tegishli o'zgarishlar kiritiladi:

```
...  
Label 1, 3;  
...  
Begin  
...
```

```
1: Readln(ss);
```

```
...
```

```
End.
```

Javob to'g'ri kiritilgan holdan sikl qiymati oshib boradi va sikl tugagach, sinov tugatiladi va bu haqida habar beriladi:

```
...
```

```
writeln;
```

```
writeln('_____');
```

```
writeln('Sinov tugadi');
```

Sinov yakuniga yetgani uchun natijalar e'lon qilinadi. Bunda samaradorlik foiz hisobida aniqlanadi. Samaradorlikni aniqlash uchun 100 % ni urinishlar soniga bo'lib, sinov soniga ko'paytirish kifoya. Bo'lish natijasini butun songacha yahlitlash uchun **round** funksiyasidan foydalanish mumkin. Urinishlar soni **urin** o'zgaruvchisining qiymati yordamida aniqlanadi:

```
...
```

```
writeln;
```

```
writeln('Samaradorlik: ',round((100/urin)*ss),' %');
```

```
writeln('Urinishlar soni: ',urin);
```

```
writeln('_____');
```

Shu bilan dasturni tayyor deb hisoblashimiz mumkin. Lekin keeling sinovlar tugagandan so'ng uni qaytadan boshlash imkoniyatini kiritib ketaylik. Buning uchun yuqorida char tipiga oid d o'zgaruvchisi e'lon qilingan. Avval ekranda davom ettirish taklifi berilsin:

```
...
```

```
writeln('Davom etasizmi? Ha - Enter');
```

Shundan so'ng bosilgan tugma aniqlansin:

```
...
```

```
d:=readkey;
```

Agar tugmaning kodi 13 ga teng bo'lsa (Enter tugmasining tartib raqami - 13) dastur ishni davom ettirishi uchun boshqaruv dastur boshiga, aks holda, ya'ni boshqa ixtiyoriy tugma bosilgan holda boshqaruv dastur oxiriga uzatilsin. Bosilgan tugmaning kodini aniqlash uchun **ord** funksiyasidan foydalanish mumkin:

```
...
```

```
if ord(d)=13 then goto 3;
```

```
2:
```

```
end.
```

Javob berish imkoniyatlari soni 3 tadan oshib ketganda ham sinov to'xtatilgani uchun yuqoridagi uch qator kod biroz o'zgartirilgan holda imkoniyatdan foydalanib bo'lgach sinov to'ztatilgan qatordan keyin ham yoziladi:

```
...
if imk=3 then
    begin
        writeln;
        writeln('_____');
        Writeln('Uch imkoniyatdan foydalanib bo"ldingiz!');
        writeln('Sinov tugadi');
        writeln('_____');
        Writeln('Davom etasizmi? Ha - Enter');
        d:=readkey;
        if ord(d)=13 then goto 3 else goto 2;
    end
end
```

Dasturning to'liq kodi

```
uses crt;
label 1, 2, 3;
var
  i, ss, k, j, l, urin, imk:integer;
  d:char;
begin
3:  clrscr;
   Writeln('Karra sinovi:');
   writeln('_____');
   writeln;
   Writeln('Sinovlar sonini kiriting:');
   Readln(ss);
   if ss<=0 then goto 3;
   randomize;
   for i:=1 to ss do
   begin
     k:=random(11);
     l:=random(11);
     imk:=0;
1:  write(i, ' ', k, '*', l, '=');
     readln(j);
     inc(urin);
     if j<>k*l then
     begin
       inc(imk);
       writeln('Xato');
       if imk=3 then
       begin
         writeln;
         writeln('_____');
         Writeln('Uch imkoniyatdan foydalanib bo"ldingiz!');
         writeln('Sinov tugadi');
         writeln('_____');
         Writeln('Davom etasizmi? Ha - Enter');
         d:=readkey;
         if ord(d)=13 then goto 3 else goto 2;
       end
     else
       goto 1;
     end;
   end;
   writeln;
   writeln('_____');
```

```
writeln('Sinov tugadi');  
writeln;  
Writeln('Samaradorlik: ',round((100/urin)*ss),' %');  
writeln('Urinishlar soni: ',urin);  
writeln('_____');  
Writeln('Davom etasizmi? Ha - Enter');  
d:=readkey;  
if ord(d)=13 then goto 3 else goto 2;  
2:  
end.
```

FOYDALANILGAN ADABIYOTLAR.

1. S.S. G`ulomov, B.A. Begalov; Informatika va axborot texnologiyalari: Oliy ta`lim muassasalari talabalari uchun darslik. O`z.R Oliy va o`rta maxsus ta`lim vazirligi, Toshkent Davlat Iqtisodiyot Universiteti.-T.: Fan,2010.-704 bet
2. N.A.Otaxanov. "Programmash asoslari" fani bo'yicha ma'ruzalar matni. Namangan, 2005 y
3. Informatika. Akademik S.S.G`ulomov umumiy tahriri ostida. Darslik. Toshkent. – TDIU, 2007.
4. R.Alimov, B.Xodiev va boshqalar. "Axborot texnologiyasi" O'quv qo'llanma. T.: - TDIU, 2004
5. B.E.Yuldashev, M.T.Eshmuradov, M.Ko'kanova. "Turbo Pacalda misol va masalalar". O'R.F.A. "Fan", 2008 y