O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS TA'LIM VAZ' 'LIGI

AJINIYOZ NOMIDAGI NUKUS DAVLAT PEDAGOGIKA INSTITUTI

Matematika-informatika fakulteti

"Informatika o'qitish metodikasi" kafedrasi

5110700 Informatika oqitish metodikasi bakalavriat ta'lim yo'nalishi 4[b]kurs talabasi Sultanov Raxmatjonning

BITIRUV MALAKAVIY ISHI

Mavzu: TURBO PASCAL tilida sonlar nazariyasining asosiy algoritmlarini dasturlash metodikasi

Talaba:		R.Sultanov
Ilmiy rahbar:		phD G.Abilova
Kafedra mudiri:		f.-m.f.n. M.Alaminov

Kafedra majlisining 2019-yil __-may sanasidagi №___bayonnomasi bilan himoyaga ruxsat berildi

Nukus 2019

MUNDARIJA:

KIRISH

So‘nggi vaqtlarda dasturlash zarurati va bu yo‘l bilan yechimi topiladigan masalalar soni ortib bormoqda. Sababi sifatida kundalik hayotimizga axborot texnologiyalarining kirib kelishidir. Agar inson kompyuter bilan ish olib boradigan

bo‘lsa, ertami-kechmi u dasturlashtirishga qiziqadi yoki dasturlashni o‘rganishga majbur bo‘ladi.

Bu masalaning yechimi sodda Paskal kabi dasturlash tillaridan foydalanish orqali topiladi.

Oliy o‘quv yurtiga o‘qishga qabul qilingan talabalar kollejlarda bitta algoritmik tilini o‘rganadilar : Paskal. Paskal tilida oldindan ma’lum bo‘lgan algoritmlarni tahlil qilish dasturlash texnologiyasini egallashga yordam beradi. Bu algoritmlar guruhining asosiy qismini sonlar nazariyasining asosiy algoritmlari egallaydi.

Shuning uchun bu malakaviy bitiruv ishida sonlar nazariyasining asosiy algoritmlarini dasturlash texnologiyasi ko‘rib chiqildi.

Bitiruv ishi kirish, 2 bob, xulosa va adabiyotlar ro‘yxatidan iborat. Birinchi bobda Paskal dasturlash tili haqida batafsil ma’lumot beriladi. U 2 paragrafdan iborat bo‘lib, birinchi paragrafda paskal dasturlash tili haqida umumiy ma’lumotlar beriladi. Ikkinchi paragrafda Paskal tilida dasturlash texnologiyasi ko‘rib chiqiladi.

Ikkinchi bobning birinchi paragrafida sonlar nazariyasining asosiy tushuncha va algoritmlari yoritiladi. Ikkinchi paragrafda ushbu algoritmlarni dasturlash texnologiyasi keltiriladi.

Tadqiqotimizning asosiy maqsadi Paskal algoritmik tilini o‘rganuvchi talabalarga dasturlash texnologiyalarini o‘rgatishdan iborat.

I BOB. PASKAL TILI HAQIDA UMUMIY MA’LUMOTLAR

1.1. Paskal dasturlash tili

Hozirda insoniyat faoliyatining barcha jabhalariga shaxsiy elektron hisoblash mashinalari (SHEXM) shaxdam qadamlar bilan kirib bormoqda. Asosan SHEXMlarga mo‘ljallangan, hamda murakkab jarayonlarning hisob ishlarini

bajarish va juda katta ma’lumotlar tizimi bilan ishlashni tashkil etuvchi yangi algoritmik tillar sinfi borgan sari kengayib bormoqda. Bu tillar jumlasiga quyidagi ko‘proq ishlatilayotgan tillarni kiritish mumkin: Paskal tili; C tili va hokazo.

Paskal tili 1969 yili N.Virt tomonidan yaratilib mashhur olim Blez Paskal nomi bilan ataldi. Bu til N.Virtning o‘ylashi bo‘yicha dasturlashning zamonaviy texnologiyasiga va uslubiga, strukturali dasturlash nazariyasiga asoslangan va boshqa dasturlash tillaridan muayyan yutuqqa ega til bo‘lishi lozim edi.

Operator tushunchasi Paskal tilining eng asosiy tushunchalaridan biri bo‘lib, har bir operator tilning yakunlangan jumlasi hisoblanadi va ma’lumotlar tahlilining tugallangan bosqichini ifodalaydi.

Operatorlarni ikki guruhga ajratish mumkin. 1-guruh operatorlarining tarkibida boshqa operatorlar qatnashmaydi va bu operatorlarni asosiy operatorlar deb ataladi. Asosiy operatorlar jumlasiga quyidagi operatorlar kiradi: o‘zlashtirish operatori, prosedura operatori, o‘tish operatori, bo‘sh operator. 2-guruh operatorlarining tarkibida esa boshqa operatorlar ham qatnashib, ular tarkibiy operatorlar deb ataladi. Ular jumlasiga quyidagi operatorlar kiradi: tashkiliy operator, tanlov operatori, takrorlash operatori, ulash operatori.

Masalani yechish algoritmida yuqoridagi ikki guruh operatorlarining ketma-ketligi cheklanmagan miqdorda qatnashishi mumkin. Bu ketma-ketlikdagi operatorlar o‘zaro „ ; “ ajratish belgisi orqali ajratiladi, ya’ni dastur matnining yozuvi alohida operatorlarga bo‘linadi. Shunday qilib, S orqali ixtiyoriy yozish mumkin bo‘lgan operatorni belgilasak, masala yechilishining algoritmi quyidagi ketma-ketlik bo‘yicha ifodalanishi mumkin:

S; S; ...;S.

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

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

<identifikator>::=<harf>|<identifikator><harf>|<identifikator><raqam>

Agar quyidagi oraliq tushunchani kiritsak:

<harf yoki raqam>::= <harf>|<raqam>

Yuqoridagi aniqlashni quyidagicha ham yozish mumkin:

<identifikator>::= <harf> {<harf yoki raqam>}.

Xizmatchi so‘zlardan identifikator sifatida foydalanish mumkin emas. Odatda identifikator so‘zining o‘rniga qulayroq va qisqaroq qilib ism deyish mumkin. Dasturda qatnashuvchi obyektlarga ismlarni dastur tuzuvchi o‘z ixtiyoriga ko‘ra tanlab olishi mumkin. Bir xil ism bilan bir necha xil obyektlarni nomlash mutlaqo mumkin emas. Turbo Pascal muhitida ismda qatnashuvchi belgilar soni (ism uzunligi) 63 ta belgidan oshmasligi kerak.

Ismlarga misollar sifatida quyidagilarni keltirsak bo‘ladi: *_Burchak, _A1, Ahmad_Berdiev, C, Summa, Time, A, S1, …*

Paskal tilining asosiy tushunchalaridan biri e’lon qilish hisoblanadi. Dasturda qatnashuvchi barcha obyektlarning ismlari mos ravishda dasturning bosh qismida, ularning qanday tipdagi qiymatlar qabul qilishi mumkinligiga qarab, e’lon qilinib qo‘yilishi kerak. Paskal tilida e’lon qilishning 5 xil turi mavjud:

- metkalar e’loni;
- o‘zgarmaslar e’loni;
- tip aniqlash uchun e’lon;
- o‘zgaruvchilar e’loni;
- prosedura va funksiyalar e’loni.

Umuman olganda, yuqorida sanab o'tilgan e’lonlarning vazifalari ularning nomlaridan ham sezilib turibdi, e’lonning vazifalari esa keyinroq to'la ochib beriladi.

O'zgaruvchi, dastur obyekti bo'lib, turli xil qiymatlarni xotirada ma’lum nom bilan saqlab turish uchun ishlatiladi. O'zgaruvchi o'z qiymatini dasturni bajarilish davomida, o'zlashtirish operatori yordamida qabul qiladi. qabul qilingan qiymat, o'zgaruvchiga boshqa yangi qiymat berilmaguncha saqlanib turiladi va yangi qiymat berilishi bilan eski qiymat butunlay o'chib, yo'q bo'lib ketadi. Har bir o'zgaruvchiga ma’lum bir tipga tegishli qiymatlarnigina qabul qilish huquqi beriladi. Boshqa tipdagi qiymatlarni o‘zlashtirishga urinish dasturning xatoligini ta’minlaydi.

O'zgaruvchi ─ bu identifikatordir. Uning ismi o'zgaruvchining qiymatiga murojaat qilishda ishlatiladi. Boshqacha aytganda, dastur matnidagi ism, shu o' zgaruvchining qiymatini ifodalaydi.

Algoritmik tillarda, faqat qiymatini hisoblash algoritmlari ma’lum bo'lgandgina funksiyalar ishlatiladi. Dastur tuzuvchi dastur uchun lozim bo'lgan keraklicha funksiyalarni o'z dasturiga kiritishi mumkin.

Xuddi funksiyalar kabi hal qilinayotgan masalaning ma’lum bir tugallangan bosqichlarini hisoblash vazifasini proseduralar zimmasiga yuklasa ham bo'ladi. Funksiyani hisoblash natijasida faqat, yagona natijaviy qiymatga erishiladi, proseduradan foydalanganda esa, natijaviy qiymatlar soni yetarlicha ko'p bo'lishi mumkin.

Dasturda aniqlangan funksiya va proseduralar o'zgaruvchilarning e’loni bo'limida e’lon qilinib qo‘yilishi kerak. Bunda har bir funksiya va proseduraga, ularning bajaradigan vazifasiga mos ismlar berib qo'yiladi. Ularni aniqlashda formal parametrlardan foydalaniladi. O'z navbatida, bu parametrlarning tiplari funksiya va proseduraning ichida aniqlanilib, e’lon qilinadi.

Dasturda aniqlangan funksiya va proseduralardan foydalanish uchun dastur matnida ularning ismlari va formal parametrlarga mos, faktik parametrlari berilishi kerak.

Ma’lumki, matematika kursidagi elementar funksiyalardan dastur tuzishda juda ko'p foydalanishga to'g'ri keladi. Masalan $\sin x, \cos x, \ln x, e^x$ va hokazo. Bunday funksiyalarni boshqa tillardagi kabi standart funksiyalar deb ataladi va standart funksiyalarning ismlaridan boshqa maqsadda foydalanish maqsadga muvofiq emas.

Har bir algoritmik tilning dastur matnini yozish qoidalari turlicha bo'ladi. Dasturlash tillaridan eng soddasi Beysik tilining ma’lum versiyalarida, dasturning har bir operatori qat’iy aniqlangan qator nomerlari orqali yoziladi. Paskal tilida esa, operatorlar ketma-ket yozilib, o'zaro ";" belgisi bilan ajratib boriladi. Bundan tashqari, yozilgan dasturning o'qishga oson va undan foydalanish qulay bo'lishligi uchun dasturda ,,matnni ajratish“ tushunchasidan foydalaniladi (bo'sh joy, qatorni tugashi va izohlar).

Bo'sh joy (probel) grafik tasvirga ega emas belgi bo'lib, qatordagi bo'sh joyni anglatadi. Lekin, bo'sh joy belgisi o'zining sonli kodiga ega va dastur matnidagi boshqa belgilar kabi kompyuterga kiritiladi.

Qator oxiri (tugashi) boshqaruvchi belgi bo'lib, u ham grafik tasvirga ega emas. Ma’lumki, dastur matnini yozish davomida uni tabiiy ravishda yangi qatorlarga ajratilib yoziladi. Chunki, shu matnni yozmoqchi bo'lgan qog'ozning ham, kompyuter ekranining ham o'lchamlari cheklangan. Dastur matnini alohida qatorlarga ajratmay yozish ham mumkin, lekin bir satrga 256 tadan ortiq belgi sig'maydi. Dastur matnini alohida qatorlarga ajratish, dastur tuzuvchining xohishiga qarab bajariladi. Ma’lum bir qator tugamay turib, yangi qatorga o'tish uchun ,,qator oxiri“ tugmachasi bosiladi. Bu tugmacha ham o'zining maxsus sonli kodiga ega.

Izohlar dasturni o'qishga oson bo'lishi, uni qiynalmay tekshirib, yo'l qo'yilgan xatolarni to'g'irlash va dasturda bajarilayotgan ishlarni tushuntirib borish uchun qo'yiladi. Izohsiz yozilgan dasturni hujjat sifatida qabul qilinmaydi.

Muvaffaqiyatli qo'yilgan izoh dasturning va dastur tuzuvchining katta yutug'i hisoblanadi. Izohlar ixtiyoriy vaqtda dastur matniga kiritilishi yoki olib tashlanishi mumkin. Bu bilan dasturning ishi o'zgarib qolmaydi. Izohlarni "{" va "}" qavslari ichiga olinib yoziladi.

Dastur ,,matn ajratgich“ laridan foydalanishning quyidagi qoidalariga amal qilish lozim:

- tilning ketma-ket yozilgan ikkita konstruksiyasi orasiga albatta bo'sh joy yozilishi kerak;
- ajratgichlarni xizmatchi so'zlar, sonlar va ismlar orasiga qo'yish maqsadga muvofiq emas.

Quyida yuqoridagi qoidalar asosida yozilgan dasturga doir misol keltirilgan.

Misol. Quyidagi berilgan funksiyalarning qiymatlarini [a,b] oralig'idagi $x=a=ih$, $h=\frac{b-a}{n}$ lar uchun (n-berilgan son) hisoblash dasturini tuzing: $f_1(x)=x^2$,$f_2(x)=3-x$, $f_3(x)=0,5-\sin x$

```
Program P1;
 { f1(x)=x*x; f2(x)=3-x; f3(x)=0.5-sin(x) funksiyalar qiymatini [a,b]
 oralig‘ida hisoblash dasturi }
  const
     n=10; {[a,b] oraliqni 10 ta bo‘lakchalarga ajratdik}
  Var
     a,b:real;
     i:integer;
     x,h,y1,y2,y3:real;
   Begin
      read(a,b);                {[a,b] oraliqni chegaralarini ajratish}
      h:=(b-a)g‘n;  x:=a;  i:=0;  {Boshlang‘ich ma’lumotlar hisoblandi}
      Repeat
          y1:=x*x;
          y2:=3-x;
          y3:=0.5-sin(x);
Writeln (x,y1,y2,y3); {Funksiyalar hisoblanib, natijalar chop etilmoqda}
           x:=x=h;  i:=i=1;
       Until i=n=1
```

{Hisob ishlari yakunlandi}

end.

1.2. Paskal tilining dasturlash texnologiyasi

Odatda, dasturda ishlatiluvchi ma'lumotlar quyidagi tiplarning birortasiga tegishli bo'ladi: butun qiymatli tiplar, haqiqiy qiymatli tiplar, belgili va satrli tiplar, mantiqiy qiymatli va ko'rsatkichli tiplar. Umuman olganda, tiplarni ikkita guruhga ajratish mumkin: asosiy (yoki oddiy) va hosilaviy. Yuqorida sanab o'tilgan tiplar asosiy guruhga tegishli bo'lgan tiplardir. Hosilaviy tiplar esa, asosiy yoki hosilaviy guruhga tegishli tiplardan hosil qilinadi.[10]

Butun qiymatli tipga tegishli songa misollar: -1501, 0, 9999.

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

Haqiqiy qiymatli tipga tegishli sonlarga misollar:

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

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

Hamma harflar, belgi va raqamlar, masalan A, b, '', !, $, S belgili tipga tegishlidir. Belgili tipni qabul qiluvchi o'zgaruvchilarni e'lon qilish uchun ***Char*** xizmatchi so'zidan foydalanish mumkin.

Belgilarning ixtiyoriy yig'ilmasi (ketma-ketligi) qatorlar deb ataladi. Misol, 'Rahmatjon', '$25', '_START'. qator hatto bo'sh ham bo'lishi mumkin (' '). Bu tipdagi o'zgaruvchilarni e'lon qilish uchun String xizmatchi so'zidan foydalaniladi.

Mantiqiy o'zgaruvchilar faqat *True* (rost) va *False* (yollg'on) qiymatlarining bittasinigina qabul qilishi mumkin. Bu tip o'zgaruvchilarini e'lon qilish uchun *Boolean* xizmatchi so'zi ishlatiladi.

Ko‘rsatgichlar ma’lumotlarning kompyuter xotirasidagi turar joyi (adresi)ni aniqlab beradi va ularni e’lon qilish uchun *Pointer* xizmatchi so‘zidan foydalaniladi.

Hosilaviy tiplarni hosil qilish va ularni e’lon qilish yo‘llarini kelgusi bo‘limlarda to‘liq tushuntirib o‘tiladi.

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

Butun qiymatli tiplarning barchasi quyidagi jadvalda keltirilgan:

Tip ko‘rinishi	Mazkur tipli o‘zgaruvchining qabul qiladigan qiymatlar oralig‘i	O‘zgaruvchining kompyuter xotirasidan egallaydigan joyi
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

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

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

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

{Dastur sarlavhasini yozamiz}

Program Sonlar;

```
Var  {dasturda foydalanish mumkin bo‘lgan barcha o‘zgaruvchilar shu var
so‘zidan so‘ng e’lon qilinadi}
m,n:integer;     {m va n o‘zgaruvchilar o‘rtacha kattalikdagi butun sonlar}
k1,k2,k3:integer;{k1=m+n, k2=m-n, k3=m*n – bajarilgan arifmetik amallar natijasini
        xotirada saqlash uchun tanlangan butun tipli o‘zgaruvchilar}
begin
{Paskal dasturi begin (boshlanmoq) so‘zi bilan boshlanib,
end.(tamom) so‘zi bilan tamomlanadi}
        Readln(m,n);
{m va n butun sonlarini kiritish so‘ralyapti, agar kiritilayotgan son butun bo‘lmasa
“Error 106:Invalid numeric format.” xatosi xabar qilinadi va dastur ishini
to‘xtatadi}
        k1:=m+n;
        k2:=m-n;
    k3:=m*n;                {so‘ralgan amallar bajarildi}
writeln (k1,k2,k3);  {hisoblangan k1=m+n, k2=m-n, k3=m*n natijaviy qiymatlarni
chop etish tashkil etildi}
end.                              {Dastur tamom bo‘ldi}
```

Bundan tashqari, Turbo-Paskalda o‘n oltilik sanoq sistemasida yozilgan butun sonlardan foydalanishga ham ruxsat beriladi. O‘n oltilik sanoq sistemasidagi butun sonni aniqlashda uning oldiga “$” (dollar) belgisi qo‘yiladi.

Misol, $11 o‘nli sanoq sistemasidagi 17 ga, $12 soni esa 18 ga teng.

Endi shu holatga doir quyidagi sodda dasturni keltiramiz:

```
Program Sanoq_sistema;
Var
N:integer;
Begin
N:=12; {N butun qiymatli o‘zgaruvchiga o‘nlik sanoq sistemasidagi 12 soni
o‘zlashtirilyapti}
```

N:=$12; {N butun qiymatli o‘zgaruvchiga o‘n oltilik sanoq sistemasidagi 12 soni o‘zlashtirilyapti. Bu son amaldagi o‘nli sanoq sistemasida 18 ga teng}
End.

Haqiqiy sonlar matematika kursidan ma’lum bo‘lgan oddiy o‘nlik kasr sonlardir. Agar kompyuteringiz matematik soprosessorli bo‘lsa quyidagi jadvalda sanab o‘tilgan barcha tiplar o‘rinli bo‘ladi (2-jadval):

Agar kompyuterda matematik soprosessor bo‘lmasa, faqat *Real* tipinigina ishlatish mumkin. Haqiqiy sonlarga doir quyidagi misolni ko‘raylik:2-jadval.

Tip ko‘rinishi	Mazkur tipli o‘zgaruvchining qabul qiladigan qiymat oralig‘i	O‘zgaruvchining kompyuter xotirasidan egallaydigan joyi
Real	2.9E-39..1.7E38	6 bayt
Single	1.5E-45..3.4E38	4 bayt
Double	5.0E-324..1.7E308	8 bayt
Extended	3.4E-4932..1.1E4932	10 bayt
Comp	-9.2E18..9.2E18	8 bayt

Berilgan *m* va *n* haqiqiy sonlari ustida to‘rt matematik amalni bajarish dastursini tuzing.

```
Program arifmetika;
    Var
  m,n:real; {m va n o‘zgaruvchilarni haqiqiy sonlar tipiga tegishli deb e’lon qildik}
    Begin
     Readln (m); {m sonini kiritish so‘ralyapti}
     Readln (n); {n sonini kiritish so‘ralyapti}
     Writeln (‘sonlar yig‘indisi=’,m+n);
     Writeln (‘sonlar ayirmasi=’,m-n);
     Writeln (‘sonlar ko‘paytmasi=’,m*n);
```

If n<>0 then Writeln (‘sonlar bo‘linmasi=’,m/n);

{Agar n soni nolga *teng bo‘lmasa m/n amaliyning natijasi mos izoh bilan chop etiladi}*

end.

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

Misol: *‘A‘*, yoki #60.

Qator – bu ′(apostrof) belgisi ichiga olib yozilgan belgilarning oddiy ketma-ketligidir: 'Ab21#9!cd', 'Sultonov Rahmatjon'.

Qator bo‘sh yoki bitta belgili bo‘lishi ham mumkin. qatorli o‘zgaruvchi uzunligi 255 gacha bo‘lgan belgili qiymatlarni qabul qilishi mumkin. Umuman olganda, har bir qatorli o‘zgaruvchiga xotiradan 256 bayt joy ajratiladi. Xotirani tejash uchun, qatorning tipini quyidagicha ko‘rsatish maqsadga muvofiqdir: *String*[N], N - qatordagi belgilar soni. Bu holda belgili o‘zgaruvchi uchun N bayt joy ajratiladi.

Belgilar va qatorlar ustida bir qancha amallar bajarish mumkin, ya’ni qatordan kerakli bo‘lakni kesib olish, qatorlarni bir-biriga qo‘shish va natijada yangi qatorlar hosil qilish, qatorlar haqidagi to‘liq ma’lumotni kerakli bo‘limdan olish mumkin. Belgilar va qatorlarga doir quyidagi sodda dasturni keltiramiz:

Program String;

Var

ch: char; {ch o‘zgaruvchi belgili qiymat qabul qiladi}

qator1,qator2:String; {qator1 va qator2 o‘zgaruvchilar uzunligi 255 dan ortmagan qatorlarni o‘zlashtirishi mumkin}

N:String[5]; {N o‘zgaruvchisi 5 ta belgidan tashkil topgan qatorlarni o‘zlashtiradi}

Begin

ch:='A'; {ch o'zgaruvchisi A belgini o'zlashtirdi}

N:='Ascar'; {N o'zgaruvchisi 5 ta harfli Ascar so'zini o'zlashtirdi}

qator1:=ch+'li '+N; {qator1 o'zgaruvchisi natijaviy Ali Ascar so'zini o'zlashtirdi}

qator2:=''; {qator2 o'zgaruvchisi bo'sh qatorni ifodalayapti lekin, bu o'zgaruvchi uchun xotiradan 256 bayt joy ajratilgan}

end.

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

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

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

Mantiqiy ko'paytirish	Mantiqiy qo'shish	Mantiqiy rad etmoq
True and true = true	true or true = true	not true = false
True and false= false	true or false=true	not false= true
False and true = false	False or true = true	
False and false = false	False or false = false	

Ixtiyoriy, qiymatlarni solishtirish amali ham mantiqiy qiymatni beradi: Misol:

3>2 natijasi ***true***

0<-1 natijasi ***false***.

Paskal tilida tilning standart tiplaridan yoki oldin hosil qilingan yangi tiplardan foydalanib yana yangi tiplar yaratish mumkin. Dasturda yangi tiplarni kiritish uchun maxsus tip aniqlash bo'limi mavjud. Bu bo'lim ***type*** xizmatchi so'zidan keyin boshlanadi.

Har bir yangi tipni e’lon qilishdan oldin uning nomi (tipning identifikatori), so‘ng esa tipni nimadan tashkil topganligi ko‘rsatiladi.

Yangi tip yozuv ham bo‘lishi mumkin, uning maydoni esa standart tipdan yoki oldingi kiritilgan tiplardan tashkil topishi mumkin. O‘z o‘rnida kiritilgan yangi tip dasturni yozishda juda qo‘l keladi va dasturning sifatini keskin oshiradi.

Paskal - dasturi – dastur sarlavhasi va nuqta bilan tugovchi dastur tanasidan tashkil topgan. Dastur sarlavhasi va dastur tanasini; (nuqta vergul) belgisi bilan ajratiladi:

<Paskal dastursi>::=< sarlavhasi>;< tanasi>.

Dastur sarlavhasi ***program*** xizmatchi so‘zidan boshlanadi va undan so‘ng dasturga foydalanuvchi bergan nom yoziladi:

< sarlavhasi>::q ***program*** < ismi>;

Dasturning asosiy qismi uning tanasi hisoblanadi. Dasturning tanasini qisqacha qilib blok ham deb atash mumkin. Umuman olganda, blok qat’iy ketma-ketlikda yoziluvchi oltita bo‘limdan tashkil topgan:

<blok>::= <metkalar bo‘limi>
<o‘zgarmaslar bo‘limi>
<yangi tiplar bo‘limi>
<o‘zgaruvchilar bo‘limi>
<prosedura (qism dastur) va funksiyalar bo‘limi>
<operatorlar bo‘limi>

Dastur tanasining asosiy qismi bu operatorlar bo‘limidir. Har qanday dasturda bu bo‘lim albatta bo‘lishi kerak. Dasturga qo‘yilgan masalani yechish shu bo‘limda amalga oshiriladi. Boshqa bo‘limlar esa yordamchi bo‘limlar bo‘lib, tiplarni e’lon qilish bo‘limlari deb ataladi. Bu yordamchi bo‘limlar dasturda qatnashishi yoki qatnashmasligi ham mumkin, lekin ularning yozilish ketma-ketligi saqlanib qolinishi zarur.

Paskal ─ dasturning umumiy ko‘rinishini quyidagi ko‘rinishda yozib olaylikda, so‘ng har bir bo‘limni to‘laroq tahlil qilib chiqamiz:

Program < ismi>;

label

<metkalar ro‘yxati>;

const

<o‘zgarmaslar va ularning qiymatlari>;

type

<ma’lumotlarning yangi, nostandart tiplarini aniqlash>;

var

<o‘zgaruvchilarni, proseduralar va

funksiyalarni e’lon qilish>;

begin

<operatorlar bo‘limi>

end.

Dasturning ixtiyoriy operatorini boshqa operatorlar orasida deb ataladi. Metkalar, dasturda o‘tish operatoridan foydalangandagina ishlatiladi. Metka sifatida oddiy identifikatorlardan yoki sonlardan foydalanilsa bo‘laveradi. Dasturda ishlatilgan barcha metkalar ***label*** xizmatchi so‘zidan keyin boshlanuvchi metkalar bo‘limida e’lon qilinib qo‘yilishi kerak:

<metka bo‘limi>::q<bo‘sh> | ***label*** <metkalar ro‘yxati>;

Metkalarning nomlari original, ya’ni o‘xshashi yo‘q bo‘lishi kerak.

Misol: ***label*** L1, L2, A3;

Bu yerda L1, L2, A3 lar dasturda ishlatiluvchi metkalarning nomlari.

```
Label  m10, m20, StopLabel, 1;
Var
      I:ShortInt;
Begin
      1:
         If  i<10 Then Goto m10 Else Goto m20;
  m10:  Writeln('i  kichik 10 dan');
         Goto StopLabel;
  M20:   i:qi-1;
```

***Goto** 1;*

StopLabel:

End.

O‘zgarmas ─ bu dasturni ishlashi davomida o‘zgarmay qoladigan miqdordir. Agar miqdor dasturda ko‘p marta ishlatilsa, uni dastur matnida qayta - qayta yozgandan ko‘ra, bu miqdorni o‘zgarmas deb aniqlab olib, dasturdagi miqdorni o‘rniga o‘zgarmasni ismini yozish qulay bo‘ladi. Masalan, hammaga ma’lum π (π=3,1415926535…) soni. Bu sonni bir necha marta takroran dasturda yozish noqulay, shuning uchun, uni o‘zgarmas sifatida aniqlab olish maqsadga muvofiqdir.

O‘zgarmas ***const*** xizmatchi so‘zidan keyin e’lon qilinadi (aniqlanadi):

<o‘zgarmasni aniqlash>::=<o‘zgarmas ismi>=<o‘zgarmas>;

bu yerda <o‘zgarmas>::= <skalyar miqdor>|<belgili qator>|<o‘zgarmas ismi>|=<o‘zgarmas ismi>|-<o‘zgarmas ismi>;

<o‘zgarmaslar bo‘limi>::=<bo‘sh>| ***const*** <o‘zgarmasni aniqlash>;

Misol: *Program L1;*

***const** Pi=3.1415926535;*

var A,B: real;

Begin

A: =Pi=Sin(Pi*(Pi-1));

B:qsqrt(a);

end.

Paskal tilida qiymatlarning to‘rtta standart tiplari mavjud: *integer* (butun), *real* (haqiqiy), *Char* (belgili) va *boolean* (mantiqiy). Bu tiplar bilan bir qatorda, dasturning avtori o‘zi uchun zarur bo‘ladigan tiplarni aniqlab olib, ulardan foydalanishi mumkin. Buning uchun, muomalaga kiritilayotgan har bir yangi tipga o‘ziga xos ism berish kifoya va o‘zgaruvchilarni e’lon qilish bo‘limida bu tipdan bemalol foydalanish mumkin bo‘ladi.

Tip e’lon qilish quyidagi metaformula asosida amalga oshiriladi:

<tip e’loni>::=<tip ismi>=<tip>;

<tip>::=<tip ismi>|<tip vazifasi>;

Tiplarni aniqlash bo‘limi esa quyidagicha aniqlanadi:

<tip aniqlash bo‘limi>::=<bo‘sh> | ***type*** <tip e’loni>

Misol:

type

Mantiqqboolean;

b=integer;

Hafta= (dush, sesh, chor, pay, ju, shan, yaksh);

Ish_Kuni=dush..ju;

Har qanday dasturda o‘zgaruvchilar deb ataluvchi dastur obyektlaridan foydalaniladi. O‘zgaruvchi ─ qiymat qabul qiluvchi obyektdir. Ularga qiymat dasturni bajarilishi davomida beriladi. Har bir o‘zgaruvchiga uni qabul qiladigan qiymati va vazifasiga qarab ismlar beriladi. Shunday qilib, o‘zgaruvchi o‘zining nomi va qabul qiladigan qiymati bilan xarakterlanadi.

Dasturda ishlatiluvchi har bir o‘zgaruvchi o‘zi qabul qiladigan qiymatlarining tiplariga mos holda o‘zgaruvchilar bo‘limida e’lon qilinib qo‘yilishi kerak:

<o‘zgaruvchilar bo‘limi>::=<bo‘sh> | ***Var*** <o‘zgaruvchilar e’loni>;

<o‘zgaruvchilar e’loni>::=<o’zgaruvchi ismi >:<tip>

Misol:

Var

n,m,k : integer;

d : real;

x,y : real;

S : boolean;

Dasturda ishlatilgan o‘zgaruvchilar faqat bir martagina e’lon qilinishi kerak.

Boshqa tillardagi kabi, Paskal tilida ham dasturni yozuvchi o‘zi uchun qulay va zarur bo‘lgan prosedura va funksiyalarni muomalaga kiritishi mumkin. Tabiiyki, bu prosedura va funksiyalar ham xuddi o‘zgaruvchilar kabi e’lon qilinib qo‘yilishi

kerak. Hamma ishlatiluvchi prosedura va funksiyalarga ism berilib, shu ismi bilan ular chaqirilib dasturning zarur joyida ishlatiladi. Ularga murojaat xuddi oddiy standart funksiyalarga murojaat kabi amalga oshiriladi.

Prosedura va funksiyalar bo‘limi o‘zgaruvchilar bo‘limining davomi bo‘lib ***procedure*** eki ***function*** xizmatchi so‘zlari bilan boshlanib, ixtiyoriy ketma-ketlikda e’lon qilinadi.

Bu bo‘lim dasturning eng asosiy bo‘limi bo‘lib, dastur bo‘yicha hisoblanuvchi barcha ishlar shu bo‘limda bajariladi. Bo‘limning metaformulasi quyidagicha yoziladi:

<operatorlar bo‘limi>::=

begin

<operatorlar ro‘yxati>

end.

Dastur o‘z ishini shu bo‘limda yozilgan operatorlar ro‘yxati bo‘yicha, qat’iy ketma-ketlikda bajaradi.

Butun yoki haqiqiy tipli, sonli natija beruvchi ifodani (odatda bunday ifodani arifmetik ifoda deb ataladi) hisoblash uchun arifmetik o‘zlashtirish operatoridan foydalaniladi. Arifmetik ifodada qatnashuvchi barcha o‘zgaruvchilar haqiqiy yoki butun tipli bo‘lishi kerak. Arifmetik ifoda - sonlar, o‘zgarmaslar, o‘zgaruvchilar va funksiyalardan tashkil topadi, hamda =, -, *, div, mod kabi amallar yordamida yoziladi. Arifmetik amallarni bajarilishi quyidagi tartibda bo‘ladi: *, -, div, mod, =,

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

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

7 div 2 = 3	5 div 3 = 1	-7 div 2 = -3
-7 div –2 = 3	2 div 5 = 0	3 div 4 = 0

Mod – butun sonlar bo‘linmasining qoldig‘ini aniqlaydi. m mod n qiymat faqat n>0 dagina aniqlangan. Agar m≥0 bo‘lsa m mod n=m-((m div n)*n), m<0 bo‘lsa m mod n=m-((m div n)*n)=n, m mod n ning natijasi doim musbat sondir.

Misol:

7 mod 2 = 1 3 mod 5 = 3 (-14) mod 3 = 1 (-10) mod 5 = 0

Paskal tilida ham boshqa algoritmik tillar kabi arifmetik standart funksiyalari mavjud. Bu funksiyalarni matematik yozilishi va Paskal tilida ifodalanishi quyidagi jadvalda keltirilgan:

Funksiyalar	Paskal tilida ifodalanishi
sin x	sin(x)
Cosx	cos(x)
Arctg x	arctan(x)
Lnx	ln(x)
Exp(x)	exp(x)
$\sqrt{x}$	Sqrt(x)
|x|	Abs(x)
x2	sqr(x)

Arifmetik ifodaga doir misollar :

2*5 - 4*3, 9 div 4/2, 45/5/3, a = b/2*7.2 - sqrt(7),

exp(2 - a)*9.7 - 6.1*6.1

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

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

lny=lnan, logarifm xossasiga ko‘ra

lny=nlna, bu tenglikdan "u" ni aniqlaymiz,

u=enlna - bu tenglikni Paskal tilida quyidagicha yozish mumkin: y=exp(n*ln(a)).

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

Matematik yozuvi	Paskal tilidagi yozuvi
$\frac{a \cdot (a+b)}{bc}$	a*(a+b)/(b*c)
$\frac{1}{1-\frac{1}{1-\frac{1}{x}}}$	1/(1-1/(1-1/x))
2-(x-b)2-eax=sinCX	2-sqr(x-b)-exp(a*x)=sin(c*x)
$x \cdot \frac{e^{x^2+y^2}-1}{\sqrt{\lvert x^2+y^2 \rvert}}$	x*(exp(x*x=y*y)-1)/sqrt(abs(x*x=y*y))

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

x:= 0;

c:= sqrt(a*a=b*b);

y:= 2*pi*r; i:= i=1; i:= 5/4; x:= a - bg‘2;

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

To‘g‘ri tuzilgan dastur

```
Program Misol;
Var
   a,x,y:Real;
Begin
```

Noto‘g‘ri tuzilgan dastur

```
Program Misol;
Var
   a,x,y:Real;
Begin
```

```
a: =2.3;
x: =3.1;
y: =a*x;
Writeln('y=',y);
End.
```

```
a: =2.3;
y: =a*x;
{o'zlashtirish operatorining o'ng
tomonidagi "X" o'zgaruvchining
qiymati aniqlanmagan}
Writeln('y=',y);
End.
```

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

Mantiqiy ifoda ─ arifmetik ifoda, solishtirish belgilari va mantiqiy amallardan tashkil topadi. Mantiqiy ifodaning natijaviy qiymati true (rost) ¸ki false (elg'on) bo'ladi.

Mantiqiy ifodada amallarning bajarilish tartibi quyidagicha:

Not

*, /, div, mod, and

=,-,or

=, <, >, <=, >=,<>

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

Mantiqiy ifodaga doir misollar:

x<2*y

true

not not d

(x>y/2)

d and (x=y) and b

(C or D) and (xqy) or not B

Mantiqiy o'zlashtirish operatoriga doir misollar:

```
d:=true;
b:= (x>y) and (kq0);
c:=D or B and true;

VAR Global Flag:Boolean;
FUNCTION GETSQR( x:real );
Const  SQRMAXq100;
Begin
        X:=x*x;
        GlobalFlag:=( x>SQRMAX );
        If  GlobalFlag  then  x:=SQRMAX;
        GetSQR:qx;
End;
```

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

Shartli operatorning sintaksis qoidasiga ko‘ra then va else xizmatchi so‘zlaridan so‘ng faqat bitta operator yozilishi mumkin, agar bir nechta operatorlarni yozish lozim bo‘lsa u holda, bu operatorlar ketma-ketligi begin va end xizmatchi so‘zlari orasiga olinib tashkiliy operator hosil qilinadi.

Ko‘pgina operatorlar kabi, shartli operator ham rekursivlik xossasiga ega ya’ni, shartli operator ichida yana shartli operator qatnashishi mumkin. Lekin, chala shartli operatorning ichida yana shartli operator yozishda ehtiyot bo‘lmoq zarur, chunki yozilgan operatorni ikki xil ma’noda tushunish mumkin:

```
        if B1 then if B2 then S1 else S2
```

bu operator quyidagicha tushunilishi mumkin:

1) `if B1 then begin if B2 then S1 else S2 end`

`if B1 then begin if B2 then S1 end else S2`

II BOB. TURBO PASCAL TILIDA SONLAR NAZARIYASINING ASOSIY ALGORITMLARINI DASTURLASH TEXNOLOGIYASI

2.1. Sonlar nazariyasining asosiy tushunchalari va algoritmlari

Tub son va uning xossalari.

Har qanday natural a sonning kamida 2 ta bo‘luvchisi bor. 1 soni va a sonining o‘zi.

Ta’rif. Faqat ikkita bo‘luvchisi bor natural son *tub son* deyiladi.

Masalan, 3, 5, 17 sonlari tub son, chunki ularning 1 va o‘zidan boshqa bo‘luvchilari yo‘q. 14 soni tub emas, ung 1 va 14 dan boshqa bo‘luvchilari ham bor , ular 2, 7 sonlari.

Tub sonlar quyidagi xossalarga ega:

1º. Agar p tub soni 1 dan farqli birorta n songa bo‘linsa, $p = n$ bo‘ladi.

Isbot: haqiqatdan ham $p \neq n$ bo‘lsa, p sonning 3 ta turli bo‘luvchisi bor bo‘ladi, bular:$1, p, n$. Bu esa shartga zid, demak p tub son bo‘la olmaydi.

2º. Agar p va q turli tub sonlar bo‘lsa, p tub son q tub songa bo‘linmaydi.

Isbot: p tub son bo‘lgani uchun u faqat 1 ga va p ga bo‘linadi. q ≠ p q ≠ 1 (q –tub son, 1 tub son emas) bo‘lgani uchun p q ga bo‘linmaydi.

3º. Agar a va b natural sonlar ko‘paytmasi p tub songa bo‘lishsa, bu sonlardan biri p ga bo‘linadi.

Isbot: Faraz qilaylik $a \vdots p$, u holda p tub son bo‘lgani uchun ularning 1 dan boshqa umumiy bo‘luvchisi yo‘q $a \cdot b \vdots p \Rightarrow b \vdots p$.

4º. 1 dan katta istalgan natural sonning hech bo‘lmaganda bitta tub bo‘luvchi bor.

5º. a murakkab sonning eng kichik tub bo‘luvchisi $\sqrt{a}$ dan katta emas

Tub sonlar to‘plamining cheksizligi.

Tub sonlar to‘plamining cheksiz ekanligi eramizdan avvalgi III asrda Aleksandriyada yashagan grek matematigi Evklid tomonidan isbot qilingan.

Evklid teoremasi. Tub sonlar to‘plami cheksizdir.

◯ Isbot. Tub sonlar to‘plami chekli deb faraz qilaylik.

U holda $p = \{p_1, p_2, \dots, p_n\}$ tub sonlar to‘plamiga ega bo‘lamiz.

$a = p_1 \cdot p_2 \cdot \dots \cdot p_n + 1$ sonni hosil qilaylik. a soni tub emas, chunki u $p_1 \cdot p_2 \cdot \dots \cdot p_n$ tub sonlarning hammasidan katta va barcha tub sonlar to‘plami p ga kirmaydi. a soni murakkab ham bo‘la olmaydi, chunki 4° ko‘ra barcha murakkab sonlarning kamida bitta tub bo‘luvchisi bo‘lishi kerak. Lekin a son bu tub sonlarning birortasiga ham bo‘linmaydi. Demak, p to‘plamga kirmaydigan bitta bo‘lsa ham tub son bor ekan. Bu qarama qarshilik farazimiz noto‘g‘riligini ko‘rsatadi. Demak, tub sonlar to‘plami cheksiz ekan.

Eratosfen g‘alviri

Eratosfen elagi (Eratosfen g‘alviri) —butun son n gacha bo‘lgan barcha tub sonlarni topish algoritmi bo‘lib, qadimiy Grek matematigi Eratosfen Kireniyga bag‘ishlab nomlangan. Eratosfen elagi algoritmi kichik (odatda, 10 milliondan kichik bo‘lgan) tub sonlar topishning eng tez uslub hisoblanadi. Butun son n gacha bo‘lgan barcha tub sonlarni topish Eratosfen uslubiga asosan quyidagi bosqichlardan iborat:

1. Ikkidan boshlab n gacha bo‘lgan barcha sonlarni yozib chiqamiz (2, 3, 4, … ,n).
2. O‘zgaruvchi p boshida 2 — birinchi butun songa teng deb qabul qilamiz.
3. Yozib chiqilgan sonlardan p dan boshlab p qadam bilan n gacha barcha sonlarni o‘chiramiz, (ya’ni $2p, 3p, 4p,$ ... kabi sonlar).
4. p da katta birinchi o‘chirilmagan sonni p deb yangidan qabul qilamiz.
5. 3- va 4-qadamni p^2 qiymati n dan katta yoki teng bo‘lguncha takrorlaymiz.

Natijada ro‘yxatdagi o‘chirilmagan sonlarning barchasi tub son bo‘ladi.

Amaliyotda, ushbu algoritmni quyidagicha yaxshilash (tezlashtirish) mumkin. Algoritmdagi 3-qadamda sonlarni p^2 dan boshlab o‘chirish yetarli, chunki bundan kichik sonlar avval o‘chirilgan bo‘ladi. Va, algoritm p^2 qiymati n ga teng yoki katta bo‘lganda to‘xtatiladi.

Masalan n = 40 bo‘lsin. Qo‘yidagi qatorga ega bo‘lamiz.

1 2 3 ~~4~~ ~~5~~ ~~6~~ 7 ~~8~~ ~~9~~

~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19

~~20~~ ~~21~~ ~~22~~ 23 ~~24~~ ~~25~~ ~~26~~ ~~27~~ ~~28~~ 29

~~30~~ 31 ~~32~~ ~~33~~ ~~34~~ ~~35~~ ~~36~~ 37 ~~38~~ ~~39~~ ~~40~~

1 dan 40 gacha bo‘lgan tub sonlar qo‘yidagilardan iborat.

2,3,5,7,11,13,17,19,23,29,31,37

Evklid algoritmi.

Evklid algoritmi — ikkita butun sonning eng katta umumiy bo‘luvchisi(EKUB)ni topish, shuningdek ikkita o‘lchovdosh kesmaning umumiy o‘lchovini topish usuli. Ikkita musbat butun sonning eng katta umumiy bo‘luvchisi(EKUB)ni topish uchun avvalo katta sonni kichik songa bo‘lish, so‘ngra kichik sonni katta sonning qoldig‘iga, keyin esa birinchi qoldiqni ikkinchi qoldiqqa va hokazo bo‘lish lozim. Bu jarayondagi noldan farqli oxirgi qoldiq berilgan sonlarning eng katta umumiy bo‘luvchisi bo‘ladi.

$$EKUB(a,b) = \begin{cases} EKUB(a-b,b), & agar\ a > b\ bo'lsa; \\ a, & agar\ a = b\ \ bo'\ lsa; \\ EKUB(a,b-a), & agar\ b > a\ bo'\ lsa; \end{cases}$$

Masalan:

EKUB(4565,960)=EKUB(3605,960)=EKUB(2645,960)=EKUB(1685,960)=
=EKUB (725,960) =EKUB (725, 235) =EKUB (490,235) = EKUB (255,235) =
=EKUB (20,235) = EKUB (20,215) = EKUB (20,195) = EKUB (20,175) =
=EKUB(20,155) =EKUB(20,135)=EKUB(20,115)=EKUB(20,95)=EKUB(20,75)=
=EKUB(20,55)=EKUB(20,35)=EKUB(20,15)=EKUB(5,15)=EKUB(5,10)=
=EKUB (5, 5)= 5

Sonlarning kanonik yoyilmasini topish ularni tub ko‘paytuvchilarga ajratish bilan bog‘liq edi. Ko‘p xonali sonlarning tub ko‘paytiruvchilarini topish ba’zi hollarda qiyinlik qiladi.

Masalan, 8897 sonini tub ko‘paytuvchilarga ajratishda avval 7 ga so‘ng 1271 sonini 2,3,5,7,11,13,17,19,23,29,31 sonlariga bo‘lib ko‘ribgina, 31 tub bo‘luvchini topamiz. Shunday hollarda EKUBni tezkor topish imkonini beruvchi boshqa usullardan foydalanish mumkin. Bu usul Evklid algoritmi deyiladi va u quyidagi mulohazalarga asoslanadi.

Endi berilgan sonlarning eng katta bo‘luvchisini qanday topish mumkinligi haqida fikr yuritamiz.

$a, b \epsilon Z, a \neq 0, b \neq 0$ sonlar uchun: $c_1, c_2, c_3, \dots, c_{n-1}, c_n$ ketma-ketlikni quyidagi tuzamiz: $c_1 = a, c_2 = b$ deb olamiz. Agar bo ‘lsa,$(a, b) = b$ bo‘ladi. Aks holda c_1 ni $c_2$ga bo‘lishda hosil bo‘lgan qoldiqni $c_3$deb olamiz, c_2 ni c_3 ga bo‘lishdagi qoldiqni $c_4$deb olamiz va hokazo.

Bunday qoldiqli bo‘lishlar ketma-ketligini qulaylik uchun tengliklar zanjiri ko‘rinishida bunday yozamiz:

$$c_1 = c_2 \cdot q_2 + c_3$$
$$c_2 = c_3 \cdot q_3 + c_4$$
$$c_3 = c_4 \cdot q_4 + c_5$$
$$\dots \quad \dots \quad \dots \quad \dots$$
$$c_{n-2} = c_{n-1} \cdot q_{n-1} + c_n$$
$$c_{n-1} = c_n \cdot q_n + 0$$

qoldiqli bo‘lish natijasida qoldiqlar uchun $c_2 > c_3 > c_4 > \cdots$ munosabat o‘rinli bo‘ladi, bundan tashqari qoldiqlar nomanfiy sonlardir. Shu sababli bu qoldiqlar orasida nolga teng bo‘ladigani albatta uchraydi. cn uchun noldan farqli oxirgi qoldiqni olamiz. Bu holda c_{n-1} soni c_n ga bo‘linishi ravshan.

Hosil qilingan $c_1, c_2, c_3, \dots, c_{n-1}, c_n$ ketma-ketlik a va b sonlari uchun Evklid ketma-ketligi deyiladi.Ketma-ket qoldiqli bo‘lish yordamida bunday ketma-ketlikni hosil qilish usuli Evklid algoritmi deb ataladi.

Misol. 4565 va 960 sonlari uchun Evklid ketma-ketligini tuzing.

Yechilishi:

$$4565 = 960 \cdot 4 + 725$$

$$960 = 725 \cdot 1 + 135$$

$$725 = 135 \cdot 5 + 50$$

$$135 = 50 \cdot 2 + 35$$

$$50 = 35 \cdot 1 + 15$$

$$35 = 15 \cdot 2 + 5$$

$$15 = 5 \cdot 3 + 0$$

Bundan, izlangan ketma-ketlik hosil bo‘ladi: 4565, 960, 725, 135, 50, 35, 15, 5.

Teorema. $a\ va\ b (a \neq 0, b \neq 0)$ sonlar uchun tuzilgan Evklid ketma-ketligi, $c_1, c_2, c_3, \dots, c_{n-1}, c_n$ ning oxirgi hadi $a\ va\ b$ sonlarining eng katta umumiy bo‘luvchisi(EKUB) bo‘ladi: $c_n = (a, b)$.

Isbot: $c_1, c_2, c_3, \dots, c_{n-1}, c_n$ sonlar o‘zaro yuqorida keltirilgan qoldiqli bo‘lish munosabati bilan bog‘langan. Tengliklar zanjiri orqali yuqoriga harakatlanib, quyidagilarni aniqlaymiz:

$c_{n-1} = c_n \cdot q_n$ munosabatdan ;

$c_{n-2} = c_{n-1} \cdot q_{n-1} + c_n$ munosabatdan ;

$c_{n-3} = c_{n-2} \cdot q_{n-2} + c_{n-1}$ munosabatdan ;

va hokazo

Bunda. $c_1 = a\ va\ c_2 = b$ Shunday qilib, c_n —$a\ va\ b$ sonlarining umumiy bo‘luvchisi.

x soni $a\ va\ b$ sonlarining boshqa bir umumiy bo‘luvchisi bo‘lsin. Evklid algoritmidagi qoldiqli bo‘lishlar zanjirida pastga qarab harakatlanib, quyidagilarni aniqlaymiz:

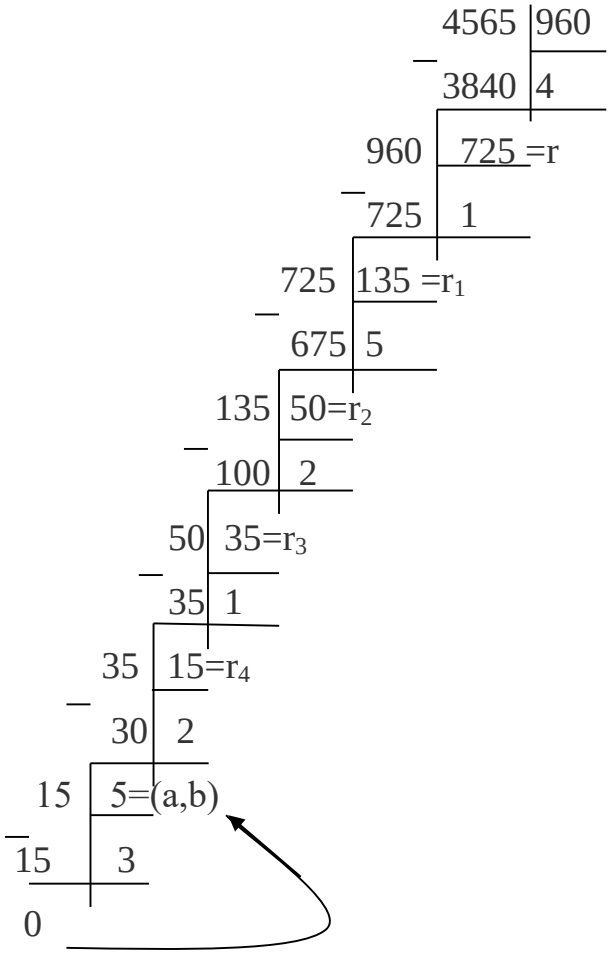
$c_1 = c_2 \cdot q_2 + c_3$ga teng kuchli $c_3 = c_1 - c_2 \cdot q_2$munosabatdan

$c_2 = c_3 \cdot q_3 + c_4$ga teng kuchli $c_4 = c_2 - c_3 \cdot q_3$munosabatdan

mulohazalarni davom ettirib,

$c_{n-2} = c_{n-1} \cdot q_{n-1} + c_n$ga teng kuchli $c_n = c_{n-2} - c_{n-1} \cdot q_{n-1}$ munosabatdan shunga ega bo‘lamiz.Shunday qilib, ta`rifga ko‘ra, $c_n = (a, b)$

Misol. (4565,960)ni topish kerak bo‘lsin. Ketma-ket bo‘lishni ixcham ko‘rinishida quyidagicha yozish mumkin:



Demak, (4565,960) = 5 ekan.

1-ta’rif. a va b natural sonlarining *eng katta umumiy bo‘luvchisi*, ya’ni EKUBi deb, ularning har ikkisini bo‘luvchi sonlar ichidan eng kattasiga aytiladi va $EKUB(a;b)$ (yoki faqat (a, b))shaklda yoziladi.

2-ta’rif . a va b natural sonlarining ***eng kichik umumiy karralisi,*** ya’ni EKUKi deb ularning har ikkisiga karrali natural sonlar ichidan eng kichigiga aytiladi va $EKUK[a;b]$ (yoki faqat [a, b])shaklda yoziladi.

3-ta’rif. Ikki a va b sonlarning **minimumi** deb, ularning kichigiga aytiladi va $min\{a,b\}$shaklda yoziladi.

4-ta’rif. Ikki a va b sonlarning **maksimumi** deb, ularning kattasiga aytiladi va *max* $\{a,b\}$ shaklda yoziladi. Masalan, *min* {0, 2} = 0, *max*{1, 3} = 3.

- $a = 2^m \cdot 3^n \cdot 5^l \cdot \cdots \cdot p^k$ tub ko‘paytuvchilarga ajratilgan a sonining bo‘luvchilari soni $(m+1)\cdot(n+1)\cdot(l+1)\cdot \cdots \cdot(k+1)$ ga teng.

- $a = 2^m \cdot 3^n \cdot 5^l \cdot \dots \cdot p^k$ tub ko‘paytuvchilarga ajratilgan a sonining bo‘luvchilari yig‘indisi:

$$\Upsilon(a) = \frac{2^{m+1}-1}{2-1} \cdot \frac{3^{n+1}-1}{3-1} \cdot \frac{5^{l+1}-1}{5-1} \cdot \dots \cdot \frac{p^{k+1}-1}{p-1}$$

- a va b sonlarning umumiy bo‘luvchilari soni ular EKUBining bo‘luvchilari soniga teng.
- $EKUB(a;b) \cdot EKUK[a;b] = a \cdot b$ tenglik o‘rinli.

Agar a va b natural sonlari $a = 2^{m_1} \cdot 3^{n_1} \cdot 5^{l_1} \cdot \dots \cdot p^{k_1}$ va $b = 2^{m_2} \cdot 3^{n_2} \cdot 5^{l_2} \cdot \dots \cdot p^{k_2}$ tub ko‘paytuvchilarga ajratilgan bo‘lsa, u holda

- $EKUB(a;b) = 2^{min\{m_1,m_2\}} \cdot 3^{min\{n_1,n_2\}} \cdot 5^{min\{l_1,l_2\}} \cdot \dots \cdot p^{min\{k_1,k_2\}}$
- $EKUK[a;b] = 2^{max\{m_1,m_2\}} \cdot 3^{max\{n_1,n_2\}} \cdot 5^{max\{l_1,l_2\}} \cdot \dots \cdot p^{max\{k_1,k_2\}}$
- O‘zaro tub a va b sonlar uchun $EKUB(a;b) = 1$ va aksincha.

5-ta’rif. Agar a son b songa bo‘lishsa, b son a sonning bo‘luvchisi deyiladi.

6-ta’rif. Agar a va b sonlar c songa bo‘linsa a va b c sonining umumiy bo‘luvchisi deyiladi.

Masalan, 24 sonining bo‘luvchilari to‘plami $A = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{6}, 8, \mathbf{12}, 24\}$

36 sonining bo‘luvchilari to‘plami $B = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{6}, 9, \mathbf{12}, 18, 36\}$, bu sonlarning umumiy bo‘luvchilari $A \cap B = \{1,2,3,4,6,12\}$ va ularning eng kattasi 12 ga teng, ya’ni 12=EKUB(24,36).

7-ta’rif. Agar a va b sonlar uchun EKUB (a,b)=1 bo‘lsa bu sonlar o‘zaro tub sonlar deyiladi.

Masalan, 12 va 35 sonlari o‘zaro tub, chunki (12,35)=1.

Sonlarning EKUB va EKUK quyidagi xossalarga ega:

1°. Agar c=(a,b) bo‘lsa, $l = \frac{a \cdot b}{c} = [a, b]$ bo‘ladi. Isbot. $l \vdots b$ ekanligini ko‘rsatamiz

$$c = (a, b) \Rightarrow a = a_1 \cdot c \wedge b \vdots c = b_1 \cdot c$$

$$l = \frac{a \cdot b}{c} = \frac{a_1 \cdot c \cdot b_1 \cdot c}{c} = a_1 \cdot b_1 \cdot c$$

$$\left.\begin{array}{l} l = a_1 \cdot b_1 \cdot c = b_1 \cdot (a, c) = b_1 \cdot a \vdots a \Rightarrow l \vdots a \\ \\ l = a_1 \cdot b_1 \cdot c = a_1 \cdot (b, c) = a_1 \cdot b \vdots a \Rightarrow l \vdots b \end{array}\right| \Rightarrow l = [a, b] ekan$$

2º $k = k \cdot (a_1 b) \Rightarrow k \vdots b \Rightarrow a \cdot k \vdots a \cdot b \Rightarrow a \cdot k \vdots g \cdot k \Rightarrow a \vdots g$

$$g = \frac{a \cdot b}{k} \Rightarrow a \cdot b = g \cdot k$$

3º $(a, b) \cdot [a, b] = \frac{a \cdot b}{k} \cdot [a, b]$= a, b, ya'ni a va b sonlarning eng katta umumiy bo'luvchisi bilan eng kichik umumiy karralisining ko'paytmasi shu sonlar ko'paytmasiga teng.

4º Agar (a,b) = 1 bo'lsa,$[a, b] = a \cdot b$, ya'ni o'zaro tub sonlarning eng kichik umumiy karralisi ularning ko'paytmasiga teng.

5º a va b sonlarning eng katta umumiy bo'luvchisi(EKUB) ularning istalgan umumiy bo'luvchisiga bo'linadi.

6º $((b, c) = 1 \wedge a \vdots b \wedge a \vdots c) \Rightarrow (a \vdots b \cdot c)$,ya'ni a son o'zaro tub bo'lgan b va c sonlarning har biriga bo'linsa, a soni ularning ko'paytmasi $b \cdot c$ ga ham bo'linadi.

Isbot. $=a \vdots b \wedge a \vdots c \Rightarrow a = [b, c] \Rightarrow a \cdot (b, c).\ (b, c) = 1 \Rightarrow [b, c] = b \cdot c.$

Demak, $a \vdots b \cdot c$. 3 va 4 xulosalardan murakkab songa bo'linish alomatlari kelib chiqadi, bunda murakkab son kamida ikkita o'zaro tub sonlar ko'paytmasidan iborat bo'lishi kerak. Bunday alomatlaridan bir nechtasini keltiramiz.

1-alomat. x son 6 ga bo'linishi uchun u 2 ga va 3 ga bo'linishi zarur va yetarli.

2-alomat. x son 12 ga bo'linishi uchun u 3 ga va 4 ga bo'linishi zarur va yetarli va hokazo.

Bunda (2,3)=1, (3,4)=1, shartlar bajarilish kerak.

Mukammal sonlar

Mukammal son deb, o'zining bo'luvchilari yig'indisiga teng bo'lgan natural songa aytiladi. Masalan, eng kichik mukammal son bu – 6 sondir. Chunki, u o'zining bo'luvchilari yig'indisiga teng: 6=3+2+1. Ikkinchi mukammal son bu – 28 bo'ladi. Chunki, u ham o'zining bo'luvchilari yig'indisiga teng: 28=14+7+4+2+1.

Natural sonlar o'z o'qida kattalashib borishi yo'nalishida , mukammal sonlar borgan sari kamayib boradi.

Mukammal sonlar cheksiz ekanligi hali isbotlanmagan . Mukammal sonlar o'ziga hos ketma-ketlik yuzaga keltiradi:

6

28

496

8128

33550336

8589869056

137438691328

2305843008138952128

2658455991569831744654692615953842176

191561942608236107294793378084303638130997321548169

…

Yuqorida aytib o‘tilgan dastlabki ikkita mukammal son - 6 va 28 dan keyingilari- 496 va 8128 bo‘lib, ular mos ravishda:

496=124+62+31+16+8+4+2+1

8128=4064+2032+1016+508+254+127+62+32+16+8+4+2+1

Bo‘luvchilari yig‘indisidan iborat.

Hozirgacha aniqlangan mukammal sonlarning hammasi juft sonlardir. Shu paytgacha, hali birorta toq mukammal son aniqlanmadi. Lekin, toq mukammal sonlarning mavjud emasligi ham matematik isbotlanmagan.

Mukammal sonni aniqlashning usuli, zamonaviy tilda aytadigan bo‘lsak, mukammal sonni aniqlash allgoritmini birinchi bo‘lib Yevklid bayon qilgan. Uning ,,asoslar“ asarining IX jildida, agar $2^p - 1$ ifoda tub son bo‘lsa, unda $2^{p-1}(2^p - 1)$ mukammal juft son bo‘lishi isboti bilan keltirgan edi. Keyinchalik, Leonard Eyler barcha juft mukammal sonlar, Yevklid keltirib o‘tgan yuqoridagi ifodaga bo‘ysunishini isbotlab berdi.

P=2,3,5,7 shartga mos keluvchi dastlabki to‘rtta mukammal sonni, ya’ni 6,28,4096 va 8128 mukammal sonlarini Geras shahridan bo‘lgan Nikomakh (Nikomakh Gerasskiy) ismli matematik o‘zining ,,arifmetika“ nomli kitobida bayon qilgan. Mukammal sonlar qatoridan beshinchisini, ya’ni , 33550336 sonini XV asrda olmon matematigi Regiomontan topgan. XVI asrda, Sheybl ismli yana bir olmon

matematigi keying ikkita mukammal sonni, ya’ni , 8589869056 va 137438691328 ni aniqladi. Ular , p=17 ba p=19 ga mos keladi. 1772-yilida Leonard Eylerning o‘zi p=31 ga mos keluvchi va tartib bo‘yicha sakkizinchi mukammal sonni hisoblab topgan. U 2305843008138952128 soni edi. 1883-yilida Rossiyalik oddiy qishloq matematigi o‘qituvchisi Pervushin tomonidan p=37 ga mos keluvchi, to‘qqizinchi mukammal sonni aniqlagan.

XIX asr oxirida va XX asr boshlarida matematiklar p=89, p=107 va p=127 ga mos keluvchi yana uchta mukammal sonlarni topishgan. Lekin bu inson imkoniyatlari va sabr-toqati bilan topilgan oxirgi mukammal sonlar bo‘ldi. Chunki, e’tibor bergan bo‘lsangiz, endilikda gap milliard yoki trilionlar xonasi haqida emas, balki , undan ham katta sonlar ustida bormoqda. Kompyuter tugul, oddiy kalkulyatorlar ham bo‘lmagan zamonlarda, matematiklar ushbu sonlarni hisoblab topish uchun qanchalik ter to‘kishganini tasavvur qilishning o‘zi qiyin.

Undan keyingi aniqlangan mukammal sonlarning barchasi komyuterlar vositasida topilgan bo‘lib, ularni matematik usul bilan qo‘lda hisoblash har qanday o‘tkir zehnli matematik uchun ham murakkablik qiladi. Hozirda, 2018-yilning 1-aprel holatiga matematika fanida jami 50ta mukammal son aniqlangan bo‘lib ularning barchasi juft sonlardir. Toq mukammal sonlarni qidirish ham davom etmoqda.

Fibonachchi sonlari haqida ma’lumot

Ushbu sonlar ketma –ketligini qaraymiz.

$$u_1, u_2, u_3, \dots , u_n, \dots \qquad (1)$$

n>2 bo‘lganda bu ketma-ketlikning hadlari quyidagi rekurrent munosabatlar orqali berilgan bo‘lsin.

$$u_n = u_{n-1} + u_{n-2} \qquad (2)$$

ya’ni (1) sonlar ketma-ketligining uchinchi hadidan boshlab har bir hadi o‘zidan oldingi ikki had yig‘indisiga teng.

Adabiyotlarda bu (2) rekurrent munosabat bilan aniqlanuvchi (1) sonlar ketma-ketligiga rekurrent yoki qaytariluvchi ketma-ketliklar deyiladi.

O‘z navbatida (2) ketma ketliklar hadlarining hosil bo‘lishiga rekurrent jarayon, (2) tenglikka esa rekurrent tenglama deyiladi.

Bunday ketma-ketliklarni (2) shart yordamida hisoblash mumkin emas, ya`ni (2) shartni qanoatlantiruvchi xohlagancha sonlar ketma-ketligini tuzish mumkin. Masalan:

2, 5, 7, 12, 19, 31, 50…

1, 3, 4, 7, 11, 18, 29…

-1, -5, -6, -11, -17, -28…

va hokazo.

Demak, (1) ketma-ketlikni tuzish uchun (2) shartning o‘zi kifoya emas ekan. Chunki (1) ketma-ketlikning ixtiyoriy hadi uchun undan oldingi ikki had mavjud emas. Shu sababli (1) ketma-ketlikni tuzish uchun uning dastlabki ikki hadi ma’lum bo‘lishi kerak ekan.

Endi mana shu shartni qanoatlantiruvchi ushbu xususiy holni qaraymiz.

Faraz qilaylik (1) ketma-ketlikning dastlabki ikki hadi $u_1 = u_2 = 1$ birga teng bo‘lsin.

Bu qiymatlardan foydalanib, (2) recurrent munosabat yordamida ushbu sonlar ketma-ketligini tuzamiz.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, … (3)

Hosil qilingan (3) sonlar ketma-ketligi **Fibonachchi qatori,** uning hadlari esa **Fibonachchi sonlari** deyiladi.

Arifmetik progressiya.

Arifmetik progressiya deb shunday sonlar ketma-ketligiga aytiladiki, unda ikkinchi hadidan boshlab har bir hadi o‘zidan oldingi hadga shu ketma-ketlik uchun o‘zgarmas bo‘lgan biror d sonni qo‘shish natijasida hosil bo‘ladi.

Masalan, 1) 1, 2, 3, 4, ... 2) 10, 12, 14, 16, . . . ketma-ketliklar arifmetik progressiya tashkil qiladi. Chunki har bir son, ikkinchisidan boshlab, mos ravishda 1 va 2 sonlarini oldingisiga qo‘shish natijasida hosil bo‘ladi.

Arifmetik progressiyani tashkil qiluvchi sonlar uning hadlari deyiladi va umumiy

$$a_1,\ a_2,\ a_3,\ \dots\ ,\ a_{n-1},\ a_n\ ,\dots \qquad (*)$$

(*) ko‘rinishda yoziladi. Arifmetik progressiyaning keyingi hadini hosil qilish uchun oldingi hadiga qo‘shiladigan d son arifmetik progressiya ayirmasi deyiladi. Agar $d > 0$ bo‘lsa, progressiya o‘suvchi, $d < 0$ bo‘lsa, progressiya kamayuvchi deyiladi. Agar $d = 0$ bo‘lsa, arifmetik progressiyaning barcha hadlari o‘zaro teng bo‘ladi. $d = 0$ hol odatda qaralmaydi.

Arifmetik progressiyaning n – hadi a_n quyidagi formula yordamida topiladi:

$$a_n = a_1 + (n-1)\cdot d$$

Arifmetik progressiya hadlarining xossalari:

1° Arifmetik progressiyaning ikkinchi hadidan boshlab istalgan hadi o‘ziga qo‘shni bo‘lgan ikki hadning o‘rta arifmetik qiymatiga teng, ya’ni

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

2° Chekli arifmetik progressiyada boshidan va oxiridan teng uzoqlikda turgan hadlar yig‘indisi chetki hadlar yig‘indisiga teng, ya’ni

$$a_1 + a_n = a_2 + a_{n-1} = a_3 + a_{n-2} = \cdots = a_k + a_{n-k+1}$$

3° Arifmetik progressiyaning dastlabki n ta yig‘indisi, ya’ni

$a_1 + a_2 + a_3 + \dots + a_{n-1} + a_n$ ni S_n bilan belgilaymiz. Arifmetik progressiyaning dastlabki n ta yig‘indisi:

$S_n = a_1 + a_2 + a_3 + \dots + a_{n-1} + a_n$

chetki hadlar yig‘indisining yarmi bilan hadlar soni ko‘paytmasiga teng, ya’ni

$$S_n = \frac{a_1+a_n}{2}\cdot n$$

Arifmetik progressiya xossalarini jamlab, ularni quyidagi tartibda keltiramiz.

1. $a_n = a_1 + (n-1)\cdot d;$ $\qquad a_n = a_{n-1} + d.$
2. $a_n - a_m = (n-m)\cdot d,$ $\qquad n > m.$
3. $a_n = \frac{a_{n-1}+a_{n+1}}{2} = \frac{a_{n-k}+a_{n+k}}{2},$ $\qquad k < n.$
4. $a_k + a_m = a_p + a_q,$ $\qquad k + m = p + q.$
5. $S_n = \frac{a_1+a_n}{2}\cdot n,$ $\qquad S_n = \frac{2\cdot a_1+d\cdot(n-1)}{2}\cdot n$
6. $S_n - S_{n-1} = a_n.$

Geometrik progressiya

Birinchi hadi noldan farqli bo‘lib, ikkinchi hadidan boshlab bir hadi o‘zidan oldingi hadni shu ketma-ketlik uchun o‘zgarmas va noldan farqli bo‘lgan biror q songa ko‘paytirishdan hosil bo‘lgan sonlar ketma-ketligi ***geometrik progressiya*** deyiladi. Masalan, 1) 1, 3, 9, ... 2) 20, 10, 5, ... ketma-ketliklar geometrik progressiya tashkil qiladi. Birinchi misolda $q = 3$, ikkinchisida $q = \frac{1}{2}$.36

Geometrik progressiyani tashkil qiluvchi sonlar uning hadlari deyiladi va umumiy $b_1,\ b_2, b_3, \dots, b_{n-1}, b_n, \dots$ $(**)$

$(**)$ ko‘rinishda yoziladi. Geometrik progressiyaning keyingi hadini hosil qilish uchun oldingi hadiga ko‘paytiriladigan q son geometrik progressiya maxraji deyiladi. Agar $b_1 > 0$ va $q > 1$ bo‘lsa, progressiya o‘suvchi deyiladi. Agar $|q| < 1$ bo‘lsa, progressiya kamayuvchi, $q < 0$ bo‘lsa, progressiya ishorasi o‘zgaruvchi deyiladi, $q = 1$ hol odatda qaralmaydi.

Geometrik progressiyaning $n-$hadi b_n quyidagi formula yordamida topiladi: $b_n = b_1 \cdot q^{(n-1)}$.

Geometrik progressiya hadlarining xossalari.

1° Agar geometrik progressiyaning barcha hadlari musbat bo‘lsa, u holda uning ikkinchi hadidan boshlab istalgan hadi o‘ziga qo‘shni bo‘lgan ikki hadning o‘rta geometrik qiymatiga teng, ya’ni

$$b_n = \sqrt{b_{n-1} \cdot b_{n+1}}\ .$$

2° Chekli geometrik progressiyada boshidan va oxiridan teng uzoqlikda turgan hadlar ko‘paytmasi chetki hadlar ko‘paytmasiga teng, ya’ni

$$b_1 \cdot b_n = b_2 \cdot b_{n-1} = b_3 \cdot b_{n-2} = \dots = b_k \cdot b_{n-k+1}$$

3° Geometrik progressiyaning dastlabki n ta hadi yig‘indisi

$$S_n = b_1 + b_2 + b_3 + \dots + b_{n-1} + b_n$$

bo‘lsin. Geometrik progressiyaning dastlabki *n* ta hadi yig‘indisi $S_n$uchun quyidagi formulalar o‘rinli:

$$S_n = \frac{b_1 - b_n \cdot q}{1-q}, \qquad S_n = \frac{b_n \cdot q - b_1}{q-1}, \qquad S_n = \frac{b_1 \cdot (q^n - 1)}{q-1}\ .$$

4° Cheksiz kamayuvchi geometrik progressiya barcha hadlarining yig‘indisi S uchun quyidagi formula o‘rinli

$$S_n = b_1 + b_2 + b_3 + \cdots + b_{n-1} + b_n + \cdots = \frac{b_1}{1-q}$$

Geometrik progressiya xossalarini jamlab, ularni quyidagi tartibda keltiramiz.

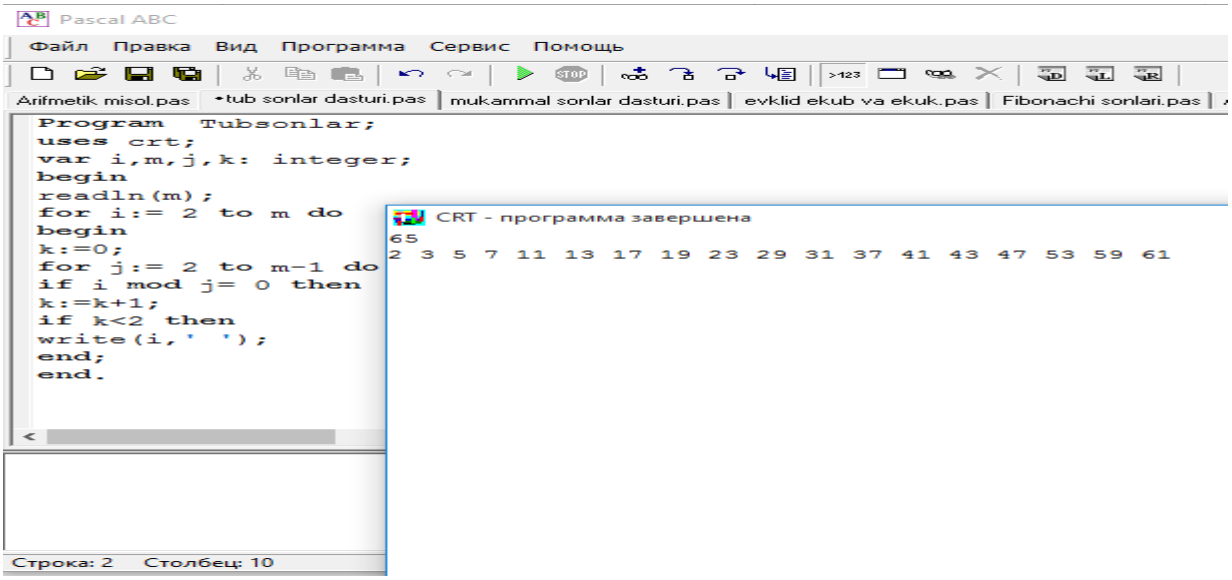
1. $b_n = b_1 \cdot q^{(n-1)}$, $b_n = q \cdot b_{n-1}$.
2. $b_n : b_m = q^{n-m}$ $n > m$.
3. $b_n^2 = b_{n-1} \cdot b_{n+1}$, $n \geq 2$.
4. $b_k \cdot b_m = b_p b_q$, $k + m = p + q$.
5. $S_n = \frac{b_1 - b_n \cdot q}{1-q}$, $S_n = \frac{b_n \cdot q - b_1}{q-1}$, $(q \neq 1)$.
6. $S_n - S_{n-1} = b_n$.
7. $S_n = \frac{b_1}{1-q}$

2.2. Sonlar nazariyasining asosiy algoritmlarini dasturlash texnologiyasi

Tub sonlarni topish dasturi

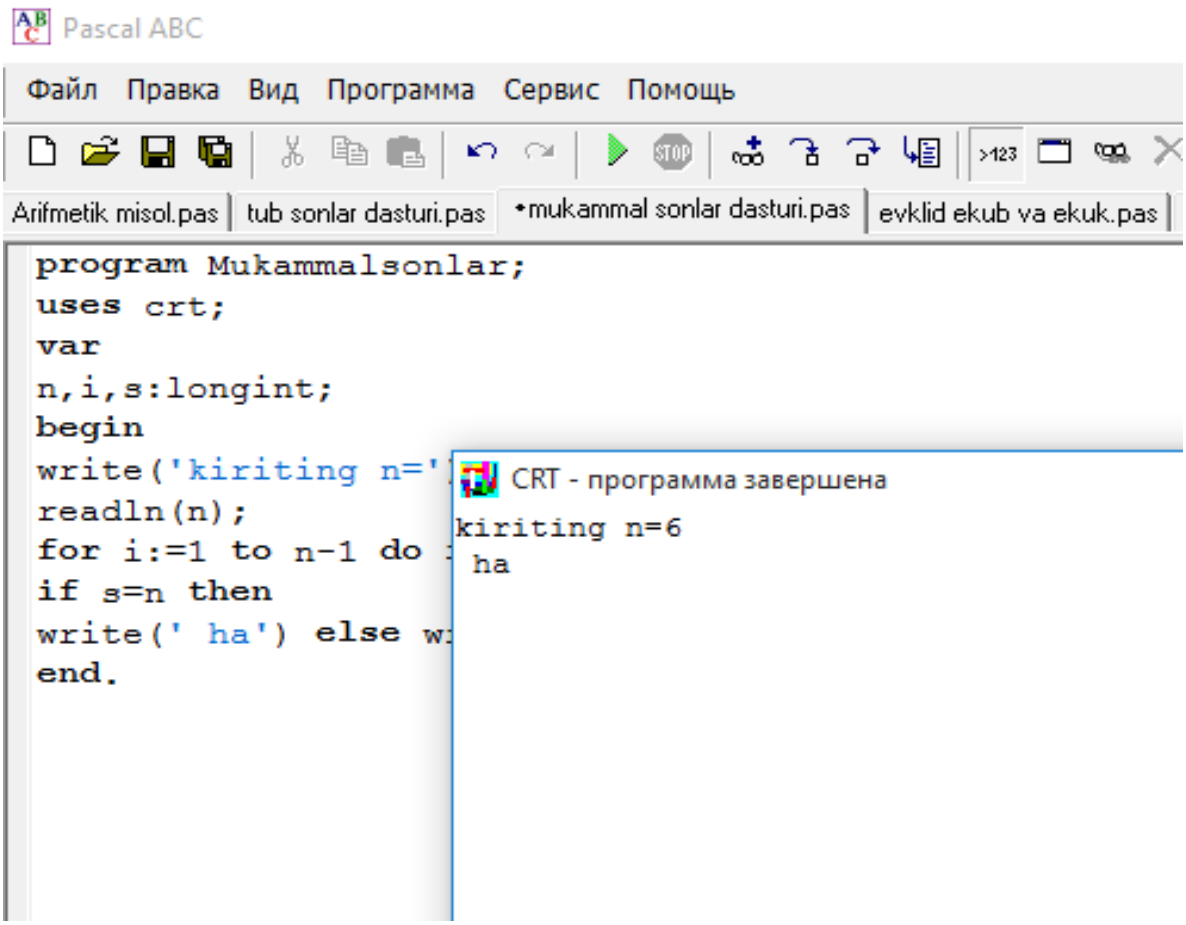
```
Program  Tub sonlar;
var i,m,j,k: integer;
begin
readln(m);
for i:= 2 to m do
begin
k:=0;
for j:= 2 to m-1 do
if i mod j= 0 then
```

```
k:=k+1;
if k<2 then
write(i,' ');
end;
end.
```



Mukammal sonlarni topish dasturi

```
program Mukammal sonlar;
var
n,i,s:longint;
begin
write('kiriting n=');
readln(n);
for i:=1 to n-1 do if n mod i=0 then s:=s+i;
if s=n then
write(' ha') else write(' yoq');
end.
```



Evklid algoritmi. EKUB va EKUK

```
Program  Evklid;
uses crt;
var a,b,p,q,nd,nk:integer;
begin
clrscr;
writeln('Ikkita natural sonni kiriting:');
write('a=');readln(a);
write('b=');readln(b);
a:=abs(a);b:=abs(b);
p:=a;q:=b;
```

```
repeat
if p>q then p:=p mod q
else q:=q mod p;
until (p=0) or (q=0);
nd:=p+q;
writeln('EKUB=',nd);
nk:=a*b div nd;
writeln('EKUK=',nk);
readln
end.
```

Fibonachi sonlari

```
program Fibonachi sonlari;
uses crt;
var n,a,b,c,i:integer;
begin
clrscr;
write(' Qancha Fibonachi sonini chiqarish kerak? n=');
readln(n);
a:=1;
write(a,' ');
b:=1;
```

```
write(b,' ');
i:=1;
while i<n do
begin
c:=a+b;
write(c,' ');
a:=b;
b:=c;
i:=i+1;
end;
readln
end.
```

Arifmetik progressiya

```
program Arifmetik progressiya;
uses crt;
const
n = 10;
var
a:array[1..n] of integer;
i:integer;
d:real;
prog:boolean;
begin
writeln(' Ketma-ketlik hadlarini kiriting‘);
```

```
for i:=1 to n do
begin
write('a[',i,'] = ');
readln(a[i]);
end;
d:=(a[2]-a[1]);
i:=2;
prog := true;
while prog and (i < n) do
begin
if a[i]<>(a[i-1]+a[i+1])/2 then prog := false;
inc(i);
end;
if not prog then writeln(‘Ketma-ketlik arifmetik progressiya emas’)
else writeln(‘Ketma-ketlik arifmetik progressiya  boladi va uning ayirmasi= ',d);
end.
```

Geometrik progressiya

```
program Geometrik progressiya ;
uses crt;
const
n = 5;
var
b:array[1..n] of integer;
i:integer;
q:real;
prog:boolean;
begin
writeln('Ketma-ketlik elementlarini kiriting');
```

```
for i:=1 to n do
begin
write('b[',i,'] = ');
readln(b[i]);
end;
q:= b[2]/b[1];
i:=2;
prog := true;
while prog and (i < n) do
begin
if b[i]= sqrt(b[i-1]/b[i+1]) then prog := false;
inc(i);
end;
if not prog then writeln('ketma-ketlik geometrik progressiya emas')
else writeln(' ketma-ketlik geometrik progressiya  bo‘ladi  va uning maxraji  = ',q);
end.
```

Xulosa

Ma’lumki, talabalarga dasturlash asoslarini o’rgatishda biz ,asosan, matematika fanidagi asosiy algoritmlardan keng foydalanamiz. Masalan, geometriyadagi Geron formulasi va h.k. Lekin matematikani yana shunday sohasi mavjudki, undagi algoritmlar juda qadimdan mashhurdir. Bu yerda biz sonlar nazariyasini nazarda tutyapmiz. Bu malakaviy bitiruv ishi Turbo Pascal tilida sonlar nazariyasining asosiy algoritmlarini dasturlash texnologiyasiga bag’ishlangan bo’lib, ushbu ish kirish, 2 bob, xulosa va foydalanilgan adabiyotlar ro’yxatidan iborat.

Birinchi bobda Paskal dasturlash tili haqida batafsil ma’lumot beriladi. U 2 paragrafdan iborat bo’lib, birinchi paragrafda paskal dasturlash tili haqida umumiy ma’lumotlar beriladi. Ikkinchi paragrafda Paskal tilida dasturlash texnologiyasi ko’rib chiqiladi.

Ikkinchi bobning birinchi paragrafida sonlar nazariyasining asosiy tushuncha va algoritmlari yoritiladi. Tub sonlar ta’rifi, tub sonlarning cheksizligi haqidagi Evklid teoremasi, mukamall sonlar haqida ma’lumot,eng kichik umumiy bo’luvchi, eng katta umumiy karrali, Fibbonachi sonlari, arifmetik va geometrik progressiyalarning asosiy hossalari haqida alohida to’xtalgan. Ikkinchi paragrafda ushbu algoritmlarni Paskal tilida dasturlash texnologiyasi keltiriladi.

Tadqiqotimizning asosiy maqsadi Paskal algoritmik tilini o’rganuvchi talabalarga dasturlash texnologiyalarini o’rgatishdan iborat.

Bitiruv malakaviy ishi shunday tuzilganki, unda Turbo Pascal dasturlash tili o’rniga boshqa dasturlash tili, masalan, C++ tilini ham bemalol o’rgansa ham bo’laveradi.

Foydalanilgan adabiyotlar:

1. Karimov I. A. O‘zbekiston buyuk kelajak sari.—Toshkent.: «O‘zbekiston», 1998.—528 6.

2. Barkamol avlod — O‘zbekiston taraqqietining poydevori. (O‘zbekiston Respublikasining «Ta’lim to‘g‘risida» va «Kadrlar tayerlash milliy dasturi to‘g‘risida»gi konunlar).—T.: SHark, 1998.-64 b.

3. Bobojonov K. Paskal dasturlash tilida o‘zgaruvchilarni tasvirlash. //”Xalq ta’limi” jurnali. 2002. №1. 95-97-betlar.

4. Algoritmik tillar va dasturlash. Namangan, 2003. Ma’ruza matni.

5. Informatika. Ma’ruza matni. Toshkent, 2000. – 38 bet.

6. Култин Н. Программирование в Turbo Pascal 7.0 и Delphi, 2-е изд., С. Петербург, БХВ-С.Петербург, 1999.

7. V.E.Figurnov. IBM PC для пользователя. M.1997. -

8. A. Ketkov. Диалог на языке бейсик для мини ЭВМ. M., 1996. -

9. C++ tilida programmalash asoslari. Madraximov Sh.F.,Gaynazarov S.M. Toshkent 2009

Internet resurslar

1. http://dastur.uz
2. http://www.edu.uz
3. http://www .acm.tuit.uz
4. http://www.arxiv.uz
5. http://www.intuit.ru
6. http://www.exponenta.ru