

Toshkent Davlat Texnika Universiteti

Kurs ishi

**Mavzu : Massivlar ustida amallar
C++ tilida**

**Bajardi: 100-13
Agzamov J.**

**Tekshirdi:
Toshmatova Sh**

Toshkent – 2014

Mundarija:

<i>Kirish</i>	<i>1</i>
<i>Mazariy qism</i>	
<i>Masala shartini o'rganish va tahlil qilish.....</i>	<i>2</i>
<i>Algoritmlar</i>	
<i>Algoritmlar haqida tushuncha.....</i>	<i>4</i>
<i>Algoritmlarning xossalari.....</i>	<i>5</i>
<i>Algotimning ifodalanishi.....</i>	<i>5</i>
<i>Algotimning blok-sxema shaklida ifodalanishi.....</i>	<i>6</i>
<i>C++tili haqida.....</i>	<i>7</i>
<i>Kurs ishi masalasinig algoritmi.....</i>	<i>8</i>
<i>Dasturi.....</i>	<i>9</i>
<i>Dasturning natijasi.....</i>	<i>10</i>
<i>Foydalanilgan adabiyotlar ro'yxati.....</i>	<i>12</i>

Kirish

Respublikamizda energrtik va xom-ashyo resurslaridan unumli foydala-nish davrida kompyuter texnologiyalaridan milliy iqtisodiyotni boshqarishda foydalanish toboro muhim bo;lib bormoqda. Kompyuterlar ayniqsa bank ishlarida va soliq qo'mitasi ma'lumotlatini yig'ish, tahlil qilishda katta ahamiyatga ega. Bundan tashqari kompyuterlar telivideniya, radio aloqalarni sifatini ta'minlashda, aloqa kompaniyalarining abonentlarga sifatli va teskor xizmat ko'rsatishda qo'llanilmoqda. Kompyuterlar puxta ishlangan dastur asosida bexato va unumli xizmat ko'rsata olodilar. Xususan, texnologik jarayonlari kompyuterlar orqali boshqariladigan sanat korxonalaridan puxta va sifatli maxsulotlar ishlab chiqariladi. Hozirgi kunda kompyuterlarni va shu soha mutaxasislarini jamiyatimizning kerakli sohalariga tadbiq etish muhim ahamiyatga ega.

Kurs ishi C++ dasturlash tili asosida tuzilgan bo'lib, kurs ishi masalasiga taaluqli nazariy qism, masalaning algoritmi, dasturi, dastur narijasi va izohlari-dan tashkil topgan. Nazariy qisimda tenglamani teng ikki qisimga bo'lib ishlash, Bolsano-Koski teoremasi bilan tanishish, algoritmlar,ularning xossalari ,ifodalanishi va blok-sxema ko'rinishlari haqida ma'lumot olish, C++ dasturlash tili va uning imkoniyatlari haqida ma'lumotga ega bo'lishingiz mumkin. Masalani algoritmi va dasturi imkon darajasida izohlar bilan keltirilgan.

Nazariy qism.

Masala shartini o'rganish va tahlil qilish.

Masalani berilishi:

Ushbu $z = ax^2 + bx + c$ funksiya hisoblansin, bu yerda $x: e^x + x + 1 = 0$ tenglamaning $[a;b]$ kesmadagi yechimi, bu tenglama ixtiyoriy usul bilan yechilsin.

$$a = \sum_{i=1}^{10} M_i ; \quad b = \sum_{i=1}^{10} N_i ; \quad c = \frac{\ln|a| - tgy}{q|a - 2| + y} ; \quad q = 0,125 ; \quad y = 32,56.$$

M va N massiv elementlarining son qiymatlarini ixtiyoriy oling.

Masalani tahlil qilish:

Bizga berilgan M va N massivlar asosida ularning elementlari yig'indisi a va b larni topish mumkin. Topilgan a , b va masalada berilgan q va y lar orqali c ifodaning qiymatini topish mumkin. Masala shartidagi z funksiyani hisoblashimiz uchun bizga a, b va c lar ma'lum bo'ladi, ammo x ni topish qoldi. Endi biz $e^x + x + 1 = 0$ tenglamaning $[a;b]$ kesmadagi yechimini topish bilan shug'ullanamiz. Biz bu tenglamani yechish uchun oraliqni teng ikkiga bo'lish usulidan foydalanamiz.

Bizga $f(x)$ funksiya berilgan bo'lsin va $[a;b]$ kesmada yechimga ega bo'lsin. $f(x)=0$ tenglamaning yechimini topamiz. Bu tenglamani yechish uchun, Bolsano-Koshi teoremasidan foydalanamiz.

Teorema(Bolsano-Koshi): $[a;b]$ kesmada uzliksiz $f(x)$ funksiya kesmaning oxirlarida turli ishorali qiymatlar qabul qilsa, u holda kesmada kamida bitta nuqta mavjudki, funksiya shu nuqtada 0 (nol)gat eng bo'ladi.

Ushbu teoreмага asoslangan holda quyidagi hollarni qarab chiqamiz va u yerda x_0 - kesmaning o'rtasi ($x_0 = (a + b) / 2$).

1-holda. $f(x)$ funksiya o'suvchi bo'lsin. U holda kesmaning o'rtasini x_0 nuqta deb olamiz va uni funksiya qo'yib ko'ramiz. Agar $f(x)$ funksiyaning x_0 nuqtadagi qiymati musbat bo'lsa, $[a; x_0]$ kesmadan ildizni qidirish kerak(1-chizma). Agar $f(x)$ funksiyaning x_0 nuqtadagi qiymati manfiy bo'lsa, $[x_0; b]$ kesmadan ildizni qidirish kerak(2-chizma). Agar $f(x)$ funksiya x_0 nuqta-da 0 (nol)gat eng bo'lsa, biz izlayotgan natija osongina yuzaga keladi, ya'ni x_0 nuqta berilgan $f(x)$ funksiyaning ildizi bo'ladi.

2-holda. $f(x)$ funksiya kamayuvchi bo'lsin. U holda kesmaning o'rtasini x_0 nuqta deb olamiz va uni funksiya qo'yib ko'ramiz. Agar $f(x)$ funksiyaning shu nuqtadagi qiymati musbat bo'lsa, $[x_0; b]$ kesmadan ildizni qidirish kerak(3-chizma). Agar $f(x)$

funksiyaning x_0 nuqtadagi qiymati manfiy bo'lsa, $[a; x_0]$ kesmadan ildizni qidirish kerak(4-chizma). Agar $f(x)$ funksiya x_0 nuqta-da 0 (nol)gat eng bo'lsa, biz izlayotgan

Demak, bu usulda tenglama yechish uchun, bizga berilgan tenglamani chap tomonini funksiya deb qarab, uni o'suvchi yoki kamayuvchi ekanligini va funksiya qiymati 0 (nol)ga teng bo'ladigan $[a;b]$ kesmani topish yetarli.

Endi biz funksiyaning o'suvchi yoki kamayuvchi ekanligini va funksiya qiymati 0 (nol)ga teng bo'ladigan $[a;b]$ kesmani topish bilan shug'ullanamiz.

Bizga $f(x) = e^x + x + 1$ bo'lsin, u holda funksiya hosilasi 0 (nol)dan katta ($f'(x) > 0$)

bo'lsa, o'suvchi, aks holda funksiya kamayuvchi bo'ladi.

$$f(x) = e^x + x + 1$$

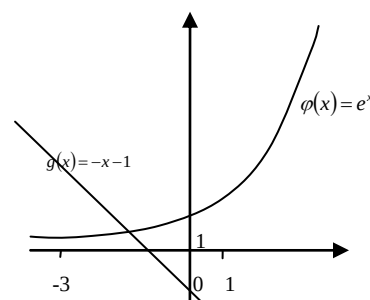
$$f'(x) = e^x + 1$$

$f'(x) > 0$ ekanligi ko'rinib turibdi, demak, $f(x)$ funksiya o'suvchi ekan.

Endi $f(x)$ funksiya ildizi joylashgan ya'ni $f(x) = 0$ bo'ladigan $[a;b]$ kesmani topish kifoya. Buning uchun $f(x)$ funksiya sonlarni o'sish yoki kamayish tartibida qo'yib ko'riladi va bu holat funksiya ishorasi o'zgargunga qadar davom ettiriladi. Shundan so'ng funksiya ishorasini o'zgartirgan son va undan bitta oldin berilgan son orqali kesma yasaladi va bu kesma biz izlayotgan kesma hisoblanadi.

Bizga berilgan funksiyaning grafik ravishda ifoda etish qiyin emas shuning uchun bu kesmani grafik usul bilan aniqlaymiz. Bunda biz $e^x + x + 1 = 0$ ifodani $e^x = -x - 1$ ko'rinishiga keltiramiz va tenglikning har ikki qismini $\varphi(x) = e^x$ va $g(x) = -x - 1$ funksiyalar deb qarab, ularning kesishish oralig'ini topomiz(5-chizma). Chizmadan ko'rinadiki $[-3;1]$ kesmada funksiyalar albatta kesishadi. Bu kesmada $f(x) = 0$ bo'lishini Bolsano-Koshi teoremasi orqali isbotlaymiz.

$$\begin{aligned} f(x) &= e^{-3} - 3 + 1 = e^{-3} - 2 & \text{va} & \quad f(x) < 0 \\ f(x) &= e^1 + 1 + 1 = e + 2 & \text{va} & \quad f(x) > 0 \end{aligned}$$



5-chizma

Demak, $f(x)$ funksiya $[-3;1]$ kesmada kamida 0 (nol)gat eng bo'ladigan kamida bitta nuqtasi mavjud.

$f(x)$ funksiya yechimga ega bo'lishi uchun M va N massivlar elementlaridan yasalgan $[a;b]$ kesma $[-3;1]$ kesmani o'z ichiga olishi kerak.

Biz kampyatrni ortiqcha zo'riqishdan holi qilish uchun M massiv elementlari yig'indisi - 3 va N massiv elementlari yig'indisi 1 bo'ladigan qilib tanlaymiz, ya'ni $M[10]=\{0,1,2,-3,4,-5,6,-7,8,-9\}$, $N[10]=\{0,1,-2,3,4,-5,-6,7,8,-9\}$.

Algoritmlar

Algoritmlar haqida tushuncha.

Algoritm so'zi *algarithmi* so'zidan olingan bo'lib, u IX asirning buyuk matematigi Muhammad al-Xorazmiy nomining lotincha shaklidir.

Informatika sohasida algoritm tushunchasi asosiy tushuncha bo'lib, u geometriya kursidagi nuqta, to'g'ri chiziqli va tekislik, matematikadagi to'plam, kimyodagi modda, fizikadagi fazo va vaqt tushunchalari kabi fundamental tushuncha hisoblanadi.

Algoritmga aniq bir ta'rif berish mushkul. Shunday bo'lsada algoritmning mohiyatini

aniq tushuntirish mumkin. Algoritm – biror masalani yechish uchun bajarilishi lozim bo'lgan buyruqlarning tartibli ketma-ketligidir. Tuzilgan algoritmni uning tuzilish qoidalarini tushunadigan va unda ko'rsatilgan buyruqlarni bajarish imkoniga ega bo'lgan insonning o'zi yoki texnik qurulma (masalan, kompyuter) bajarishi mumkin.

Odamlar har kuni bajaradigan ishlarini o'zlari bilmagan holda shu ishlarni bajarish algoritmilaridan foydalanishadi. Masalan kompyuterdan foydalanish, non topish, taom tayyorlash, telefon aparatidan foydalanish, avtomobilni boshqarish, kitob o'qish, ko'cha harakati qoidalariga rioya qilish, televizor yoki radiodan foydalanish va hokazo.

Algoritmlarning xossalari.

Algoritmlarga quyidagi qo'shimcha talablar mavjud bo'lib, bu talablar algoritmning xossalari deb yuritiladi. Ular quyidagilardan iborat:

1.Uzliksizlik xossasi. Bu xossaga ko'ra algoritm aniq va tugallangan qadamlarga bo'lingan bo'lishi kerak.

2.Aniqlilik xossasi. Algoritm shunday tuzilgan bo'lishi kerakki, har bir buyrig'i bajarilgandan keyin qaysi buyrig'i bajarilishi aniq ko'rsatilgan bo'lishi kerak.

3.Tushinarlilik xossasi. Har bir algaritm yoshi yoki qobiliyatiga qarab malum ijrochiga mo'ljallangan bo'ladi. Masalan, kvadrat tenglamani yechish algoritmi to'g'ri tuzilgan bo'lsa ham, boshlang'ich sinf o'quvchilariga tushinarli bo'lmasligi tufayli ular bu algoritmni ijro eta olmaydilar. Chunki ular diskriminant, kvadra ildiz, kvadratga ko'tarish kabi tushinchalar bilan tanish emas. Bundan tashqari, ushbu xossaga binoan algoritmning har bir buyrug'i aniq ifodalangan bo'lib, ikki xil manoni anglatmasligi lozim, ya'ni algoritmning har bir buyrug'i barcha ijrochilar tomonidan bir xil tushinilishi kerak.

4. Natijaviylik xossasi. Ijrochi algoritmning chekli sondagi buyruqlarini bajargandan so'ng, natijaga erishishi kerak.

5.Ommaviylik xossasi. Bu xossaga ko'ra tuzilgan bitta algoritm vositalariga faqat bitta emas, balki bir xil turli barcha masalalarni hal qila bilishi mumkin bulishi kerak.

Masalan, Kvadrat tenglamani yechish algoritmi yordamida ixtiyoriy kvadrat tebglamani yechish, yoki $y = a \cdot x + b$ to'g'ri chiziqli grafigini chizish algoritmi yordamida ixtiyoriy to'g'ri chiziqli grafigini chizish mumkin bo'lishi kerak.

Algotimning ifodalanishi.

Algoritm ijrosini EHM yoki avtomatik qurilmalar ham bo'lishi mumkin. EHM o'ziga tushunarli bo'lgan tilda tuzilgan algoritm (ya'ni dastur) asosidagina ishlay oladi. Demak, EHM uchun tuzilgan dasturni ham tuliq ma'noda *algoritm* deb atash mumkin.

Algoritmlarni tuzish jarayonida ularning qaysi ijrochiga muljallab tuzilayotganligiga alohida e'tibor beriladi va su'niy yoki tabiiy tillardan birida tuziladi. Ma'lumki, EHM tabiiy tillarni tushunmaydi va inson bilar ular o'rtasida bu tillarda muloqat o'rnatib bo'lmaydi. SHuning uchun inson bilan EHM o'rtasida muloqat o'rnatishga mo'ljallangan bir necha o'nlab tillar yaratilgan. Bu tillar su'niy tillar jumlasiga kiradi va *dasturlash tillari* deb ataladi. Agar algoritm EHM uchun muljallab tuzilayotgan bo'lsa, mazkur EHM tushunadigan su'niy tillardan birida tuziladi va to'liq formallashtiriladi.

Algoritm qaysi ijrochiga mo'ljallab tuzilayotganligiga qarab quyidagi shakllardan birida ifodalanishi mumkin:

1.Matn shakli; 2.Blok-sxema shakli; 3.Dastur shakli.

Algoritmlarni ifodalashning dastlabki ikkita shakli tabiiy tillar: matematik formulalar ba simvollar vositasida ifodalanadi. Uchinchi shakli esa sun'iy tillardan birida ifogalanadi.

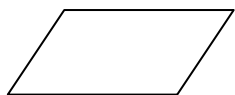
Algotimning blok-sxema shaklida ifodalanishi.

Matematikada masalalarni hisoblashda algoritmni blok-sxema tasvirlash uni tushunish uchun qulaylik tug'diradi. Blok-sxema ko'rsatgichlar bilan bo'g'anuvchi har xil blok ketma-ketligidan tashkil topadi. Algoritmning bajarish taryibi ko'rsatiladi.

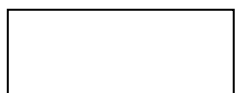
Blok-sxema algoritmning aloxida boaqichlarini belgilash uchun quyidagi belgilardan foydalaniladi:



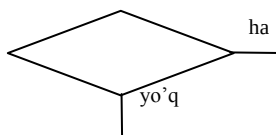
- algoritmning boshlanishi yoki tamomlanishini bildiruvchi blok;



- kerakli ma'lumotlarni kiritish yoki chiqarish bloki;



- hisoblash va ta'minlash bloki;



- shartni tekshiruvchi blok;



- siklni kiritish bloki;



- ma'lumotlarni printeriga chiqorish bloki;



- funktsiyani kiritish bloki;

Har qanday algoritm blok-sxema shaklida ifodalanganda har doim uning boshlaishini bildiruvchi blok bilan boshlanib, algoritmning tugallanganligini bildiruvchi blok bilan yakunlanadi.

Kerakli ma'lumotlarni kiritish va chiqarish blokida dastlabki berilganlar kiritiladi va olingan natijalar chiqariladi.

Shartni tekshirish bloki ichiga shart yoziladi, qo'yilgan shartning bajarilish-bajarilmasligiga qarab strelka ikki tomonga yo'naladi. Strelkaning biriga "ha" so'zi ikkinchisiga "yo'q" so'zi yoziladi. "Ha" so'zi shartni bajarilganligini "yo'q" so'zi esa shartni bajarilmaganligini bildiradi. Shartni bajarilgan yoki bajarilmaganligiga qarab, algoritmning bajarilish tartibi aniqlanadi.

Hisoblash va ta'minlash blokida hisoblanishi kerak bo'lgan ifoda yoziladi. Ifodaning qiymati hisoblanib, tenglik belgisini chap tomonidagi o'zgaruvchiga ta'minlanadi.

C++tili haqida.

C++ dasturlash tili C tiliga asoslangan. C esa o'z navbatida B va BCPL tillaridan kelib chiqqan. BCPL 1967 yilda Martin Richards tomonidan tuzilgan va operatsion sistemalarni yozish uchun mo'ljallangan edi. Ken Thompson o'zining B tilida BCPL ning ko'p hossalarni kiritgan va B da UNIX operatsion sistemasining birinchi versiyalarini yozgan. BCPL ham, B ham tipsiz til bo'lgan. Yani o'zgaruvchilarning ma'lum bir tipi bo'lmagan- har bir o'zgaruvchi kompyuter hotirasida faqat bir bayt yer egallagan. O'zgaruvchini qanday sifatda ishlatish esa, yani butun sonmi, kasrli sonmi yoki harfdekmi, dasturchi vazifasi bo'lgan.

C tilini Dennis Ritchie B dan keltirib chiqardi va uni 1972 yili ilk bor Bell Laboratoriyasida, DEC PDP-11 kompyuterida qo'lladi. C o'zidan oldingi B va BCPL tillarining juda ko'p muhim tomonlarini o'z ichiga olish bilan bir qatorda o'zgaruvchilarni tiplashtirdi va bir qator boshqa yangiliklarni kiritdi. Boshlanishda C asosan UNIX sistemalari keng tarqaldi. Hozirda operatsion sistemalarning asosiy qismi C/C++ da yozilmoqda. C mashina arhitekturasiga bog'langan tildir. Lekin yaxshi rejalashtirish orqali dasturlarni turli kompyuter platformalarida ishlaydigan qilsa bo'ladi.

1983 yilda, C tili keng tarqalganligi sababli, uni standartlash harakati boshlanadi. Buning uchun Amerika Milliy Standartlar Komiteti (ANSI) qoshida texnik komitet tuzildi. Va 1989 yilda ushbu standart qabul qilindi. Standartni dunyo bo'yicha keng tarqatish maqsadida 1990 yilda ANSI va Dunyo Standartlar Tashkiloti (ISO) hamkorlikda C ning ANSI / ISO 9899:1990 standartini qabul qilishdi. Shu sababli C da yozilgan dasturlar kam miqdordagi o'zgarishlar yoki umuman o'zgarishsiz juda ko'p kompyuter platformalarida ishlaydi.

C++ 1980 yillar boshida Bjarne Stroustrup tomonidan C ga asoslangan o'z ichiga olgan, lekin eng asosiysi u ob'ektlar bilan dasturlashga imkon beradi.

Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoqda. Buni ta'minlash uchun ob'ektli dasturlash g'oyasi ilgari surildi. Huddi 70-chi yillar boshida strukturali dasturlash kabi, programmalarni hayotdagi jismlarni modellashtiruvchi ob'ektlar orqali tuzish dasturlash sohasida inqilob qildi.

C++ dan tashqari boshqa ko'p ob'ektli dasturlashga yo'naltirilgan tillar paydo bo'ldi. Shulardan eng ko'zga tashlanadigani Xerox ning Palo Alto da joylashgan ilmiy-qidiruv markazida (PARC) tuzilgan Smalltalk dasturlash tilidir. Smalltalkda hamma narsa ob'ektlarga asoslangan. C++ esa gibrid tildir. Unda C ga o'hshab strukturali dasturlash yoki yangicha, ob'ektlar bilan dasturlash mumkin. Yangicha deyishimiz ham nisbiydir. Ob'ektli dasturlash falsafasi paydo bo'lganiga ham yigirma yildan oshyapti. C++ funksiya va ob'ektlarning juda boy kutubhonasiga ega. Yani C++ da dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ ni o'zini o'rganish, ikkinchisi esa C++ ning standart kutubhonasidagi tayyor ob'ekt funksiyalarni qo'llashni o'rganishdir.

TIL TUZILISHI.

1. Alfavit, identifikator, xizmatchi so'zlar.

Alfavit. C++ alfavitiga quyidagi simvollar kiradi.

- Katta va kichik lotin alfaviti xarflari (A,B,...,Z,a,b,...,z)
- Raqamlar: 0,1,2,3,4,5,6,7,8,9
- Maxsus simvollar: " , { } | [] () + - / % \ ; ` . : ? < = > _ ! & * # ~ ^
- Ko'rinmaydigan simvollar ("umumlashgan bushliq simvollar"). Leksemalarni uzaro ajratish uchun ishlatiladigan simvollar (misol uchun bushlik, tabulyatsiya, yangi qatorga o'tish belgilari).

Izohlarda, satrlarda va simvolli konstantalarda boshqa literalalar, masalan rus xarflarini ishlatilishi mumkin.

C++ tilida olti xil turdagi leksemalar ishlatiladi: erkin tanlanadigan va ishlatiladigan identifikatorlar, xizmatchi suzlar, konstantalar(konstanta satrlar), amallar(amallar belgilari), azhratuvchi belgilar.

Identifikator.Identifikatorlar lotin xarflari,ostki chiziq belgisi va sonlar ketma ketligidan iborat buladi. Identifikator lotin xarfidan yoki ostki chizish belgisidan boshlanishi lozim.

Misol uchun:

A1, _MAX, adress_01, RIM, rim

Katta va kichik xarflar farklanadi, shuning uchun ohirgi ikki identifikator bir biridan farq qiladi.

Borland kompilyatorlaridan foydalanilganda nomning birinchi 32 xarfi ,ba'zi kompilyatorlarda 8 ta xarfi inobatga olinadi. Bu holda NUMBER_OF_TEST va NUMBER_OF_ROOM identifikatorlari bir biridan farq qilmaydi.

Xizmatchi so'zlar. Tilda ishlatiluvchi ya'ni dasturchi tomonidan uzgaruvchilar nomlari sifatida ishlatish mumkin bulmagan identifikatorlar xizmatchi so'zlar deyiladi.

C ++ tilida quyidagi xizmatchi so'zlar mavjud:

int	extern	else
char	register	for
float	typedef	do
double	static	while
struct	goto	switch
union	return	case
long	sizeof	default
short	break	entry
unsigned	continue	
auto	if	

O'zgaruvchilar. (VARIABLES)

O'zgaruvchilar ob'ekt sifatida. C++ tilining asosiy tushunchalaridan biri nomlangan hotira qismi – ob'ekt tushunchasidir. Ob'ektning xususiy holi bu o'zgaruvchidir. O'zgaruvchiga qiymat berilganda unga ajratilgan hotira qismiga shu qiymat kodi yoziladi. O'zgaruvchi qiymatiga nomi orqali murojaat qilish mumkin, hotira qismiga esa faqat adresi orqali murojaat qilinadi. O'zgaruvchi nomi bu erkin kiritiladigan identifikatordir. O'zgaruvchi nomi sifatida xizmatchi so'zlarni ishlatish mumkin emas.

O'zgaruvchilar tiplari. O'zgaruvchilarning qo'yidagi tiplari mavjuddir:

char – bitta simvol;
long char – uzun simvol;
int – butun son;
short yoki **short int** – qisqa butun son;
long yoki **long int** – uzun butun son;
float - haqiqiy son;
long float yoki **double** – ikkilangan haqiqiy son;
long double – uzun ikkilangan haqiqiy son;

Butun sonlar ta'riflanganda ko'rilgan tiplar oldiga **unsigned** (ishorasiz) ta'rifi kushilishi mumkin. Bu ta'rif qushilgan butun sonlar ustida amallar **mod 2ⁿ** arifmetikasiga asoslangandir. Bu erda n soni **int** tipi hotirada egallovchi razryadlar sonidir. Agar ishorasiz k soni uzunligi **int** soni razryadlar sonidan uzun bulsa, bu son qiyjmati $k \bmod 2^n$ ga teng bo'ladi. Ishorasiz k son uchun $-k$ amali $2^n - k$ formula asosida hisoblanadi. Ishorali ya'ni **signed** tipdagi sonlarning eng katta razryadi son ishorasini ko'rsatish uchun ishlatilsa **unsigned** (ishorasiz) tipdagi sonlarda bu razryad sonni tasvirlash uchun ishlatiladi.

O'zgaruvchilarni dasturning ixtiyoriy qismida ta'riflash yoki qayta ta'riflash mumkin.

Misol uchun:

Int a, b1, ac; eki

Int a;

int b1;

int ac;

O'zgaruvchilar ta'riflanganda ularning qiymatlari aniqlanmagan bo'ladi. Lekin o'zgaruvchilarni ta'riflashda initsializatsiya ya'ni boshlang'ich qiyjmatlarini ko'rsatish mumkin.

Misol uchun:

Int I=0;

Char c='k';

Typedef ta'riflovchisi yangi tiplarni kiritishga imkon beradi.

Misol uchun yangi COD tipini kiritish:

Typedef unsigned char COD;

COD simbol;

KONSTANTALAR. (CONSTANTS)

Konstanta bu o'zgartirish mumkin bulmagan qiymatdir. C++ tilida besh turdagi konstantalar ishlatilishi mumkin: butun sonlar, haqiqiy sonlar, simvollar, sanovchi konstantalar va nul kursatkich.

1. Ma'lumotlarning butun son turi.

Butun sonlar o'nlik, sakkizlik yoki un oltilik sanoq sistemalarida berilishi mumkin.

O'nlik sanoq sistemasida butun sonlar 0-9 raqamlari ketma ketligidan iborat bo'lib, birinchi raqami 0 bulishi kerak emas.

Sakkizlik sanoq sistemasida butun sonlar 0 bilan boshlanuvchi 0-7 raqamlaridan iborat ketma ketlikdir.

O'n oltilik sanoq sistemasida butun son 0x eki 0X bilan boshlanuvchi 0-9 raqamlari va a-f yoki A-F xarflaridan iborat ketma ketlikdir.

Masalan 15 va 22 o'nlik sonlari sakkizlikda 017 va 026, un oltilikda 0xF va 0x16 shaklda tasvirlanadi.

Ma'lumolarning uzun butun son turi.

Oxiriga l eki L harflari quyilgan o'nlik, sakkizlik yoki o'n oltilik butun son.

Ma'lumotlarning ishorasiz (unsigned) butun son turi:

Ohiriga u yoki U harflari quyilgan o'nlik, sakkizlik yoki o'n oltilik oddiy yoki uzun butun son.

2. Ma'lumotlarning haqiqiy son turi:

Olti qismdan iborat bulishi mumkin: butun qism, nuqta, kasr qism, yoki E belgisi, o'nlik daraja, F eki f suffikslari.

Masalan : 66. .0 .12 3.14F 1.12e-12

Ma'lumolarning uzun haqiqiy son turi :

Ohiriga L eki l suffikslari quyilgan haqiqiy son.

Masalan: 2E+6L;

3. Simvolli konstanta.

Bittalik qavslarga olingan bitta yoki ikkita simvol. Misol uchun 'x', '*', '\012', '\0', '\n'- bitta simvolli konstanta; 'dd', '\n\t', '\x07\x07' ikki simvolli konstantalar.

'\ ' simvolidan boshlangan simvollar eskeyp simvollar deyiladi. Simvolli konstanta qiymati simvolning kompyuterda qabul qilingan sonli kodiga tengdir.

ESC (eskeyp) simvollar jadvali:

Yozilishi	Ichki kodi	Simvoli (nomi)	Ma'nosi
\a	0x07	bel (audible bell)	Tovush signali
\b	0x08	Bs (bascspase)	Bir qadam qaytish
\f	0x0C	Ff (form feed)	Sahifani qaytarish
\n	0x0A	lf (line feed)	Qatorni o'tkazish
\r	0x0D	Cr (carriage return)	Karetkani qaytarish
\t	0x09	Ht (horizontal tab)	Gorizontal tabulyatsi
\v	0x0B	Vt (vertical tab)	Vertikal tabulyatsi
\\	0x5C	\ (bacslash)	Teskari chiziq
\'	0x27	' (single out)	Apostrif (oddiy qavs)
\"	0x22	" (double quote)	Ikkilik qavs
\?	0x3F	? (question mark)	Savol Belgisi
\000	000	Любой (octal number)	Simvol sakkizlik kodi
\xhh	0xhh	Любой (hex number)	Simvol o'n oltilik kodi

Satrlı konstanta.

Satrlı konstantalar C++ tili konstantalariga kirmaydi, balki leksemalari alohida tipi hisoblanadi. Shuning uchun adabiyotda satrlı konstantalar satrlı leksemalar deb ham ataladi..

Satrlı konstanta bu ikkilik qavslarga olingan ixtiyoriy simvollar ketma ketligidir. Misol uchun "Men satrlı konstantaman".

Satrlar orasiga eskeyp simvollar ham kirishi mumkin. Bu simvollar oldiga \ belgisi quyiladi. Misol uchun :

"\n Bu satr \n uch katorga \n zhoyjlashadi".

Satr simvolları hotirada ketma-ket joylashtiriladi va har bir satrlı konstanta ohiriga avtomatik ravishda kompilyator tomonidan '\0' simvoli qo'shiladi. Shunday satrning hotiradagi hazhmi simvollar sonı+1 baytga tengdir.

Ketma-ket kelgan va bushlik, tabulyatsiya yoki satr ohiri belgisi bilan ajratilgan satrlar kompilyatsiya davrida bitta satrga aylantiriladi. Misol uchun:

"Salom" "Toshkent "

satrlari bitta satr deb qaraladi.

"Salom Toshkent"

Bu qoidaga bir necha qatorga yozilgan satrlar ham buysinadi. Misol uchun :

"O'zbekistonga "

"bahor "

"keldi"

qatorlari bitta qatorga mos:

"O'zbekistonga bahor keldi"

Agar satrda '\ ' belgisi uchrasa va bu belgidan so'ng to '\n' satr ohiri belgisigacha bushlik belgisi kelsa bu bushlik belgilari '\ ' va '\n' belgisi bilan birga satrdan uchiriladi. Satrning uzi keyingi satrda kelgan satr bilan qo'shiladi.

"Ozbekistonga \

" bahor\

" keldi"

qatorlari bitta qatorga mos:

"Uzbekistonga baxor keldi"

Sanovchi konstanta.

Sanovchi konstantalar **enum** hizmatchi so'zi yordamida kiritilib, **int** tipidagi sonlarga qulay suzlarni mos quyish uchun ishlatiladi.

Misol uchun:

enum{one=1,two=2,three=3};

Agar son qiymatlari ko'rsatilmagan bulsa eng chapki so'zga 0 qiymati berilib qolganlariga tartib buyicha usuvchi sonlar mos quyiladi:

Enum{zero,one,two};

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, two=2;

Konstantalar aralash ko'rinishda kiritilishi ham mumkin:

Enum(zero,one,for=4,five,seeks};

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, for=4,five=5,seeks=6;

Yana bir misol:

Enum BOOLEAN {NO, YES};

Konstantalar qiymatlari:

NO=0, YES=1;

Nomlangan konstantalar.

CI ++ tilida o'zgaruvchilardan tashqari nomlangan konstantalar kiritilishi mumkin. Bu konstantalar qiymatlarini dasturda o'zgartirish mumkin emas. Konstantalar nomlari dasturchi tomonidan kiritilgan va hizmatchi so'zlardan farqli bo'lgan identifikatorlar bulishi mumkin. Odatda nom sifatida katta lotin harflari va ostiga chizish belgilari kombinaciyasidan iborat identifikatorlar ishlatiladi. Nomlangan konstantalar quyidagi shaklda kiritiladi:

Const tip konstanta_nomi=konstanta_kiyjmati.

Misol uchun:

Const double EULER=2.718282;

Const long M=999999999;

Const R=765;

Ohirgi misolda konstanta tipi kursatilmagan, bu konstanta **int** tipiga tegishli deb hisoblanadi.

Nul ko'rsatkich.

NULL- ko'rsatkich yagona arifmetik bulmagan konstantadir. Konkret realizatsiyalarda null ko'rsatkich 0 eki 0L eki nomlangan konstanta NULL orqali tasvirlanishi mumkin. Shuni aytish lozimki bu konstanta qiymati 0 bo'lishi eki '0' simvoli kodiga mos kelishi shart ehmas.

Quyidagi jadvalda konstantalar chegaralari va mos tiplari ko'rsatilgan:

Ma'lumotlar turi	Hajm, bit	Qiymatlar chegarasi	Tip vazifasi
Unsigned char	8	0...255	Kichik butun sonlar va simvollar kodlari
Char	8	-128...127	Kichik butun sonlar va ASII kodlar
Enum	16	-32768...32767	Butun sonlar tartiblangan katori
Unsigned int	16	0...65535	Katta butun sonlar
Short int	16	-32768...32767	Kichik butun sonlar, tsikllarni boshqarish
Int	16	-32768...32767	Kichik butun sonlar, tsikllarni boshqarish
Unsigned long	32	0...4294967295	Astronomik masofalar
Long	32	-147483648... ...2147483647	Katta sonlar
Float	32	3.4E-32...3.4E+38	Ilmiy hisoblar (7 raqam)
Double	64	1.7E- 308...1.7E+308	Ilmiy hisoblar(15 raqam)
Long double	80	3.4E-4932... 1.1E+4932	Moliyaviy hisobalr (19 raqam)

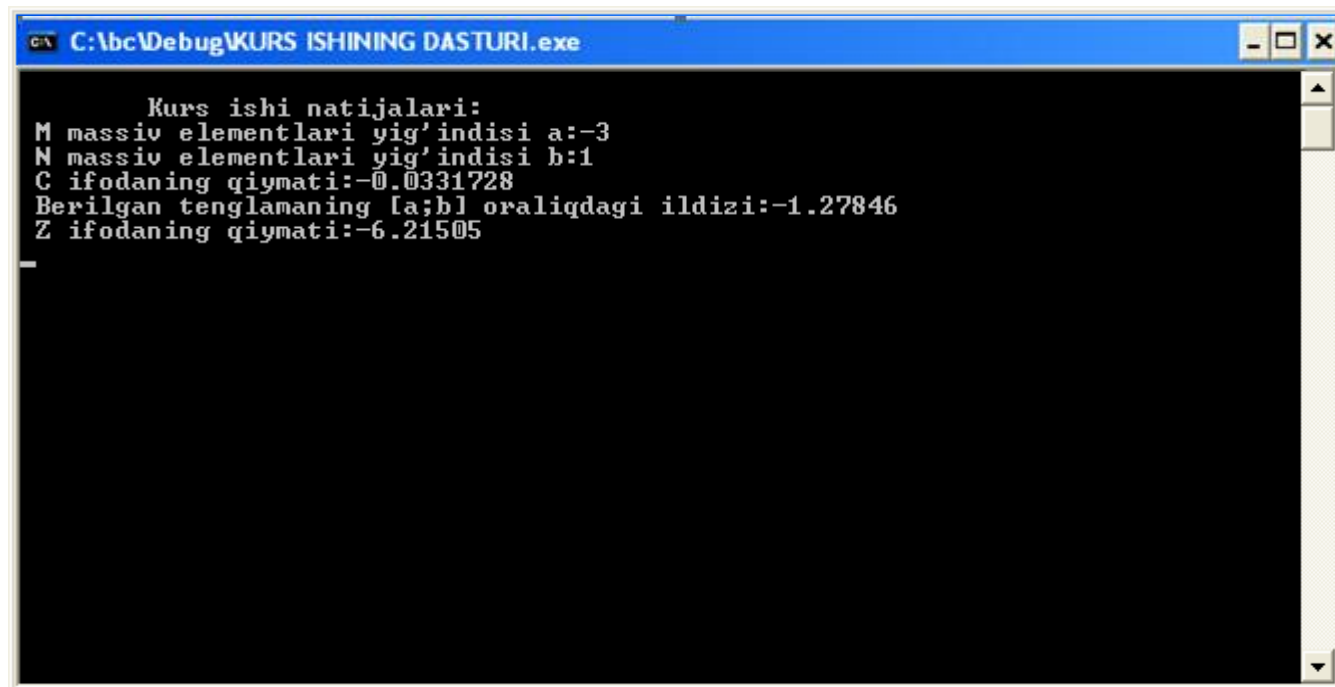
Dasturi.

```
#include<iostream.h>
#include<math.h>
#include<conio.h>
void main()
{
    int i;
    float x,a,b,A,B,c,q=0.125, y=32.56,z,
          M[10]={0,1,2,-3,4,-5,6,-7,8,-9},
          N[10]={0,1,-2,3,4,-5,-6,7,8,-9};
    clrscr();
    a=0;
    b=0;
    for(i=0;i<10;i++)
    {
        a=a+M[i];
        b=b+N[i];
    }
    c=(log(fabs(a))-tan(y))/(q*fabs(a-2)+y);
    A=a;
    B=b;
    cout<<"\n\tKurs ishi natijalari:"<<endl;
    cout<<" M massiv elementlari yig'indisi a:"<<a<<endl;
    cout<<" N massiv elementlari yig'indisi b:"<<b<<endl;
    cout<<" C ifodaning qiymati:"<<c<<endl;
    if(a>b)cout<<" Mavjud bo'lmagan [a;b] oraliq hosil bo'ldi."<<endl;
    else
    {
```

}

Dasturning natijasi.

Yuqoridagi dasturga asosan quyidagi yechimga erishamiz(6-rasm). Budastur yechimlari ekranga nisbatan ajralib turishi uchun chiqarish operatori orqali chiqarilayotgan matn oldidan probel orqali bir katak joy tashlandi va “Kurs ishining natijalari:” matni alohida ajralib turishi uchun “\t”-tabulyatsiyadan foydalanildi.



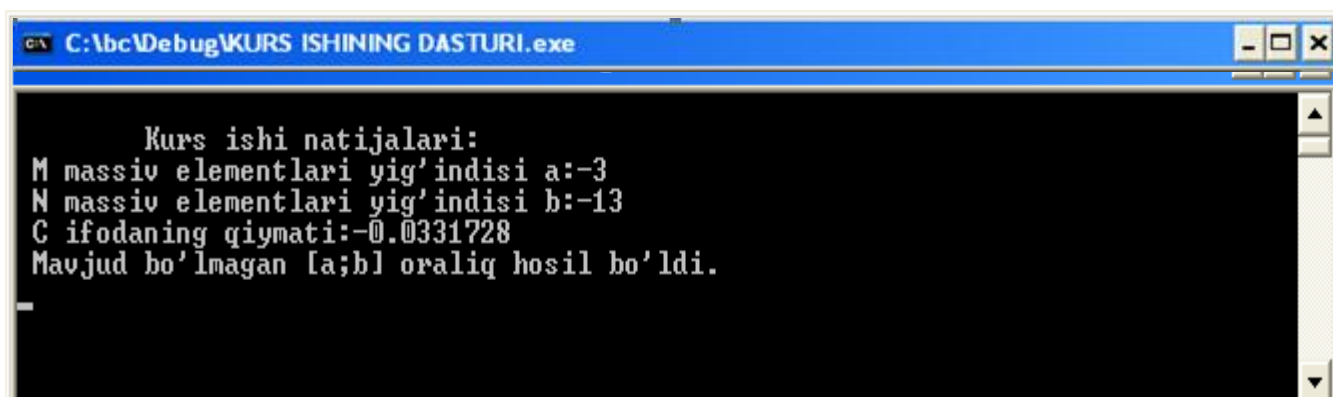
```
C:\abc\Debug\KURS ISHINING DASTURI.exe

Kurs ishi natijalari:
M massiv elementlari yig'indisi a:-3
N massiv elementlari yig'indisi b:1
C ifodaning qiymati:-0.0331728
Berilgan tenglamaning [a;b] oraliqdagi ildizi:-1.27846
Z ifodaning qiymati:-6.21505
```

6-rasm

[a;b] oraliqni ifodalashda $a < b$ tengsizlik o'rinli bo'lishi kerak. Dasturdagi M va N massivlar elementlari yig'indisi a va b lar uchun $a > b$ holat ham kuzatilishi mumkin. Bu holda ham dastur ishlaydi ammo xato yechim hosil bo'ladi. Buni oldini olish uchun dasturda ildizni topishdan oldin $a > b$ shart qo'yildi va bu shart bajarilsa, dasturni davom ettirishga ruxsat etildi. Agar bu shart bajarilmasa, topish mumkin bo'lgan boshqa parametrlar (M massiv elementlari yig'indisi-a, N massiv elementlari yig'indisi-b va c ifodaning qiymati) natijasini ekranga chiqorishga ruxsat etildi. Bu esa oraliq xoto ekanligini to'gridan-to'gri ekrandan ko'rish imkonini beradi.

Bu holda dasturni tekshirib ko'ramiz. Bizga berilgan $M[10] = \{0, 1, 2, -3, 4, -5, 6, -7, 8, -9\}$, $N[10] = \{0, 1, -2, 3, 4, -5, -6, 7, 8, -9\}$; massilar N massivdagi 7 ni -7 ga almashtiramiz. Bu holda M massiv elementlari yig'indisi o'zgarmay (-3) qoladi va N massiv elementlari yig'indisi 14 ga kamayadi ya'ni $1-14 = -13$. Dastur ishga tushirilgandan so'ng quyidagi yechim hosil bo'ladi(7-rasm):



```
C:\abc\Debug\KURS ISHINING DASTURI.exe

Kurs ishi natijalari:
M massiv elementlari yig'indisi a:-3
N massiv elementlari yig'indisi b:-13
C ifodaning qiymati:-0.0331728
Mavjud bo'lmagan [a;b] oraliq hosil bo'ldi.
```


Foydalanilgan adabiyotlar ro'yxati

1. T.X.Xolmatov, N.I.Taylaqov, U.A.Nazarov "Informatika va hisoblash texnikasi" Toshkent 2001-yil.
2. F.Rajabov, S.Masharipov "Oliy matematika" Toshkent "TURON-IQBOL" 2007-yil.
3. Y.U.Soatov "Oliy matematika" Toshkent "O'qituvchi" 1994-yil.
4. R.Karimov va boshqalar "Dasturlash" Toshkent 2003-yil.
5. B.A.Abdalimov "Oliy matematika" Toshkent "O'qituvchi" 1994-yil.