

**O'ZBEKISTON RESPUBLIKASI AXBOROT
TEKNOLOGIYALARI VA KOMMUNIKATSIYALARNI
RIVOJLANTIRISH VAZIRLIGI**

**TOSHKENT AXBOROT TEKNOLOGIYALARI UNIVERSITETI
SAMARQAND FILIALI**

**“KOMPYUTER INJINIRINGI” FAKULTETI
«DASTURIY INJINIRING» KAFEDRASI**

I.M.Boynazarov, T.R.Sokiyev, S.A.Nuriyev

**«C++da dasturlash»
fanidan**

laboratoriya mashg'ulotlari uchun uslubiy ko'rsatma

Samarqand – 2015

Mundarija

Laboratoriya ishi №1	3
Mavzu: Sanoq sistemalari.	3
Laboratoriya ishi №2	10
Mavzu: Algoritmash asoslari.	10
Laboratoriya ishi №3	22
Mavzu: Chiziqli dasturlash.....	22
Laboratoriya ishi №4	30
Mavzu: Tarmoqlanuvchi dasturlash.....	30
Laboratoriya ishi №5	37
Mavzu: Takrorlanuvchi dasturlash.....	37
Laboratoriya ishi №6	57
Mavzu: Bir o'lchamli massivlar.....	57
Laboratoriya ishi №7	65
Mavzu: Ikki o'lchamli massivlar.	65
Laboratoriya ishi №8.....	69
Mavzu: Ko'rsatkichlar.....	69

Laboratoriya ishi №1
Mavzu: Sanoq sistemalari.

I. Variantlar.

Quyidagi variantlarda berilgan sanoq sistemasidagi sonlarni boshqa sanoq sistemalariga o'tkazing.

№	Variantlar
1	$1001001,01_2$; $257,75_8$; 1326_{10} ; $F1A,25_{16}$
2	100111011_2 ; $1524,5_8$; $195,75_{10}$; $DE,A1_{16}$
3	$111111101,11_2$; 7767_8 ; $9856,16_{10}$; $CB,A9_{16}$
4	$11101101,011_2$; $37,62_8$; $1250,61_{10}$; $C91A_{16}$
5	1001110111_2 ; $55,05_8$; $11,01_{10}$; $F1,A_{16}$
6	$1010011001,001_2$; 150_8 ; $98,021_{10}$; DD,F_{16}
7	$100101101,0001_2$; $741,025_8$; 1111_{10} ; $FE,1A_{16}$
8	$10000100001,11_2$; $10010,1_8$; $1010,101_{10}$; 101_{16}
9	1010100001_2 ; $154,2_8$; $123,9_{10}$; $AA,(A)_{16}$
10	$101011,(11)_2$; 4503_8 ; $5649,1_{10}$; $BA,2_{16}$
11	$11001,1000011_2$; $60,1(2)_8$; 1156_{10} ; $99,F_{16}$
12	$10101,101_2$; $205,17_8$; $56,1(9)_{10}$; $AB0C_{16}$
13	1101011_2 ; $123,123_8$; $2563,6_{10}$; $B,A9F_{16}$
14	$1001,01001_2$; 7071_8 ; $98501,1(4)_{10}$; $DE,1F_{16}$
15	$1001,0101_2$; $251,7_8$; 9562_{10} ; $F01,A1_{16}$
16	$10101,001_2$; $121,5_8$; $1751,01_{10}$; $CB9_{16}$
17	1010100011_2 ; 777_8 ; $139,91_{10}$; $FD,1(E)_{16}$
18	$10101,00101_2$; $1010,7(3)_8$; 9801_{10} ; $F1,F1_{16}$
19	$1000101,10011_2$; $701,12_8$; $120,19_{10}$; 999_{16}
20	111101_2 ; $1010,11_8$; $2515,9_{10}$; CD,F_{16}

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

Quyida berilgan $1001,11_2$; $75,01_8$; $95,6_{10}$; $F1,0A_{16}$ sonlarni boshqa (2lik, 8lik, 10lik, 16lik) sanoq sistemalariga o'tkazing.

Ishdan maqsad.

Bir sanoq sistemasida berilgan sonni boshqa sanoq sistemasiga o'tkazish usullarini o'rganish va berilgan misolni ishlash.

Nazariy qism. Ikkilik, sakkizlik, o'nlik, o'n oltilik va boshqa sanoq sistemalari

Odatda biz 10lik sanoq sistemasidan foydalanamiz. 10lik sanoq sistemi 10 ta raqamdan tashkil topgan – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Kompyuterda bajariladigan amallar 2lik sanoq sistemasida bo'lib ular quyidagilar – 0, 1.

Raqaamlari soni n tadan tashkil topgan sanoq sistemi n lik sanoq sistemi deyiladi. Agar raqaamlari - 0, 1, 2, 3, 4 bo'lgan sanoq sistemi 5lik sanoq sistemi deyiladi. Lekin sanoq sistemalarini o'rganishda odatda 2lik va 10lik sanoq sistemalariga qo'shib 8lik va 16lik sanoq sistemalarini ham o'rganamiz. Chunki, o'rganuvchilar uchun 2lik sanoq sistemasidan ushbu sanoq sistemalariga o'tkazishni o'rganish qulay. 8lik sanoq sistemi quyidagi 8 raqamdan tashkil topgan – 0, 1, 2, 3, 4, 5, 6, 7. 16lik sanoq sistemi quyidagi 16 ta raqamdan tashkil topgan – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Ixtiyoriy m -lik $a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-k} \dots$ sanoq sistemasidagi

sonni 10lik sanoq sistemasiga quyidagi formula yordamida o'tkaziladi:

$$b_{10} = a_n \cdot m^n + a_{n-1} \cdot m^{n-1} + \dots + a_1 \cdot m^1 + a_0 \cdot m^0 + a_{-1} \cdot m^{-1} + \dots a_{-k} \cdot m^{-k} + \dots$$

1 – misol: Ikkilik sanoq sistemasida berilgan $110,101_2$ sonni o'nlik sanoq sistemasiga o'tkazing.

$$b_{10} = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$b_{10} = 4 + 2 + \frac{1}{2} + \frac{1}{8} = 6,625_{10}$$

2 – misol: Yettilik sanoq sistemasida berilgan $65,05_7$ sonni o'nlik sanoq sistemasiga o'tkazing.

$$b_{10} = 6 \cdot 7^1 + 5 \cdot 7^0 + 0 \cdot 7^{-1} + 5 \cdot 7^{-2} = 42 + 5 + \frac{5}{49} = 47 \frac{5}{49}$$

Ushbu 10lik son davriy kasr va davri kata bo'lganligi sababli shu holatda qoldi.

3 – misol: O'n oltilik sanoq sistemasida berilgan $A9,0D_{16}$ sonni o'nlik sanoq sistemasiga o'tkazing.

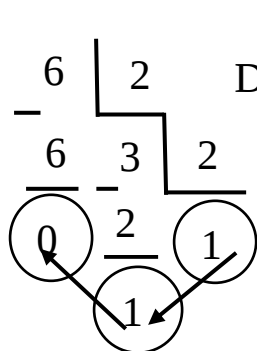
Bu yerda 16lik sanoq sistemasidagi A 10lik sanoq sistemasida 10 ga D 13 ga to'g'ri keladi.

$$b_{10} = 10 \cdot 16^1 + 9 \cdot 16^0 + 0 \cdot 16^{-1} + 13 \cdot 16^{-2} = 160 + 9 + \frac{13}{256}$$

$$b_{10} = 169 \frac{13}{256} = 169,05078125.$$

O'nlik sanoq sistemasida berilgan sonni **ml**ik sanoq sistemasiga o'tkazish uchun berilgan sonning butun qismini **m** ga bo'lamiz va bo'linmani yana **m** ga bo'lamiz. Bu jarayon bo'linma **m** dan kichik bo'lib qolguncha davom ettiriladi va oxirgi bo'linmadan boshlab qoldiqlarni ketma-ket yozib chiqamiz. Qoldiq qismini **m** ga ko'paytiramiz va qoldiq qismini yana **m** ga ko'paytiramiz. Bu jarayon qoldiq qismi nolga teng bo'lguncha yoki davriy takrorlanguncha davom etadi. Quyidagi misollarda aniqlik kiritamiz.

4 – misol: O'nlik sanoq sistemasida berilgan $6,625_{10}$ sonni ikkilik sanoq sistemasiga o'tkazing. Berilgan sonning butun qismi 6 va kasr qismi 0,625 ga teng.



$$6_{10} = 110_2 \text{ va } 0,625_{10} = 0,101_2$$

Demak, $6,625_{10} = 110,101_2$ ga teng.

×	0	625
		2
	1	250
×		2
	0	500
×		2
	1	000

5 – misol: O'nlik sanoq sistemasida berilgan $0,3_{10}$ sonini sakkizlik sanoq sistemasiga o'tkazing.

×	0	3
		8
2		4
×		8
3		2
×		8
1		6
×		8
4		8
×		8
6		4
×		8
3		2
×		

Demak, $0,3_{10} = 0,2(3146)_8$ ga teng. Agarda berilgan son kasr qismini ko'paytirganda nollar hosil bo'lmasa u davriy kasr bo'ladi va uni takrorlanguncha ko'paytirib chiqamiz.

Ikkilik sanoq sistemasida berilgan sonni sakkizlik sanoq va o'n oltilik sistemasiga jadval asosida o'tkazamiz.

2lik	000	001	010	011	100	101	110	111
8lik	0	1	2	3	4	5	6	7

2lik	0000	0001	0010	0011	0100	0101	0110	0111
16lik	0	1	2	3	4	5	6	7
2lik	1000	1001	1010	1011	1100	1101	1110	1111
16lik	8	9	A	B	C	D	E	F

6 – misol: Ikkilik sanoq sistemasida berilgan $110,101_2$ sonni 8lik va 16lik sanoq sistemalariga o'tkazing.

$$110_2 = 6_8; \quad 101_2 = 5_8; \quad \text{Demak: } 110,101_2 = 6,5_8.$$

Ushbu sonni o'n oltilik ga o'tkazish uchun uni quyidagicha yozib olamiz: $0110,1010_2$ chunki, 16likka o'tkazish uchun to'rttadan qilib olishimiz lozim. Demak butun qismining boshiga va kasr qismining

oxiriga yetmagan qismiga kerakli noll qo'shishimiz mumkin. Bu yerda:

$$0110_2 = 6_{16}; \quad 1010_2 = E_{16}; \quad \text{Demak: } 110,101_2 = 6, E_{16}.$$

7 – misol: Sakkizlik sanoq sistemasida berilgan $65,65_8$ sonni o'n oltilik sanoq sistemasiga o'tkazing.

Ushbu 8lik sonni 16likka o'tkazishning ikki usuli mavjud. Birinchisi – 8lik sonni 10likka o'tkazamiz va natijani 16likka o'tkazamiz. Ikkinchisi – 8lik sonni 2likka o'tkazib, natijani 16likka o'tkazamiz. Qulayi ikkinchisi.

$$6_8 = 110_2; \quad 5_8 = 101_2; \quad \text{Demak: } 65,65_8 = 110101,110101_2$$

Endi natijani quyidagicha yozamiz: $00110101,11010100_2$ va

$$0011_2 = 3_{16}; \quad 0101_2 = 5_{16}; \quad 1101_2 = D_{16}; \quad 0100_2 = 4_{16};$$

$$110101,110101_2 = 35, D4_{16}; \quad \text{Natija: } 65,65_8 = 35, D4_{16}.$$

Berilgan masala uchun algoritm.

$$1001,11_2; \quad 75,01_8; \quad 95,6_{10}; \quad F1,0A_{16}.$$

Berilgan $1001,11_2$ sonni sakkizlik va o'n oltilik sanoq sistemalariga o'tkazish uchun jadvaldan foydalanish qulay usul hisoblanadi. O'nlik sanoq sistemasiga o'tkazish uchun

$$b_{10} = a_n \cdot m^n + a_{n-1} \cdot m^{n-1} + \dots + a_1 \cdot m^1 + a_0 \cdot m^0 + a_{-1} \cdot m^{-1} + \dots + a_{-k} \cdot m^{-k} + \dots$$

formulasidan foydalanamiz.

$75,01_8$ sonini ikkilikga jadval asosida o'tkazamiz va natijani yana jadval asosida o'n oltilikka o'tkazamiz. O'nlikka formula yordamida o'tkazamiz.

$95,6_{10}$ sonini 2likka butun qismini bo'lish va kasr qismini ko'paytirish usuli yordamida o'tkazamiz va natijani jadval asosida 8lik va 16likka o'tkazamiz.

$F1,0A_{16}$ sonni 2likka jadval asosida o'tkazamiz va natijani 8likka o'tkazamiz. 10likka formuladan foydalanamiz.

1 – misol: $1001,11_2$ sonini 8likka o'tkazish uchun uchtdan qilib ajratib olamiz butun qismining yetmaganiga boshidan nol qo'shamiz, kasr qismining yetmaganiga oxiridan nol qo'shamiz $001001,110_2$ va jadval asosida 8likka aylantiramiz.

$$1001_2 = 1_8; 001_2 = 1_8; 110_2 = 6_8; \text{ natija: } 1001,11_2 = 11,6_8.$$

16likka ham jadval asosida aylantiramiz, lekin raqamlarni to'rttdan qilib ajratib olamiz $1001,1100_2$.

$$1001_2 = 9_{16}; 1100_2 = C_{16}; 1001,11_2 = 9, C_{16}.$$

10lik sanoq sistemasiga formula yordamida quyidagicha o'tkazamiz:

$$1^3 0^2 0^1 1^0, 1^{-1} 1^{-2}_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 8 + 1 + \frac{1}{2} + \frac{1}{4} = 9,75_{10}; \text{ natija: } 1001,11_2 = 9,75_{10}.$$

2 – misol: $75,01_8$ 8lik sonini ikkilikga jadval asosida o'tkazamiz:

$$7_8 = 111_2; 5_8 = 101_2; 0_8 = 000_2; 1_8 = 001_2;$$

$$\text{Natija: } 75,01_8 = 111101,000001_2$$

2likka o'tkazilgan natijani 16likka o'tkazamiz: $00111101,00000100_2$

$$0011_2 = 3_{16}; 1101_2 = D_{16}; 0000_2 = 0_{16}; 0100_2 = 4_{16};$$

$$\text{Natija: } 75,01_8 = 3D,04_{16}.$$

$75,01_8$ 8lik sonini 10likka formula asosida o'tkazamiz.

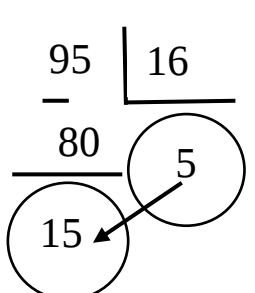
$$7^1 5^0, 0^{-1} 1_8^{-2} = 7 \cdot 8^1 + 5 \cdot 8^0 + 0 \cdot 8^{-1} + 1 \cdot 8^{-2} = 61,015625_{10}$$

$$\text{Natija: } 75,01_8 = 61,015625_{10}.$$

3 – misol: $95,6_{10}$ ni 16likka o'tkazamiz:

$$5_{10} = 5_{16}; 15_{10} = F_{16}; 95_{10} = 5F_{16}$$

$$0,6_{10} = 0,(9)_{16}$$

$$\text{Natija: } 95,6_{10} = 5F,(9)_{16}$$


$$\begin{array}{r|l} \times & 06 \\ \hline & 16 \\ \hline \times & 96 \\ \hline & 16 \\ \hline & 96 \end{array}$$

$5F, (9)_{16}$ natijani ikkilikka o'tkazish qulay:

$$5_{16} = 0101_2; F_{16} = 1111_2; 0, (9)_{16} = 0, (1001)_2;$$

$$\text{Natija: } 95,6_{10} = 1011111, (1001)_2$$

Natijani 8likka o'tkazamiz:

$$001011111, (100110011001)_2 = 137, (4631)_8$$

4 – misol: $F1,0A_{16}$ sonini ikkilikka o'tkazamiz:

$$F_{16} = 1111_2; 1_{16} = 0001_2; 0_{16} = 0000_2; A_{16} = 1010_2$$

$$\text{Natija: } F1,0A_{16} = 11110001,00001010_2$$

Natijani $011110001,00001010_2$ 8likka o'tkazamiz:

$$011_2 = 3_8; 110_2 = 6_8; 001_2 = 1_8; 000_2 = 0_8; 010_2 = 2_8; 100_2 = 4_8$$

$$\text{Natija: } F1,0A_{16} = 361,024_8$$

$F1,0A_{16}$ sonini formula yordamida 10likka o'tkazamiz:

$$F^1 1^0, 0^{-1} A_{16}^{-2} = 15 \cdot 16^1 + 1 \cdot 16^0 + 0 \cdot 16^{-1} + 10 \cdot 16^{-2}$$

$$F1,0A_{16} = 241,0390625_{10}$$

Natijalar listingi.

1. $1001,11_2 = 11,6_8 = 9,75_{10} = 9, C_{16}$
2. $75,01_8 = 111101,000001_2 = 61,015625_{10} = 3D, 04_{16}$
3. $95,6_{10} = 1011111, (1001)_2 = 137, (4631)_8 = 5F, (9)_{16}$
 $F1,0A_{16} = 11110001,00001010_2 = 361,024_8 =$
4. $241,0390625_{10}$

Foydalanilgan adabiyotlar.

1. C va C++ tili, Nazirov Sh. A., Kabulov R. V., Babajanov M. R., Raxmanov Q. S., Toshkent – 2013.
2. C/C++ tilida masalalar va topshiriqlar, Yusupov R. A., Mirzanova Sh., Iskandarova S. N., Samarqand – 2013.
3. <http://acm.tuit.uz>.

Laboratoriya ishi №2
Mavzu: Algoritmash asoslari.

I. Variantlar.

Quyidagi variantlarda berilgan masalalarning algoritmlarini tuzing.

№	Variantlar
1	Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshirish algoritmini tuzing: A va B sonlarning faqat bittasi toq son".
2	Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshirish algoritmini tuzing "A va B sonlarining har ikkalasi ham yoki toq son yoki juft son".
3	Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshirish algoritmini tuzing : A, B, C sonlarning har biri musbat".
4	Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshirish algoritmini tuzing:A, B, C sonlarning hech bo'lmaganda bittasi musbat".
5	Uchta A, B, C butun sonlar berilgan. Jumlani Jumlani rostlikka tekshirish algoritmini tuzing:A, B, C sonlaridan faqat bittasi musbat son".
6	Musbat butun son berilgan. Jumlani rostlikka tekshirish algoritmini tuzing: "Berilgan son ikki xonali juft son".
7	Musbat butun son berilgan. Jumlani rostlikka tekshirish algoritmini tuzing: "Berilgan son uch xonali toq".
8	Jumlani rostlikka tekshirish algoritmini tuzing "Berilgan uchta butun sonlarning hech bo'lmaganda 2 tasi bir biriga teng".
9	Jumlani rostlikka tekshirish algoritmini tuzing: "Berilgan uchta butun sonlarning hech bo'lmaganda bir jufti o'zaro qarama-qarshi".
10	Uch xonali son berilgan. Jumlani rostlikka tekshirish algoritmini tuzing: "Ushbu sonning barcha raqamlari xar xil".
11	n butun soni berilgan ($n > 0$). Quyidagi yig'indini hisoblovchi algoritm tuzilsin. $S = 1 + 1/2 + 1/3 + \dots + 1/n$

12	n butun soni berilgan ($n > 0$). Quyidagi yig'indini hisoblovchi algoritm tuzilsin. $s = n^2 + (n+1)^2 + (n+2)^2 + \dots (2 \cdot n)^2$
13	n butun soni berilgan ($n > 0$). Quyidagi yig'indini hisoblovchi algoritm tuzilsin. $S = 1 - 2 + 3 - \dots$ (n ta qo'shiluvchi, ishoralar almashib keladi.)
14	n butun soni berilgan ($n > 0$). Shu sonning kvadratini quyidagi formula asosida hisoblovchi algoritm tuzilsin. $n^2 = 1 + 3 + 5 + \dots + (2 \cdot n - 1)$ har bir qo'shiluvchidan keyin natijani ekranga chiqarib boring. Natijda ekranda 1 dan n gacha bo'lgan sonlar kvadrati chiqariladi.
15	n butun soni va a haqiqiy soni berilgan ($n > 0$). a ning n - darajasini aniqlovchi algoritm tuzilsin. $a^n = a \cdot a \cdot a \dots a$;
16	n butun soni va a haqiqiy soni berilgan ($n > 0$). Bir sikldan foydalanib a ning 1 dan n gacha bo'lgan barcha darajalarini chiqaruvchi algoritm tuzilsin.
17	n butun soni va a haqiqiy soni berilgan ($n > 0$). Bir sikldan foydalanib quyidagi a ning 1 dan n gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblovchi algoritm tuzilsin. $1 + a + a^2 + a^3 + \dots a^n$
18	n butun soni va a haqiqiy soni berilgan ($n > 0$). Bir sikldan foydalanib quyidagi a ning 1 dan n gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblovchi algoritm tuzilsin. $1 - a + a^2 - a^3 + \dots (-1)^n a^n$
19	n butun soni berilgan ($n > 0$). Quyidagi ko'paytmani hisoblovchi algoritm tuzilsin. $S = 1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$ (n ta ko'payuvchi)
20	n va k butun son berilgan. k sonini n marta chiqarish algoritmini tuzing.

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

Quyida berilgan $a \cdot x^2 + b \cdot x + c = 0$ kvadrat tenglamaning ishlanish algoritmini va uning blok-sxemasini tuzing.

Ishdan maqsad.

Berilgan masala uchun uni ishlanish algoritmini va blok-sxemasini tuzish usullarini o'rganish va berilgan misolni ishlash.

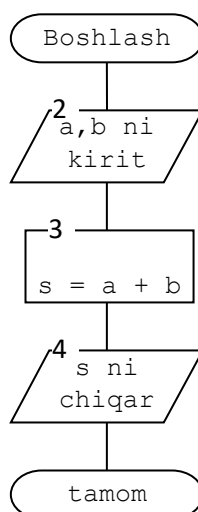
Nazariy qism.

Algoritm – bu berilganlar asosida natijaga olib keluvchi jarayonning aniq tavsifidir.

50-yillardan boshlab dasturchilar algoritmnı tasvirlash uchun blok-sxema deb nomlanuvchi grafik sxemalardan foydalana boshladi.

Masalani yechimini grafik tasvirlash – algoritmnı yozish uchun eng qulay usuldir. Algoritmıları grafik tasvirlashning keng tarqalgan usulları blok-sxemalar va Nassi-Shnayderman strukturagrammalarıdır.

Quyida ikki son yig'indisining hisoblash algoritmi blok-sxeması keltirilgan.



Algoritmning xossalari:

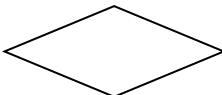
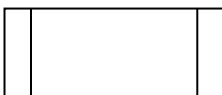
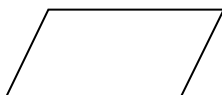
- **bir qiymatlilik** – amallarnı bajarishni tushuntirishning yagona qoidası va bajarilish tartibi mavjudligi;
- **tugallanganlik** – algoritmnı tashkil etuvchi har bir amalning albatta tugallanishi hamda butun algoritm bajarilishining to'liq tugallanishi;
- **natijaviylik** – algoritm bajarilishi natijasida aniq natijaga erishish;

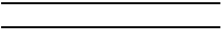
- **umumiylik** – algoritmnining butun bir sinf masalalarini yechishga qo'llash mumkinligi (bunda kiritilayotgan qiymatlar berilgan oraliqlarda o'zgarganda algoritm ishining to'g'ri bajarilishi nazarda tutiladi);
- **to'g'rilik** – qo'yilgan masalani yechishda algoritmnining to'g'ri natijalar berish imkoniyati.

Blok-sxemalar ma'lum qoidalar asosida yaratiladi va amallarning bajarilish tartibini ko'rsatuvchi, o'zaro chiziqlar (strelkalar) bilan bog'langan geometrik shakllar (belgilar bloki) dan iborat.

Belgilar bloki standartlashtirilgan va bajarilayotgan amal turiga qarab farqlanadi.

Quyida eng ko'p ishlatilayotgan belgilar bloki keltirilgan:

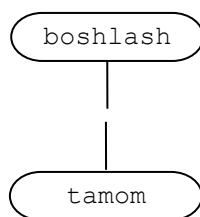
Nomlanishi	Belgi o'lchami	Vazifasi
Jarayon		Natijada qiymati o'zgaradigan, bir guruh amalni tasvirllovchi shakl yoki ma'lumotlar joylashishi
Qaror		Biror qiymat asosida algoritm yoki dasturning bajarilish yo'nalishni tanlovchi shart
Modifikasiyalash		Dasturni o'zgartiruvchi amallar bajarilishining o'zgarishini ko'rsatuvchi buyruq yoki buyruqlar guruhi
Oldindan aniqlangan jarayon		Oldindan yaratilgan va alohida tavsiflangan algoritm yoki dasturdan foydalanish
Kiritish-chiqarish		Ma'lumotlarni qayta ishlash uchun yaroqli holga keltirish (kiritish) yoki qayta ishlangan

		natijalarni tasvirlash (chiqarish)
Oqim chizig'i		Belgilarni qayta ishlash ketma-ketligini ko'rsatish
Parallel amallar		Bir yoki undan ortiq bir vaqtda bajariluvchi amallarni boshlanishi va tugallanishi
Ulagich (bog'lagich)		Belgilarni bog'lovchi oqimlar chizig'i orasida uzilish mavjudligini ko'rsatish
Boshlash-to'xtash		Dasturlar bajarilishi yoki ma'lumot-larni qayta ishlash jarayonidagi uzilish to'xtash yoki boshlanish
Izoh		Sxema elementlari va tushintirishlar o'rtasidagi aloqa
Sahifalararo ulagich (bog'lagich)		Turli varaqlarda joylashgan, bo'laklarga ajratilgan sxemalar o'rtasidagi bog'liqlik mavjudligini ko'rsatish

Oqim chizig'i varaq shaklining tashqi chetidan parallel holda o'tkaziladi. Ularning kesishishiga ruxsat beriladi yoki 90^0 burchak ostida qaytarishga yo'l qo'yiladi. Yuqoridan pastga va chapdan o'ngga yo'naltirilgan oqim chiziqlari asosiy deb hisoblanadi: agar asosiy yo'nalishdagi oqim chiziqlari sinishlarsiz bo'lsa, unda ularni strelkasiz chizish mumkin. Qolgan hollarda oqim chizig'ini strelkalar orqali tasvirlash shart.

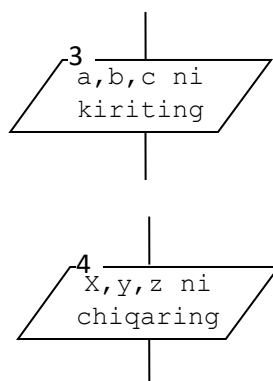
Blok-sxemalar asosiy grafik elementlarining qo'llanilishini batafsil ko'rib chiqamiz.

Sxemalarda algoritmning boshlanishi va oxiri hamda yordamchi algoritmlarning kirish va chiqishlari mos ravishda algoritmning boshi va oxirini blokli belgilar bilan ifodalanadi.

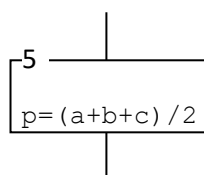


Bu blokli belgilar boshqa belgilardan farqli ravishda bitta kirish va bitta chiqishga ega bo'lib, ma'lumotlarni qayta ishlash yo'lining boshi va oxirini bildiruvchilar sifatida ishlatiladi. Har bir sxema ana shu belgilar bilan boshlanishi va tugashi lozim.

Keyingi parallelogram ko'rinishidagi belgilar ma'lumotlarni kiritish va chiqarish amallari uchun foydaniladi.

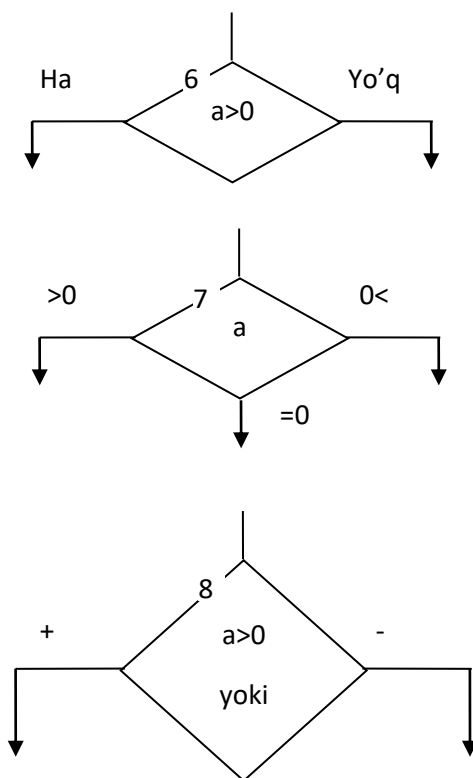


Hisoblash jarayonlarni ifodalovchi blok, amallarni (yoki guruh amallarni), qiymati o'zgaruvchanlarni, tasvirlash shakllarini yoki ma'lumotlar joylashini ifodalashda qo'llaniladi.



Bu blokda tavsiflanuvchi amallar yechimini bildiruvchi matematik formula, ifoda va ixtiyoriy tildagi tushuntirishlarda yozilishi mumkin.

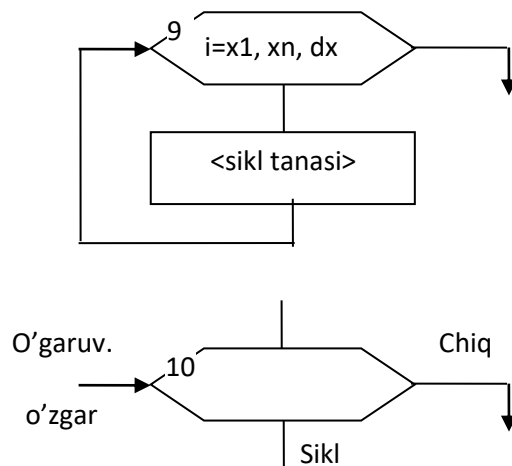
Yechimni bildiruvchi, mantiqiy blok belgisi, biror-bir shart asosida algoritm bajarilishi yo'nalishini tanlashda ishlatiladi.



Blokda shart, savol yoki algoritm keyingi yo'nalishni aniqlaydigan yechim ko'rsatiladi. Shart oddiy (6 blok) yoki tarkibli (8 blok) bo'lishi mumkin. Masalalarni yechish jarayonida shart bloklarida yuz berishi mumkin bo'lgan barcha holatlar hisobga olinishi kerak.

Bitta chiqishga ega bo'lgan blokli belgilardan farqli ravishda shartli bloklardan ikki yoki uchta informatsiyali oqim yo'nalishlari chiqishi mumkin. Blokdan chiqish chiziqlari algoritm yo'nalishini amalga oshiruvchi shartlar (masalan, "ha" yoki "yo'q", <0 , $=0$ yoki >0 , $+$ yoki $-$ va h.k.) bilan ta'minlangan bo'lishi kerak.

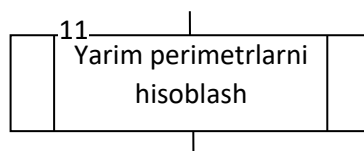
Modifikatsiya blok belgilari siklik hisoblashlarning boshlanishini (sikl sarlavhasi) bildirib, siklni boshqarish uchun ishlatiladi.



Blok ichida sikl o'zgaruvchilari va uning o'zgarishni aniqlash qonuniyati xususiyatlari ko'rsatiladi, masalan, $i=x_1, x_n, dx$, bu yerda i – o'zgaruvchisi, x_1 va x_n – sikl parametrining boshlang'ich va oxirgi qiymatlari, dx – o'zgarish qadami (sikl parametri x_1 dan x_n gacha dx qadam bilan o'zgaradi). Agar qadam 1 ga teng bo'lsa, unda dx ni ko'rsatmaslik ham mumkin.

Modifikasiya blokida kirish chizig'idan tashqari bitta chiqish (rasmda “Chiqish” orqali ko'rsatilgan) ham mavjud bo'lib, bu chiziq sikl o'zgarishini hisoblash jarayonining boshiga qayta ishlash uchun uzatadi (“Sikl”) va sikl parametrini o'zgarish uchun boshiga qaytarish (“O'zgaruv. o'zgar.”).

Qismaniy dastur hisoblashlariga murojaat qilish uchun (standart yoki foydalanuvchi tomonidan yaratilgan) sxemada oldindan yaratilgan jarayon blok belgisi ishlatiladi.



Bu blok qismaniy dastur (yordamchi algoritm) algoritmi o'rnida ishlatilib, ma'lumotlar oqimining qismaniy dasturga uzatilishini bildiradi. Qismaniy dasturdagi hisoblash jarayoni tugallanganidan so'ng, hisoblash natijalari asosiy algoritmgaga qaytariladi, hisoblash jarayoni qismaniy dasturga murojaat blokidan keyingi jarayonidan boshlab tiklanadi.

Oldindan tayyorlangan jarayon bloki alohida modul sifatida hosil qilingan yordamchi algoritmlarni tashkillashtirishda yoki kutubxona qismaniy dasturlariga murojaatda ishlatiladi.

Ma'lumotlar oqimi yo'llari uzunligini va kesishishlar sonini kamaytirish uchun ularni uzish va uzilgan joylarga ulagichlarni o'rnatish mumkin. Agar bloklar o'rtasidagi chiziqlar uzilishi bitta sahifada yuz bersa, unda ulagich sifatida maxsus belgilardan foydalaniladi.



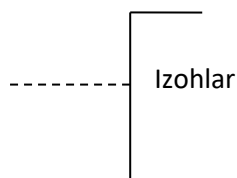
Blok belgilari ichiga hisoblash jarayoni uzatiladigan yoki qayerdan kelganligini bildiruvchi blok raqamlari yoziladi. Xudi shunday yuqori ulagich hisoblash jarayoni 15 – blok kirishga uzatilayotganligini, quyi ulagich esa hisoblash jarayoni 10 – blokdan chiqayotganligini bildiradi.

Agar ulagich chiziqlari turli xil sahifalarda joylashgan bloklarni ulash jarayonini ko'rsatadigan bo'lsa, unda sahifalararo ulagich belgilaridan foydalaniladi, hamda bu blok ulagichlari ichiga nafaqat blok raqamlari, balki sahifa raqamlari ham yoziladi.



Yuqoridagi sahifalararo ulagich tasvirlari hisoblash jarayoni 10-sahifada joylashgan 23-blok kirishiga uzatilayotganligini bildiradi. Quyi sahifalararo ulagich esa, ma'lumotlar 6-sahifada joylashgan 12-blok chiqishidan uzatilayotganligini (ya'ni oldin sahifa raqami, uning ostidan esa, uzatilishi lozim bo'lgan yoki boshqarishni qabul qiluvchi blok raqami ko'rsatiladi) bildiradi.

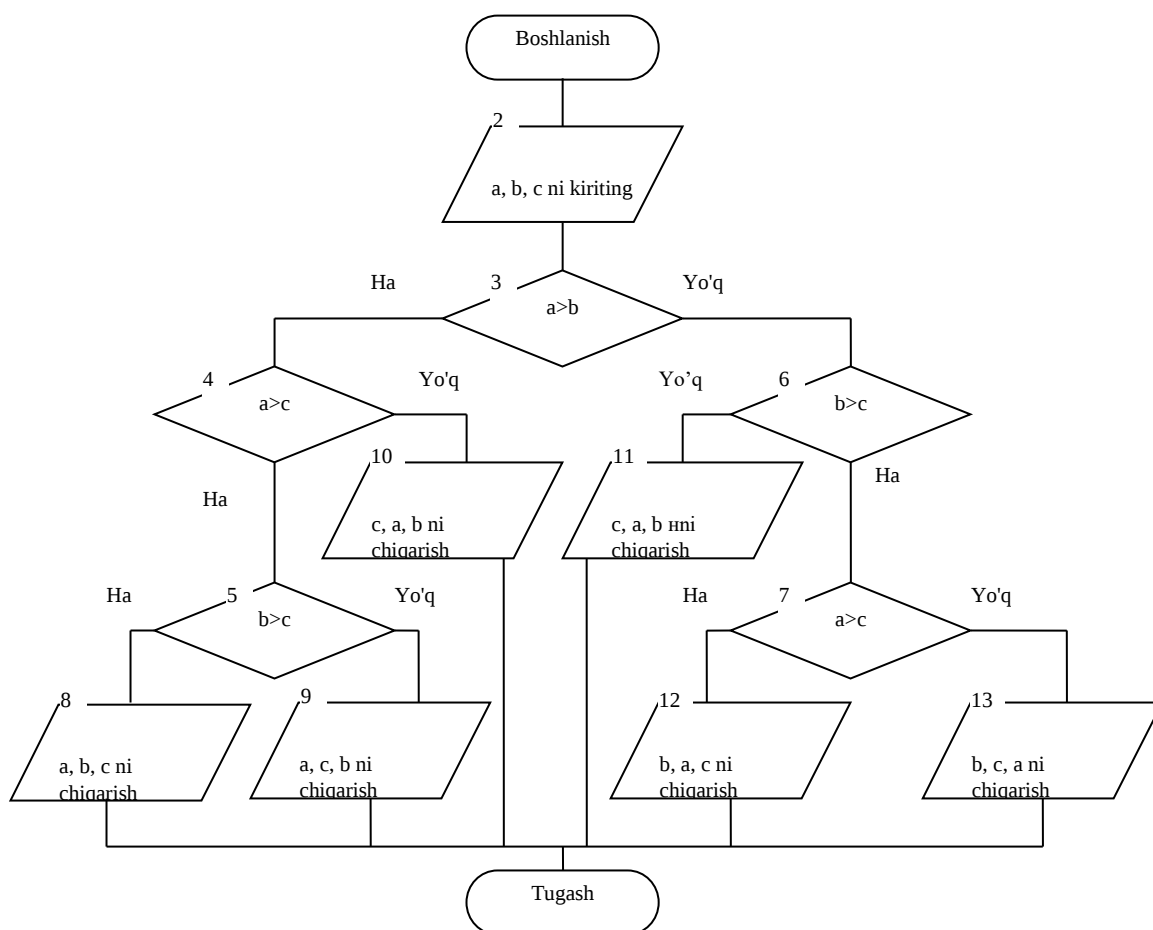
Alohida bloklarning yoki bloklar guruhining funksiyalarini, xususiyatlarini, alohida olingan elementlar uchun qabul qilingan ruxsatlar va tayinlanishlarni tavsifini, o'zgaruvchilarning belgilanishi uchun algoritm sxemalarida izohlar qo'shilishi mumkin.



Sxema algoritmlarni tasvirlash uchun eng qulay va oddiy usuldir. Unda amallarning bajarilish ketma-ketligi, ma'lumotlar oqimi va ularning yurish yo'llari aniq kuzatiladi. Bloklarga nisbatan chiziqlar kiradigan va chiqadigan bo'lishi mumkin. Barcha bloklar uchun kirish chiziqlari cheklanmaydi – ular bitta, ikkita, uchta va ko'p bo'lishi mumkin. Ko'pchilik bloklar uchun chiqish chiziqlari faqat bitta bo'lishi mumkin (shartni tekshirish bloki bundan mustasno).

Sxemadagi bloklar, tutashtirgichlardan tashqari, ularning bajaradigan ishlarini keyinchalik oson tavsiflash, izohlarni tashkillashtirish va tutashtirgichlardan foydalanish uchun raqamlanishi mumkin. Raqamlar, oldingi rasmlarda ko'rsatilganidek, grafik belgilarning yuqori qismidagi shakl uzilishlariga qo'yiladi. Boshlanish va to'xtash bloklariga raqamlar qo'yilmaydi, ammo boshqa bloklarni raqamlashda albatta hisobga olinadi.

Blok ichida va uning yonlarida uning bajaradigan funksiyalarini tavsiflovchi yozuvlar ko'rsatiladi. Bunday yozuvlar yaratuvchiga qulay bo'lgan istalgan shaklda ko'rsatilishi mumkin. Ular sezilarli darajadagi cheklashlarga (tiliga, belgilariga, belgilashlariga va boshqalariga) ega emas, ammo algoritmdan foydalanuvchilarning barchasiga tushunarli bo'lishi kerak. Yagona cheklanish yozuvlar ketma-ketligiga qo'yiladi – ular ma'lumotlar oqimining yo'nalishidan qat'iy nazar o'ngdan chapga qarab o'qilishi va yuqoridan pastga qarab yozilishi kerak.



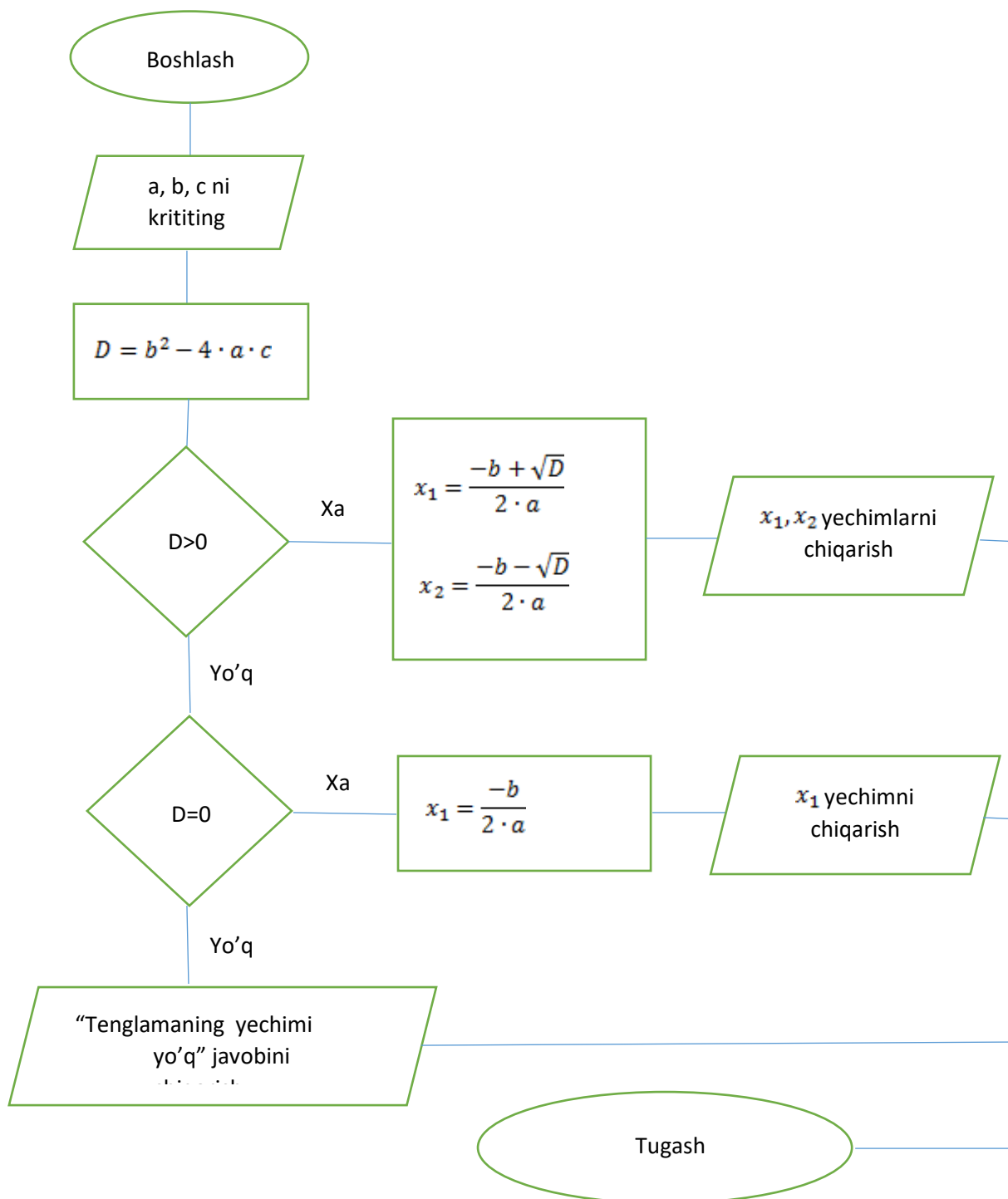
Qiymatlarining kamayishiga nisbatan uchta sonni ekranga chop etuvchi blok-sxema

Berilgan masala uchun algoritm.

Ushbu $a \cdot x^2 + b \cdot x + c = 0$ kvadrat tenglama quyidagicha yechiladi:

1. a, b, c – koeffitsientlar kiritiladi;
2. D – diskreminant, $D = b^2 - 4 \cdot a \cdot c$ formula yordamida hisoblanadi;
3. $D > 0$ bo'lsa, $x_1 = \frac{-b + \sqrt{D}}{2 \cdot a}$; $x_2 = \frac{-b - \sqrt{D}}{2 \cdot a}$; yechimlar aniqlanadi, aks holda $D = 0$ bo'lsa, $x = \frac{-b}{2 \cdot a}$; yechimi aniqlanadi, aks holda $D < 0$ bo'lsa, berilgan tenglamaning yechimi yo'qligi aniqlanadi.

Natijalar listingi.



Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. C/C++ tilida masalalar va topshiriqlar, Yusupov R. A., Mirzanova Sh, Iskandarova S. N., Samarqand – 2013.
3. <http://algo.urgench-tuit.uz>.

Laboratoriya ishi №3
Mavzu: Chiziqli dasturlash.

I. Variantlar.

Quyidagi variantlarda berilgan chiziqli masalalar dasturini tuzing.

№	Variantlar	№	Kirish	Chiqish
1	$A = \frac{\sqrt{ x-1 - \sqrt[3]{ y }}}{1 + \frac{x^2}{2} + \ln \frac{y^2}{4}};$ $B = x \cdot (\arctg(A) + e^{-(x-1)});$	1	x=3 y=-1.4	A=0.20 B=0.99
		2	x=-1.2 y=2	A=0.56 B=-11.45
2	$A = \frac{3 + e^{y-x}}{1 + x^2(y - \cos(x-3))};$ $B = 1 + \sqrt{ y-x } + \frac{(y-A)^2}{2} \cdot \sqrt[3]{(y-x)};$	1	x=2 y=3.1	A=0.534 B=5.447
		2	x=3 y=4.5	A=0.230 B=12.659
3	$A = (1+y) \cdot \frac{x + \frac{y}{x^2+4}}{e^{-(x+2)} + \sqrt{x^2+4}};$ $B = 1 + tg^2\left(\frac{A \cdot \pi}{2}\right) \cdot \sqrt[5]{x^2+4};$	1	x=1 y=-2.1	A=-0.28 B=1.30
		2	x=1 y=2.6	A=2.39 B=1.70
4	$A = y + \frac{x}{x^2 + \left \frac{x^2}{e^y + x^3/3} \right };$ $B = \frac{1 + \cos(A-2)}{x^4 + \sqrt[3]{\sin^2(x-y)}};$	1	x=2 y=1.2	A=1.68 B=0.115
		2	x=4 y=6.2	A=6.450 B=0.003
5	$A = \frac{2 \cdot \cos(x - \pi/6)}{\sin(2 \cdot x) + \sin^2(x-y)};$ $B = \cos^2\left(\arctg \frac{1}{A}\right) + \sqrt[3]{e^{x+y}};$	1	x=1.6 y=-6.2	A=1.011 B=0.721
		2	x=2 y=5.2	A=-0.250 B=11.082
6	$A = \frac{1 + \sin^2(x+y)}{2 + \left x - \frac{2 \cdot x}{(1+x^2 \cdot y^2)} \right };$ $B = \cos^2\left(1 + \frac{A^2 \cdot (x+y)}{e^x \cdot x^2 \cdot y^2}\right);$	1	x=4 y=3.4	A=1.891 B=0.290
		2	x=3 y=6.3	A=1.646 B=0.289
7	$A = \ln \left (y - \sqrt{x}) \cdot \left(x + \frac{e^y}{\cos^2 x + \frac{y^2}{4}} \right) \right ;$ $B = (x + tg \frac{2 \cdot \pi}{A}) \cdot (5 \cdot 10^{-6} + \sqrt[4]{A \cdot y});$	1	x=4.5 y=3.4	A=2.934 B=5.227
		2	x=2.6 y=3.4	A=2.967 B=1.708

8	$A = \left \frac{\sin^3(\pi - x)}{\sqrt{(x - y)^2 + e^{-x^2}}} \right ;$ $B = \sqrt[3]{\operatorname{tg} \frac{\pi}{A} \cdot \ln(2 \cdot 10^3 - \cos^2(x - y))};$	1	x=5 y=-3	A=0.11 B=4.68
		2	x=5 y=1.2	A=0.23 B=8.64
9	$A = e^{-(x+1)} \cdot \sqrt{\frac{17 \cdot 10^6 + \sin \pi x}{(1.1 - \cos(y^2 + 18))^2}};$ $B = x \cdot \left(\ln \left \frac{A}{x^2 + 1} \right + \operatorname{ctg} \left(\frac{y^2 + 18}{x} \right) \right);$	1	x=0.2 y=4	A=637.316 B=1.825
		2	x=0.5 y=2.6	A=5454.451 B=3.649
10	$A = \frac{e^{x+y} + \sqrt[3]{x+y} - 1.6 \cdot 10^{-7}}{2 - \sin(x+y) + \left \frac{x}{y} \right };$ $B = \operatorname{arctg} \left(\frac{x+y}{A} \right) + \frac{A}{\cos(xy)} \cdot \frac{x}{y};$	1	x=-2.3 y=4.5	A=6.065 B=-4.804
		2	x=-0.5 y=2.4	A=6.279 B=3.904
11	$A = \left(x \cdot \operatorname{ctg} \frac{y}{x^2 + y^2} + \frac{y}{2} \cdot \ln(x^2 + y^2) \right)^3;$ $B = \frac{\sin^2(x^2 + y^2) + A + 7.6}{3.2 \cdot 10^{-4} + 2 \cdot x^2 + \sqrt{x^2 + y^2}};$	1	x=1.5 y=-2.3	A=-355.97 B=-47.95
		2	x=1.5 y=-2.4	A=-397.45 B=-53.05
12	$A = \operatorname{tg} x^2 + \left(\sin^2 2 \cdot x + \frac{\lg x^2 + 2y }{2 \cdot x + y + 3.57} \right)^3;$ $B = A \cdot x \cdot \sqrt[3]{\frac{(2 \cdot x + y) \cdot 1.3 \cdot 10^6}{7 - \cos^2 y}} \cdot e^x;$	1	x=3 y=1.24	A=-0.447 B=-404.240
		2	x=5 y=1.6	A=-0.073 B=-251.282
13	$A = \cos^2(x^2 + 2y) + \frac{\ln x^2 + 2y }{(x^2 + 2y) + e^x};$ $B = \frac{6.3 \cdot \sin^2 \pi x}{2y} + \sqrt[3]{\frac{\ln x }{xy}};$	1	x=3 y=1.24	A=0.294 B=0.666
		2	x=5 y=1.6	A=1.013 B=0.586
14	$A = \frac{x}{y} + \frac{\frac{x}{y} + x + 1.75 \cdot 10^3 + y + \sqrt[3]{y}}{\sin(x+y) \cdot \operatorname{tg}^2 \frac{x}{x+y} + 4.32};$ $B = e^{Ax} \sqrt{\frac{\ln \left \frac{x}{y} (x + y)^2 \right }{\cos \frac{A}{x+y} + 14.3 \cdot 10^5}};$	1	x=-3.2 y=2.2	A=403.641 B=0.000
		2	x=-1.5 y=1.6	A=397.485 B=0.000
15	$A = \sqrt{\frac{x^3 + 7.3 \cdot 10^{-6} + 2y + e^y}{\cos^2(x^2 + e^y) - 0.743 + \sqrt[3]{x}}};$ $B = \operatorname{tg}^2(A + x^3 + e^y) + \frac{\lg Ay }{5.2 + \sin(x^2 + e^y + 2)};$	1	x=3 y=1.24	A=4.42 B=0.23
		2	x=5 y=1.6	A=11.66 B=0.29
16	$A = \operatorname{ctg} \frac{\frac{xy^2}{6.2} + \ln(xy^2) - \sin^2 x}{e^{\sin x} + 0.64^{-5} \cos \left(\frac{12.7}{xy^2} \right)};$	1	x=3 y=1.24	A=-3.214 B=-2.338

	$B = \lg \left \frac{2A + \sqrt{xy^2 + \sin x - y}}{e^{Ax} + (1 + xy^2)^4} \right ;$	2	x=5 y=1.6	A=1.250 B=-3.810
17	$A = \lg x + y^2 + \cos^3 \sqrt{y^3 \cdot \frac{\arctg(2.7+y^2)}{\lg x +y^2}};$ $B = \frac{\sqrt{A^3(y^2+\lg x)+31.1}}{2.4 \cdot 10^6 - \cos^2 \sqrt{x}};$	1	x=3 y=1.24	A=2.095 B=0.000
		2	x=-0.5555 y=1.6	A=2.305 B=0.000
18	$A = \frac{\sin^3(x+y)+ x +e^x}{e^{x+y}(18.6+ x +e^y)^3};$ $B = \frac{\arctg \sqrt{Ay+ x +e^y}}{ x +e^y \cdot 16.7 \cdot A^2 + \sqrt[3]{\cos^2(x+y)}};$	1	x=-1.7 y=-2.4	A=0.017 B=0.006
		2	x=-5 y=1.6	A=0.013 B=0.003
19	$A = \ctg \frac{x+y}{e^y} + \left(\frac{\sin y + \frac{e^y}{x+y}}{8.04 + \left \frac{x+y}{e^y} \right x^2} \right);$ $B = \sqrt[3]{\frac{\sqrt{\left(\frac{x+y}{e^y}\right)^2 + A^2}}{A^2 e^y + 1.8 \cdot 10^6}} - \cos^2 \pi x;$	1	x=2.5 y=0.42	A=-0.316 B=0.010
		2	x=-1.5 y=1.6	A=55.773 B=0.031
20	$A = \frac{\sin^2(x+1)^3 + y^2 + 1 + 10^{-6} \cdot x}{e^{x+1} + \sqrt{y^2 + 1}};$ $B = \ln \left \frac{A(y^2+1) - \sqrt[3]{A \cdot (x+1) \cdot y}}{\tg(y^2+1+A) + x-8 } \right ;$	1	x=-4.31 y=-2.42	A=2.952 B=0.312
		2	x=-1.5 y=-1.6	A=1.434 B=-0.398

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

Quyida berilgan chiziqli algoritmlarning dasturini tuzing.

$$A = \sin^2 \frac{(x-y^2)+\tg x}{(x-y^2)+e^{3.7}} \sqrt{1 + \lg^2 |x - y^2|};$$

$$B = \frac{\cos(Ax) \sqrt[3]{A(x-y^2)}}{10^{-5}(Ax-4)+e^A}.$$

bu yerda $x=1.27, y=0.4$. Yechimini 10^{-3} aniqlikda chiqaring.

Natija: $A=0.011$ va $B=0.227$

Ishdan maqsad.

Chiziqli algoritmlar va ular ustida amallar bajarish ko'nikmalarini oshirish. Dasturini tuzish.

Nazariy qism. Kiritish, chiqarish operatorlari, matematik funksiyalar.

Matematik funksiyalar C++ dasturlash tilida `<math.h>` kutubxonasida joylashgan bo'lib, ular quyidagicha yoziladi:

No	Matematik ko'rinishi	C++ dagi ko'rinishi	Izoh
Trigonometrik funksiyalar			
1	$\sin x$	$\sin(x)$	x radianga teng bo'lgan burchak sinusini qaytaradi
2	$\cos x$	$\cos(x)$	x radianga teng bo'lgan burchak kosinusini qaytaradi
3	$\operatorname{tg} x$	$\tan(x)$	x radianga teng bo'lgan burchak tangensini qaytaradi
4	$\operatorname{ctg} x$	$1/\tan(x)$	x radianga teng bo'lgan burchak kotangensini qaytaradi
5	$\sec x$	$\sinh(x)$	x radianga teng bo'lgan burchak giperbolik sinusini qaytaradi
6	$\operatorname{cosec} x$	$\cosh(x)$	x radianga teng bo'lgan burchak giperbolik kosinusini qaytaradi
Teskari trigonometrik funksiyalar			
1	$\arcsin x$	$\operatorname{asin}(x)$	$[-1;1]$ chegarada berilgan x ning arksinusini qaytaradi
2	$\arccos x$	$\operatorname{acos}(x)$	$[-1;1]$ chegarada berilgan

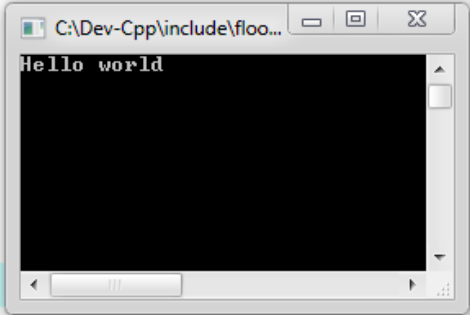
			x ning arkkosinusini qaytaradi
3	$\arctg x$	$atan(x)$	$[-1;1]$ chegarada berilgan x ning arktangensini qaytaradi
4	$\arcctg x$		$[-1;1]$ chegarada berilgan x ning arkkotangensini qaytaradi
Daraja, ildiz, logarifm, modul, e natural soni			
1	e^x	$exp(x)$	e^x qiymatini qaytaradi
2	a^x	$pow(a,x)$	a^x qiymatini qaytaradi
3	$ x $	$fabs(x)$	x sonining absolyut qiymatini qaytaradi
4	$\ln x$	$log(x)$	x sonining natural logarifmini qaytaradi
5	$\log_{10} x$	$log10(x)$	x sonining 10 asosLI logarifmini qaytaradi
6	$\sqrt{x}$	$sqrt(x)$	x sonining kvadrat ildizini qaytaradi
7	$e = 2.7182818284590452354$		e sonining qiymatini qaytaradi.
8	$\pi = 3.14159265358979323846$		π sonining qiymatini qaytaradi
1	x ga eng yaqin kichin butun son	$floor(x)$	x haqiqiy qiymatni eng yaqin kichik butun songa aylantiradi va uni haqiqiy ko'rinishda qaytaradi
2	x ga eng yaqin katta butun son	$ceil(x)$	x haqiqiy qiymatni eng yaqin katta butun songa aylantiradi va uni haqiqiy ko'rinishda qaytaradi
3	a ni b ga bo'lgandagi	$a \% b$	Butun a sonini butun b

	<i>qoldiq</i>		soniga bo'lgandagi qoldiqni butun ko'rinishda qaytaradi.
4	<i>a/b bo'linmaning butun qismi</i>	<i>a/b</i>	Butun a sonini butun b soniga bo'lgandagi bo'linmaning butun qismini qaytaradi
5	<i>$c=\sqrt{a^2 + b^2}$ gipotenuza</i>	<i>hypot(a,b)</i>	To'g'ri burchakli uchburchakning katetlari bo'yicha gipotenuzani hisoblab natijaga qaytaradi

C++ dasturlash muhitida ishlashga misollar.

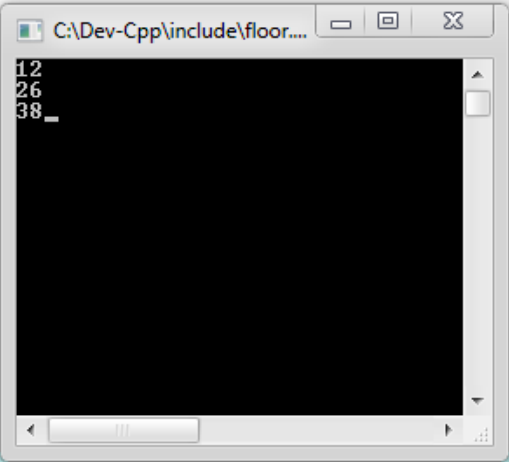
1 – misol. 1 –dastur:

```
#include <iostream>
#include <math.h>
using namespace std;
int main() {
    cout<<"Hello world";
    system("pause">>void");
    return 0;
}
```



2 – misol. ikkita a va b sonini yig'indisini hisoblovchi dastur:

```
#include <iostream>
#include <math.h>
using namespace std;
int main() {
    int a,b;
    cin>>a>>b;
    cout<<a+b;
    system("pause">>void");
    return 0;
}
```



Berilgan masala uchun algoritm.

Yuqorida berilgan masalani hisoblash uchun birinchi navbatda uni matematik ko'rinishdan C++ dasturlash muhiti o'girish lozim.

$$A = \sin^2 \frac{(x-y^2)+tgx}{(x-y^2)+e^{3.7}} \sqrt{1 + lg^2 |x - y^2|};$$
$$B = \frac{\cos(Ax) \sqrt[3]{A(x-y^2)}}{10^{-5}(Ax-4)+e^A}.$$

C++ muhitidagi ko'rinishi:

```
A=pow(sin(((x-y*y)+tan(x))/((x-y*y)+exp(3.7))),2)*sqrt(1+pow(log10(fabs(x-y*y)),2));
```

```
B=cos(A*x)*pow(A*(x-y*y),1.0/3)/(0.00001*(A*x-4)+exp(A));
```

Asosiy dastur matni:

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
double A,B,x,y;
cin>>x>>y;
A=pow(sin(((x-y*y)+tan(x))/((x-y*y)+exp(3.7))),2)*sqrt(1+pow(log10(fabs(x-y*y)),2));
B=cos(A*x)*pow(A*(x-y*y),1.0/3)/(0.00001*(A*x-4)+exp(A));
cout.precision(3);
cout<<fixed<<A<<endl<<B;
system("pause>>void");
return 0;
}
```

Berilgan misol ishlanish natijalari tahlili.

1. **#include** – kutubxonalarni qo'shish direktivasi;
2. **<iostream>** - kiritish chiqarish kutubxonasi;
3. **<math.h>** - matematik funksiyalar berilgan kutubxona;
4. **using namespace std** – std nomlar fazosidan standart oqimda kiritish, chiqarishda foydalaniladi;

```
int main()    {
...
}
```

return 0;} - asosiy funksiya dastur sintaksis to'g'ri yozilgan bo'lsa funksiya return orqali 0 qiymatni qaytaradi.

5. **double A, B, x, y** - haqiqiy turga mansub 4 ta o'zgaruvchi uchun xotiradan joy ajratadi.

6. **cin>>x>>y** - xotiradan ajratilgan x va y nomli haqiqiy o'zgaruvchilarga klaviaturadan qiymat kiritiladi.

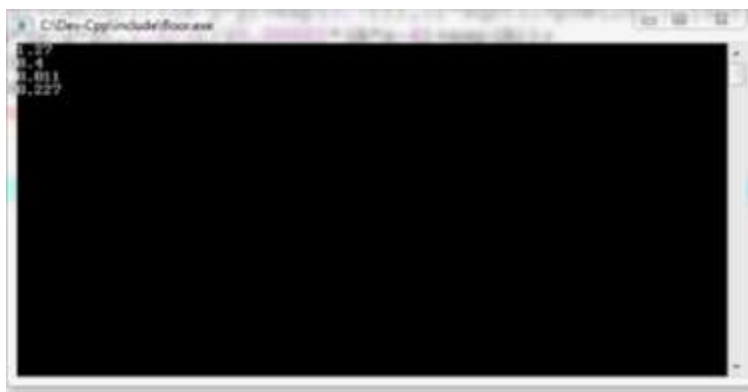
7. `A=pow(sin(((x-y*y)+tan(x))/((x-y*y)+exp(3.7))),2)*sqrt(1+pow(log10(fabs(x-y*y)),2));`
xotiradan ajratilgan A nomli o'zgaruvchiga yuqoridagi funksiya qiymati ta'minlanadi.

8. `B=cos(A*x)*pow(A*(x-y*y),1.0/3)/(0.00001*(A*x-4)+exp(A));` - xotiradan ajratilgan B nomli o'zgaruvchiga yuqoridagi funksiya qiymati ta'minlanadi.

9. **cout.precision(3); cout<<fixed<<A<<endl<<B - A** va B o'zgaruvchilarda mavjud qiymatlarni ekranga 10^{-3} aniqlikda yaxlitlab chiqaradi.

10. **system("pause>>void")** - natijani ekranda ushlab turish uchun foydalaniladi.

Natijalar listingi.



Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. C/C++ tilida masalalar va topshiriqlar, Yusupov R. A., Mirzanova Sh, Iskandarova S. N., Samarqand – 2013.
3. cplusplus.com

Laboratoriya ishi №4

Mavzu: Tarmoqlanuvchi dasturlash.

I. Variantlar. Quyidagi variantlarda berilgan tarmoqlanuvchi algoritmlar dasturini tuzing.

№	Variantlar	№	Kirish	Chiqish
1	$Y = \begin{cases} m^2n + 1 - c, & \text{agar } n + 1 > 0 \\ (m + n)^2 + cm^2, & \text{agar } n + 1 \leq 0 \end{cases}$	1	$m=1$ $n=2$ $c=3$	$Y=0.000$
		2	$m=2$ $n=3$ $c=7$	$Y=6.000$
2	$Y = \begin{cases} \frac{1}{\sqrt{9+x^2}}, & \text{agar } a < 5 \\ b \cdot \sin a, & \text{agar } a \geq 5 \end{cases}$ bu yerda $a = d^2 + \frac{cd}{c^2 - d^2}$;	1	$b=1$ $c=2$ $d=3$ $x=4$	$Y=0.999$
		2	$b=2$ $c=3$ $d=7$ $x=-6$	$Y=-1.952$
3	$Z = \begin{cases} 7x^2 - 3ab - 5ab, & \text{agar } a \geq 0; \\ 15a - 7b, & \text{agar } a < 0 \end{cases};$	1	$a=1$ $b=2$ $x=3$	$Z=47.000$
		2	$a=-2$ $b=9$ $x=7$	$Z=-93.000$
4	$Y = \begin{cases} \frac{a^2+b^2}{c} + \sqrt{a^2+x}, & \text{agar } x \geq 0 \\ \frac{\sin x + b}{a-b}, & \text{agar } x < 0 \end{cases}$ bu yerda $x = \frac{a^2-b^2}{c^2-b}$;	1	$a=1$ $b=2$ $c=3$	$Y=-1.090$
		2	$a=2$ $b=3$ $c=7$	$Y=-2.170$
5	$Y = \begin{cases} (nm^2 + d)^2, & \text{agar } d > 1 \\ \frac{d}{n^2+m^2}, & \text{agar } d \leq 1 \end{cases};$	1	$m=1$ $n=2$ $d=3$	$Y=25.000$
		2	$m=4$ $n=7$ $d=0.2$	$Y=0.003$
6	$Z = \begin{cases} \frac{ax^2}{b-1}, & \text{agar } a < 9 \\ (a+1)^3 + cx^3, & \text{agar } a \geq 9 \end{cases};$	1	$a=1$ $b=2$ $c=3$ $x=4$	$Z=16.000$
		2	$a=12$ $b=3$	$Z=7300.000$

			$c=7$ $x=9$	
7	$X = \begin{cases} \frac{a^3}{3+a}, & \text{agar } a > 0 \\ \sqrt{\left \frac{a^2+2}{1+a} \right }, & \text{agar } a \leq 0 \end{cases} \text{ bunda } a = \frac{b^2-c^2}{d};$	1	$b=4$ $c=-2$ $d=8$	$X=0.750$
		2	$b=4$ $c=-3$ $d=7$	$X=0.250$
8	$Z = \begin{cases} \sqrt{\left \frac{x}{x+1} \right } - \sqrt{x}, & \text{agar } x > 3 \\ \left(\frac{\ln x }{x} \right)^3, & \text{agar } x \leq 3 \end{cases}$	1	$x=2$	$Z=0.042$
		2	$x=3.7$	$Z=1.066$
9	$K = \begin{cases} xe^x \sin x, & \text{agar } x \geq 0 \\ \frac{1}{3} \ln^3 x , & \text{agar } x < 0 \end{cases};$	1	$x=3$	$K=8.503$
		2	$x=-7$	$K=2.456$
10	$Y = \begin{cases} \ln(x + \sqrt{x^2 + 9}), & \text{agar } x \geq 0; \\ tg^3 x, & \text{agar } x < 0 \end{cases};$	1	$x=3$	$Y=1.980$
		2	$x=-7.2$	$Y=0.308$
11	$Y = \begin{cases} \frac{a}{a^2+x}, & \text{agar } a > 5 \\ \frac{1}{2a^3+\sin a}, & \text{agar } a \leq 5 \end{cases};$	1	$a=4$ $x=9$	$Y=0.008$
		2	$a=7$ $x=1$	$Y=0.140$
12	$Y = \begin{cases} \sqrt{k}, & \text{agar } \sin k \leq 0.2 \\ \frac{1}{\sqrt{k}}, & \text{agar } \sin k > 0.2 \end{cases}$	1	$k=7$	$Y=0.378$
		2	$k=1.4$	$Y=0.845$
13	$Y = \begin{cases} \sin^2(2x) - \cos^2 x, & \text{agar } x > 0 \\ \frac{1}{x^2 - \sqrt{x}}, & \text{agar } x \leq 0.3 \end{cases};$	1	$x=0.5$	$Y=-0.062$
		2	$x=0.7$	$Y=0.386$
14	$P = \begin{cases} \frac{1}{2}(3x^2 - 1), & \text{agar } x > 0.4 \\ \frac{1}{2}(5x^3 - 3x), & \text{agar } x \leq 0.4 \end{cases};$	1	$x=7.3$	$P=79.435$
		2	$x=0.12$	$P=-0.176$
15	$Y = \begin{cases} x^2 + 4, & \text{agar } x < 10 \\ x^3 - 7, & \text{agar } x \geq 10 \end{cases} \text{ bunda } x = \frac{a^2-b}{c};$	1	$a=1$ $b=2$ $c=3$	$Y=4.11$
		2	$a=-2$ $b=3$ $c=7$	$Y=4.02$
16	$Y = \begin{cases} \sqrt{15a^2 + 21b^2}, & \text{agar } a > b; \\ \sqrt{15b^2 + 21a^2}, & \text{agar } a \leq b \end{cases};$	1	$a=7$ $b=3$	$Y=30.397$
		2	$a=1.2$ $b=3.8$	$Y=15.711$
17	$Y = \begin{cases} \ln 2x - 3z^2 , & \text{agar } x < 5z; \\ \ln 2x^2 - 3z , & \text{agar } x > 5z; \end{cases}$	1	$x=-10$ $z=3$	$Y=3.850$

		2	$x=6.3$ $z=-7$	$Y=4.609$
18	$P = \begin{cases} \sin(5k + 3m), & \text{agar } k > m , \\ \cos(5k + 3m), & \text{agar } k \leq m , \end{cases}$	1	$k=5$ $m=-6$	$Y=0.754$
		2	$k=-4.2$ $m=3.7$	$Y=0.458$
19	$Y = \begin{cases} \sqrt{2k_1 - 7k_2}, & \text{agar } k_1 k_2 < 1, \\ \sqrt{2k_1 + 7k_2}, & \text{agar } k_1 k_2 \geq 1, \end{cases}$	1	$k_1 = 0.02$ $k_2 = 4$	$Y=5.288$
		2	$k_1 = 9$ $k_2 = 7$	$Y=8.185$
20	$Y = \begin{cases} \frac{4r+3m}{r^2+m^2}, & \text{agar } r > m + \frac{1}{2}, \\ r - m , & \text{agar } r \leq m + \frac{1}{2}, \end{cases}$	1	$m=3$ $r=5$	$Y=0.853$
		2	$m=6.5$ $r=7$	$Y=0.500$

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

Quyida berilgan tarmoqlanuvchi algoritmlarning dasturini tuzing.

$$Y = \begin{cases} \sqrt{|n_1 \cdot n_2|} & \text{agar } n_1 n_2 < 0.2 \\ \sqrt{|n_1 + n_2|} & \text{agar } n_1 n_2 \geq 0.1 \end{cases}$$

Bunda $n_1 = -10$, $n_2 = 3$.

Yechimini 10^{-3} aniqlikda chiqaring. Natija: $Y = 5.477$

Ishdan maqsad.

Tarmoqlanuvchi algoritmlar va ular ustida amallar bajarish ko'nikmalarini oshirish. Dasturini tuzish.

Nazariy qism. *If* va *if else* operatorlari, *?: amali*.

Dastur bajarilishining birorta qadamida qandaydir shartni tekshirish natijasiga ko'ra boshqaruvni dasturning u yoki bu bo'lagiga uzatish mumkin (Tarmoqlanuvchi algoritmi). Tarmoqlanishni amalga oshirish uchun shartli operatoridan foydalaniladi.

If operatori. *If* qandaydir shartni rostlikka tekshirish natijasiga ko'ra dasturda tarmoqlanishni amalga oshiradi:

If (< shart >) <operator>;

Bu yerda <shart> har qanday ifoda bo'lishi mumkin, odatda u taqqoslash amali bo'ladi.

Agar shart 0 qiymatidan farqli yoki rost (true) bo'lsa, <operator> bajariladi, aks holda, ya'ni shart 0 yoki yolg'on (false) bo'lsa, hech qanday amal bajarilmaydi va boshqaruv if operatoridan keyingi operatorga o'tadi (agar u mavjud bo'lsa).

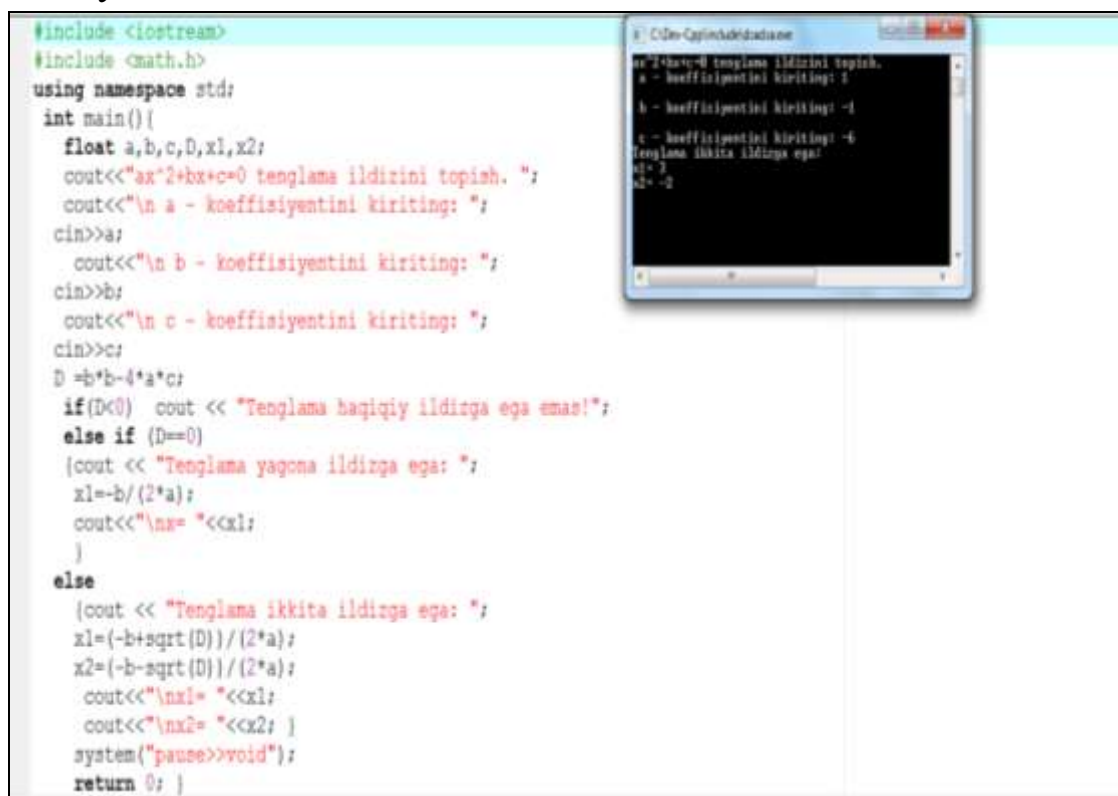
C++ tilining qurilmalari operatorlarni blok ko'rinishida tashkil qilishga imkon beradi. Blok - '{' va '}' belgi oralig'iga olingan operatorlar ketma-ketligi bo'lib, u kompilyator tomonidan yaxlit bir operator deb qabul qilinadi. Blok ichida e'lon operatorlari ham bo'lishi mumkin va ularda e'lon qilingan o'zgaruvchilar faqat shu blok ichida ko'rinadi (amal qiladi), blokdan tashqarida ko'rinmaydi. Blokdan keyin ';' belgisi qo'yilmasligi mumkin, lekin blok ichidagi har bir ifoda ';' belgisi bilan yakunlanishi shart.

Shart operatorining **if - else** ko'rinishi quyidagicha:

if (<shart-ifoda>) <operator1>; else <operator2>;

Bu yerda < shart – ifoda > 0 qiymatidan farqli yoki true bo'lsa, <operator1>, aks holda <operator2> bajariladi. Misol tariqasida diskriminantni hisoblash usuli yordamida $ax^2 + bx + c = 0$ ko'rinishidagi kvadrat tenglama ildizlarini topish masalasini

ko'raylik:



```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    float a,b,c,D,x1,x2;
    cout<<"ax^2+bx+c=0 tenglama ildizini topish. ";
    cout<<"\n a - koeffitsiyentini kiriting: ";
    cin>>a;
    cout<<"\n b - koeffitsiyentini kiriting: ";
    cin>>b;
    cout<<"\n c - koeffitsiyentini kiriting: ";
    cin>>c;
    D =b*b-4*a*c;
    if(D<0) cout << "Tenglama haqiqiy ildizga ega emas!";
    else if (D==0)
    {cout << "Tenglama yagona ildizga ega: ";
    x1=-b/(2*a);
    cout<<"\nx1= "<<x1;
    }
    else
    {cout << "Tenglama ikkita ildizga ega: ";
    x1=(-b+sqrt(D))/(2*a);
    x2=(-b-sqrt(D))/(2*a);
    cout<<"\nx1= "<<x1;
    cout<<"\nx2= "<<x2; }
    system("pause>>void");
    return 0; }
```

Dastur bajarilganda, birinchi navbatda tenglama koeffitsiyentlari - a, b, c o'zgaruvchilar qiymatlari kiritiladi, keyin diskriminant - D o'zgaruvchi qiymati hisoblanadi. Keyin D qiymatining manfiy ekanligi tekshiriladi. Agar shart o'rinli bo'lsa, yaxlit operator sifatida keluvchi '{' va '}' belgilari orasidagi operatorlar bajariladi va ekranga "Tenglama haqiqiy ildizlarga ega emas" xabari chiqadi va dastur o'z ishini tugatadi ("return 0;" operatorini bajarish orqali). Diskriminant noldan kichik bo'lmasa, navbatdagi shart operatori uni nolga tengligini tekshiradi. Agar shart o'rinli bo'lsa, keyingi qatorlardagi operatorlar bloki bajariladi - ekranga "Tenglama yagona ildizga ega:" xabari, hamda x_1 o'zgaruvchi qiymati chop etiladi va dastur shu yerda o'z ishini tugatadi, aks holda, ya'ni D qiymati noldan katta holati uchun else kalit so'zidan keyingi operatorlar bloki bajariladi va ekranga "Tenglama ikkita ildizga ega: " xabari, hamda x_1 va x_2 o'zgaruvchilar qiymatlari chop etiladi. Shu bilan shart operatoridan chiqiladi va asosiy funksiyaning return ko'rsatmasini bajarish orqali dastur o'z ishini tugatadi.

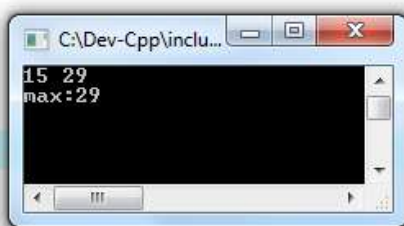
?: shart amali. Agar tekshirilayotgan shart nisbatan sodda bo'lsa, shart amalining “?:” ko'rinishini ishlatish mumkin:

$\langle \text{shart ifoda} \rangle ? \langle \text{ifoda1} \rangle : \langle \text{ifoda2} \rangle ;$

Shart amali *if* shart operatoriga o'xshash holda ishlaydi: agar $\langle \text{shart ifoda} \rangle$ 0 qiymatidan farqli yoki *true* bo'lsa, $\langle \text{ifoda1} \rangle$, holda $\langle \text{ifoda2} \rangle$ bajariladi. Odatda ifodalar qiymatlari birorta o'zgaruvchiga o'zlashtiriladi.

Misol:

```
#include <iostream>
using namespace std;
int main() {
    int a, b, max;
    cin >> a >> b;
    max = a > b ? a : b;
    cout << "max: " << max;
    system("pause >> void");
    return 0;
}
```



Berilgan masala uchun algoritm.

Yuqorida berilgan masalani hisoblash uchun birinchi navbatda uni matematik ko'rinishdan C++ dasturlash muhitiga o'tkazishda *if* shart operatoridan foydalanamiz.

$$Y = \begin{cases} \sqrt{|n_1 \cdot n_2|} & \text{agar } n_1 n_2 < 0.2 \\ \sqrt{|n_1 + n_2|} & \text{agar } n_1 n_2 \geq 0.1 \end{cases}$$

Asosiy dastur matni:

```

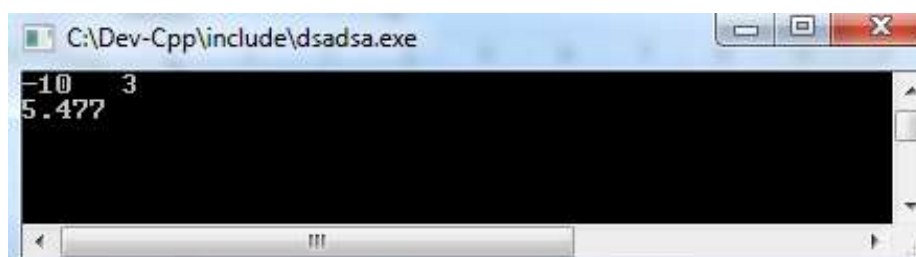
#include <iostream>
#include <math.h>
using namespace std;
int main() {
    double n1, n2, Y;
    cin >> n1 >> n2;
    if (n1*n2 < 0.2) {
        Y = sqrt(fabs(n1*n2));
    } else
    if (n1*n2 >= 0.1) Y = sqrt(fabs(n1+n2));
    cout.precision(3);
    cout << fixed << Y;
    system("pause >> void");
    return 0; }

```

Berilgan misol ishlanish natijalari tahlili.

n_1 va n_2 haqiqiy sonlari kiritilgandan so'ng ularning ko'paytmasi *if* shart operatori yordamida tekshirilib agarda $n_1 n_2 < 0.2$ shart bajarilsa $Y = \sqrt{|n_1 \cdot n_2|}$ qiymat, aks holda, $n_1 n_2 \geq 0.1$ shart bajarilsa $Y = \sqrt{|n_1 + n_2|}$ qiymatni o'zlashtiradi va natijaga 10^{-3} aniqlikda chiqaradi.

Natijalar listingi.



Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. C va C++ tili, Nazirov Sh. A., Kabulov R. V., Babajanov M. R., Raxmanov Q. S., Toshkent – 2013.
3. C/C++ tilida masalalar va topshiriqlar, Yusupov R. A., Mirzanova Sh, Iskandarova S. N., Samarqand – 2013.
4. <http://acm.tuit.uz>.

Laboratoriya ishi №5

Mavzu: Takrorlanuvchi dasturlash.

I. Variantlar.

Quyidagi variantlarda berilgan takrorlanuvchi algoritmlar dasturini tuzing.

№	Variantlar	№	Kirish	Chiqish
1	$a) \sum_{n=1}^m \frac{1}{n^3}; \quad b) \sum_{R=1}^l \frac{R^3}{R^4 + 3R^2 + e^{-R}};$ $c) \prod_{R=1}^l \prod_{i=1}^m \frac{R^i + 1}{R^4 + 3^i - \dots \cdot R + e^{-R}};$	1	$m=2$ $l=1$	$a) 1.12500000$ $b) 0.22894405$ $c) 0.08832821$
		2	$m=5$ $l=8$	$a) 1.18566204$ $b) 1.56011635$ $c) 0.00000000$
2	$a) \sum_{n=1}^k \frac{2}{n^3(n+1)}; \quad b) \sum_{R=1}^y \frac{R^2 + R-2 }{\ln R + 3R};$ $c) \sum_{R=1}^k \sum_{m=1}^p \frac{R^2 + R-2 }{\ln R + 3R};$	1	$k=2$ $p=1$	$a) 1.08333333$ $b) 0.66666667$ $c) 2.55180630$
		2	$k=5$ $p=8$	$a) 1.11076852$ $b) 12.34348563$ $c) 44.68083491$
3	$a) \sum_{n=1}^k \frac{3}{(2n+1)^3}; \quad b) \sum_{R=1}^p \frac{R+1}{R + e^{-R} + 1};$ $c) \prod_{R=1}^k \sum_{i=1}^p \frac{R+3}{R^3 + 3R + i^3};$	1	$k=2$ $p=1$	$a) 0.13511111$ $b) 0.90524345$ $c) 0.26666667$
		2	$k=5$ $p=8$	$a) 0.15022664$ $b) 169.36763033$ $c) 0.06609173$
4	$a) \sum_{R=1}^k \frac{4}{R(R+1)}; \quad b) \sum_{R=1}^p \frac{R^{R+1}}{2^{R+1} + (R+1)^4};$ $c) \sum_{R=1}^k \prod_{i=1}^p \frac{(R+1)^i + 4}{(-1)^R + 3(-1)^i + i^R};$	1	$k=2$ $p=1$	$a) 2.66666667$ $b) 0.05000000$ $c) -9.00000000$
		2	$k=5$ $p=8$	$a) 3.33333333$ $b) 20424.87608993$ $c) -204508918.09011152$
5	$a) \sum_{m=1}^k \frac{5}{m^2 + m + 4}; \quad b) \sum_{R=1}^p \frac{(100-R)^2}{\lg R + 5^{-R}};$ $c) \sum_{i=1}^k \sum_{R=1}^p \frac{(-1)^i \cos(i+R) + 5}{5i + 7^{-R} + i^{-R}};$	1	$k=2$ $p=1$	$a) 1.33333333$ $b) 49005.00000000$ $c) 1.25847755$
		2	$k=5$ $p=8$	$a) 2.00122549$ $b) 155695.30188719$ $c) 16.88785843$
6	$a) \prod_{n=1}^k \frac{n+6}{n^2 + 4n + 1}; \quad b) \sum_{i=1}^p \frac{i+6}{i^4 + 27i + 7};$	1	$k=2$ $p=1$	$a) 0.71794872$ $b) 0.20000000$ $c) 15.21598260$

	c) $\prod_{R=1}^k \prod_{i=1}^p (-1)^i \frac{\sqrt{5i^4 + e^{-R} + 6}}{\cos(i+1)^3 - R^{-i}};$	2	k=5 p=4	a)0.02128306 b)0.38410472 c)5851732838696673300000 000000.00000000
7	a) $k! + 7;$ b) $\sum_{i=1}^p \frac{(-1)^i \cdot 7^{-i}}{1+i+i^2};$	1	k=2 p=1	a)9.00000000 b)-0.0476190 c)1.2584776
	c) $\sum_{i=1}^k \sum_{R=1}^p \frac{(-1)^i \cos(i+R) + 5}{5i + 7^{-R} + i^{-R}};$	2	k=5 p=8	a)127.00000000 b)-0.0449098 c)16.8878584
8	a) $(k+1)!;$ b) $\sum_{n=1}^p \frac{10n-8}{10n^2-3n+8};$	1	k=2 p=1	a)6.00000000 b)0.13333333 c)16.43656366
	c) $\sum_{i=1}^k \sum_{m=1}^p \left[\frac{i^m + 4m + e^m}{m^i} \right];$	2	k=5 p=8	a)720.00000000 b)1.43814142 c)942.46222819
9	a) $\prod_{n=1}^k \frac{n^2 + 9^{-n}}{e^{-n} + n^{n-2}};$ b) $\sum_{i=1}^p (2i + 5i + 9);$	1	k=2 p=1	a)16.47419724 b)16.00000000 c)4.54389514
	c) $\sum_{R=1}^k \prod_{m=1}^p \sqrt{\left \frac{R + m^3 + e^{-m} + 9}{\log_2 R + (mR)^3} \right };$	2	k=5 p=8	a)9644244424.99799920 b)324.00000000 c)6.96548965
10	a) $\prod_{n=1}^k \frac{n^2 + 9^{-n}}{e^{-n} + n^{n-2}};$ b) $\prod_{n=1}^p \frac{n^2 + 3n + 10}{n^2 + 7n + 91};$	1	k=2 p=1	a)16.47419724 b)3.02633015 c)2.42622166
	c) $\sum_{R=1}^k \sum_{m=1}^p \frac{\sqrt{tg(R+m)^2 + 10R}}{R + m^{-R} + e^{m-R}};$	2	k=5 p=8	a)9644244424.99799920 b)13252651.75703296 c)28.57163344
11	a) $\prod_{n=1}^k \frac{n^2 + 9^{-n}}{e^{-n} + n^{n-2}};$ b) $\prod_{n=1}^p \frac{1}{n^4 + 1};$	1	k=2 p=1	a)16.7419724 b)0.50000000 c)136.46058442
	c) $\prod_{i=1}^k \sum_{m=1}^p \left[e^{\sqrt{i^2+m}-i} + \frac{i^2 + 11}{m^4 + i^{-m}} \right];$	2	k=5 p=8	a)9644244424.99799920 b)0.00000000 c)1293947622309.52320000
12	a) $\prod_{i=1}^k \frac{i^4 + i^2 + 12}{\sqrt{i^3 + e^{-i}}};$ b) $\sum_{R=1}^p \frac{R + 1.125}{R^5 + 5R + 1.2}$	1	k=2 p=1	a)134.29701962 b)0.29513889 c)1.48326269
	c) $\sum_{m=1}^k \prod_{n=1}^p \sqrt{\left \frac{m^3 - n^2 + 3.4}{m^{-n} - n^{-m} + 12} \right };$	2	k=5 p=8	a)5535312.16043921 b)0.39191141 c)5141.72021009
13	a) $\prod_{n=1}^k \frac{13}{n^3 + 5n + 7};$ b) $\sum_{m=1}^p \frac{(-1)^m \sqrt{m}}{2^{-2m}};$	1	k=2 p=1	a)0.52000000 b)-4.00000000 c)-0.00258075
		2	k=5	a)0.00163191

	c) $\prod_{i=1}^k \prod_{m=1}^p \operatorname{tg} \frac{i^{-m} - i^{-3} - i^2 + 1.3}{m^{-i} + m^{-6} + im + 13}$		$p=8$	b)150178.96310968 c)0.00000000
14	a) $\sum_{R=1}^k \frac{R^2 + 14}{\sqrt[3]{3^{-R} + R^3}}; \quad b) \prod_{n=1}^p \frac{n + 1.23}{n + \frac{1}{n}};$	1	$k=2$ $p=1$	a)19.31060301 b)1.11500000 c)-0.78146109
	c) $\sum_{i=1}^k \sum_{m=1}^p \lg \frac{\sqrt{m^2 + e^{m-i}}}{i^2 + 2^{m-1}};$	2	$k=5$ $p=8$	a)30.97176927 b)4.13609486 c)-28.76105925
15	a) $\prod_{i=1}^k \frac{ i - 15 + i^3}{\ln i + 7i}; \quad b) \sum_{R=1}^p \frac{(-1)^R \cdot (R + 1)}{R^3 + R^2 + 1}$	1	$k=2$ $p=1$	a)3.06265223 b)-0.66666667 c)0.03451403
	c) $\prod_{n=1}^k \sum_{m=1}^p (-1) \frac{m \ln(n + 5) + 1.5}{2^{m+3} + (n + 3)^{-m} + nm};$	2	$k=5$ $p=8$	a)50.86914472 b)-0.49910258 c)-0.07829932
16	a) $\sum_{i=-22}^k \frac{\sqrt{ i - 2i^3 } + 16}{\ln i + 3 + 1.6};$	1	$k=2$ $p=1$	a)462.95921847 b)-0.75299133 c)1.00000000
	b) $\sum_{n=1}^p (-1)^n \frac{n + 1.258974}{2n^4 + 1};$ c) $\prod_{R=4}^k \prod_{m=1}^p \frac{\sqrt{ R^m + 4R - m + 1.6 }}{\sin(m + R) - m};$	2	$k=5$ $p=8$	a)9644244424.99799920 b)0.00000000 c)1293947622309.52320000
17	a) $\prod_{R=1}^k \frac{R + 17}{2R^2 + 9};$	1	$k=2$ $p=1$	a)1.82887701 b)-0.17647059 c)1.04368781
	b) $\sum_{R=1}^p (-1) \cdot \frac{R \cdot \sqrt[R]{R + 1} + R^2}{2R^2 + 4R + 11};$ c) $\sum_{m=1}^k \prod_{n=1}^p \sqrt{\left \frac{m^3 - n^2 + 1.7}{m^n + m^m + 12} \right };$	2	$k=5$ $p=8$	a)0.25873599 b)-2.83439919 c)0.51638448
18	a) $\sum_{n=1}^k \frac{18}{5 - 17n + n^3};$	1	$k=2$ $p=1$	a)-2.49350649 b)0.41833001 c)1.00407388
	b) $\prod_{m=-12}^k \frac{m^2 \sqrt{ m } + 1.8}{m^2 + 4m + (-1)^m};$ c) $\sum_{i=1}^k \prod_{R=1}^p \frac{\sqrt{e^{i+R} (i + R)^R}}{ 4i^3 + R^4 };$	2	$k=5$ $p=8$	a)14.95912509 b)2.78668250 c)24681022165.89471400
19	a) $\sum_{n=1}^k \frac{19n}{3 + n + n^2};$	1	$k=2$ $p=1$	a)8.02222222 b)0.00000000 c)1.00000000
	b) $\sum_{t=3}^p \frac{\operatorname{tg}(t + 3)}{t^3 + 2t + e^{t-1}};$	2	$k=5$ $p=8$	a)18.00535793 b)-0.17303251 c)9737209203758591500000

	c) $\prod_{i=1}^k \prod_{m=2}^p (i \cdot m)^{\frac{i\sqrt{ i } + m}{(i+m)}}$			00000000000.00000000
20	a) $\prod_{n=1}^k (-1)^n \frac{1+n^2}{1+n^3};$	1	k=2 p=1	a)-0.55555556 b)0.00000000 c)180.00000000
	b) $\sum_{m=10}^p \frac{\sin(m)}{\sqrt[4]{m^2 + e^{m+13}}};$ c) $\prod_{n=1}^k \prod_{R=1}^p \frac{n^3 - R^2 + 20}{(n - R + n)^R};$	2	k=5 p=18	a)-0.25360195 b)-0.00392117 c)0.00000000

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

Quyida berilgan takrorlanuvchi algoritmlarning dasturini tuzing.

$$a) k!; \quad b) \sum_{n=1}^k \frac{10n - 30}{10n^2 - 3n + 8}; \quad c) \sum_{R=1}^k \sum_{m=1}^p \frac{\sqrt{(R+m)^2 + 5R}}{R + m^3 + 2^{m-R} + 30};$$

Bunda $k=2$, $p=1$. Yechimini 10^{-8} aniqlikda chiqaring.

Natija: a) 2.00000000 b) -1.33333333 c) 0.22102548

Ishdan maqsad.

Takrorlanuvchi algoritmlar va ular ustida amallar bajarish ko'nikmalarini oshirish. Dasturini tuzish.

Nazariy qism. *For*, *while* va *do while* takrorlash operatorlari, *break*, *continue*, *goto* operatorlari.

Dastur bajarilishini boshqarishning boshqa bir kuchli mexanizmlaridan biri - takrorlash operatorlari hisoblanadi. Takrorlash operatori «takrorlash sharti» deb nomlanuvchi ifodaning rost qiymatida dasturning ma'lum bir qismidagi operatorlarni (takrorlash tanasini) ko'p marta takror ravishda bajaradi (itarativ jarayon).

Takrorlash shartini tekshirish takrorlash tanasidagi operatorlarni bajarishdan oldin tekshirilishi mumkin (*for*, *while* takrorlashlari) yoki takrorlash tanasidagi operatorlari bir marta bajarilgandan keyin

tekshirilishi mumkin (**do-while**). Takrorlash operatorlari ichma-ich joylashgan bo'lishi mumkin.

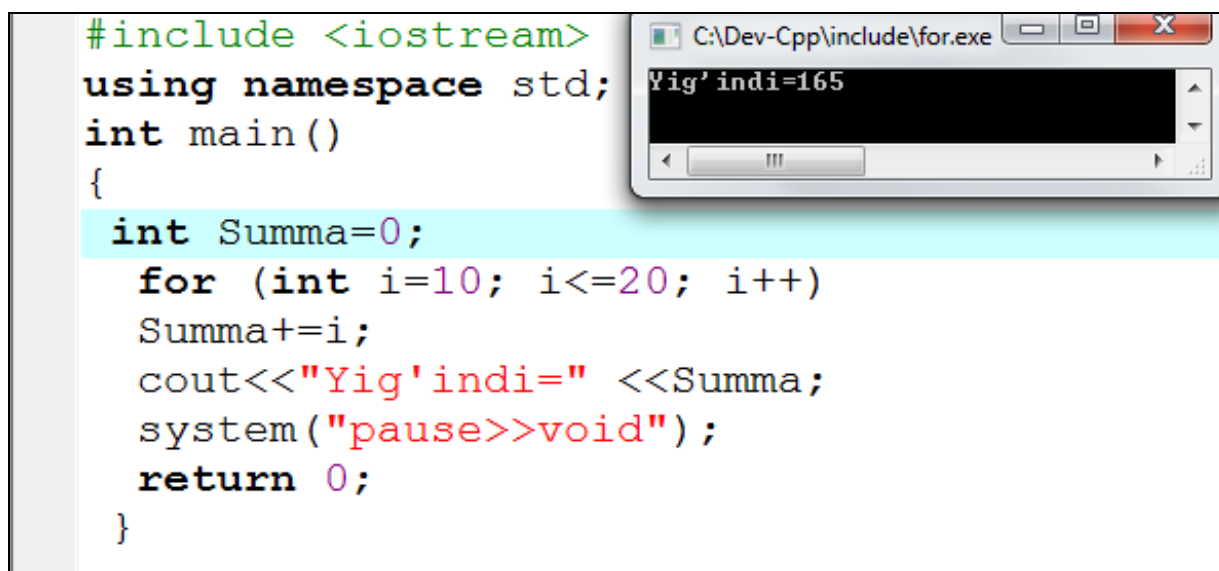
for takrorlash operatori

for takrorlash operatorining sintaksisi qo'yidagi ko'rinishga ega:

`for (<ifoda1>; <ifoda2>;<ifoda3>) <operator yoki blok>;`

Bu operator o'z ishini <ifoda1> ifodasini bajarishdan boshlaydi. Keyin takrorlash qadamlari boshlanadi. Har bir qadamda <ifoda2> bajariladi, agar natija 0 qiymatidan farqli yoki *true* bo'lsa, takrorlash tanasi - <operator yoki blok> bajariladi va oxirida <ifoda3> bajariladi. Agar <ifoda2> qiymati 0 (*false*) bo'lsa, takrorlash jarayoni to'xtaydi va boshqaruv takrorlash operatoridan keyingi operatorga o'tadi. Shuni qayd qilish kerakki, <ifoda2> ifodasi vergul bilan ajratilgan bir nechta ifodalar birlashmasidan iborat bo'lishi mumkin, bu holda oxirgi ifoda qiymati takrorlash sharti hisoblanadi. Takrorlash tanasi sifatida bitta operator, jumladan bo'sh operator bo'lishi yoki operatorlar bloki kelishi mumkin.

Misol uchun 10 dan 20 gacha bo'lgan butun sonlar yig'indisini hisoblash masalasini ko'raylik.



```
#include <iostream>
using namespace std;
int main()
{
    int Summa=0;
    for (int i=10; i<=20; i++)
        Summa+=i;
    cout<<"Yig'indi=" <<Summa;
    system("pause>>void");
    return 0;
}
```

Dasturdagi takrorlash operatori o'z ishini, i takrorlash parametriga (takrorlash sanagichiga) boshlang'ich qiymat 10 sonini berishdan boshlaydi va har bir takrorlash qadamidan (itarasiyadan)

keyin qavs ichidagi uchinchi operator bajarilishi hisobiga uning qiymati bittaga oshadi. Har bir takrorlash qadamida takrorlash tanasidagi operator bajariladi, ya'ni Summa o'zgaruvchisiga i qiymati qo'shiladi. Takrorlash sanagichi i qiymati 21 bo'lganda " $i \leq 20$ " takrorlash sharti *false* (0-qiymati) bo'ladi va takrorlash tugaydi. Natijada boshqaruv takrorlash operatoridan keyingi *cout* operatoriga o'tadi va ekranga yig'indi chop etiladi. Yuqorida keltirilgan misolga qarab takrorlash operatorlarining qavs ichidagi ifodalariga izoh berish mumkin:

<*ifoda1*> - takrorlash sanagichi vazifasini bajaruvchi o'zgaruvchiga boshlang'ich qiymat berishga xizmat qiladi va u takrorlash jarayoni boshida faqat bir marta hisoblanadi. Ifodada o'zgaruvchi e'loni uchrashi mumkin va bu o'zgaruvchi takrorlash operatori tanasida amal qiladi va takrorlash operatoridan tashqarida «ko'rinmaydi»

<*ifoda2*> - takrorlashni bajarish yoki yo'qligini aniqlab beruvchi mantiqiy ifoda, agar shart rost bo'lsa, takrorlash davom etadi, aks holda yo'q. Agar bu ifoda bo'sh bo'lsa, shart doimo rost deb hisoblanadi;

<*ifoda3*> - odatda takrorlash sanagichining qiymatini oshirish (kamaytirish) uchun xizmat qiladi yoki unda takrorlash shartiga ta'sir qiluvchi boshqa amallar bo'lishi mumkin.

Takrorlash operatorida qavs ichidagi ifodalar bo'lmasligi mumkin, lekin sintaksis ';' bo'lmasligiga ruxsat bermaydi. Shu sababli, eng sodda ko'rinishdagi takrorlash operatori quyidagicha bo'ladi:

for (;);cout <<"Cheksiz takrorlash..." ;

Agar takrorlash jarayonida bir nechta o'zgaruvchilarning qiymati sinxron ravishda o'zgarishi kerak bo'lsa, takrorlash ifodalarida zarur operatorlarni ',' bilan yozish orqali bunga erishish mumkin:

for(int i=10;j=2;i<=20;i++,j=i+10) {...};

Takrorlash operatorining har bir qadamida j va i o'zgaruvchilarning qiymatlari mos ravishda o'zgarib boradi. *for* operatorida takrorlash tanasi bo'lmasligi ham mumkin. Masalan, dastur bajarilishini ma'lum bir muddatga «to'xtab» turish zarur bo'lsa, bunga takrorlashni

hech qanday qo'shimcha ishlarni bajarmasdan amal qilishi orqali erishish mumkin:

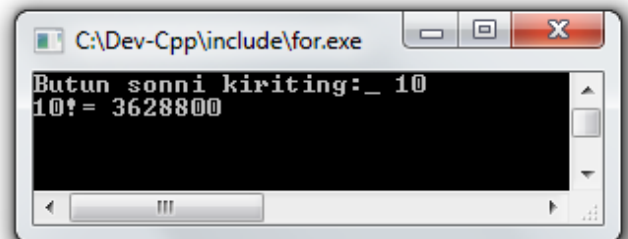
```
#include <iostream.h>
int main()
{int delay;
...
for(delay=5000;delay>0;delay--) ;//bo' shoperator
...
return 0;}
```

Yuqorida keltirilgan 10 dan 20 gacha bo'lgan sonlar yig'indisini bo'sh tanali takrorlash operatori orqali hisoblash mumkin:

```
...
for (int i=10; i<=20; Summa+=i++);
...
```

Takrorlash operatori tanasi sifatida operatorlar bloki ishlatishini faktorialni hisoblash misolida ko'rsatish mumkin:

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    unsigned long fact=1;
    cout <<"Butun sonni kiriting:_ ";
    cin >>a;
    if ((a>=0) && (a<33))
    {
        for (int i=1; i<=a; i++) fact*=i;
        cout<<a<<"!= " <<fact<<'\\n';
    }
    system("pause>>void");
    return 0;
}
```



Dastur foydalanuvchi tomonidan 0 dan 33 gacha oraliqdagi son kiritilganda amal qiladi, chunki 34! qiymati unsigned long uchun ajratilgan razryadlarga sig'maydi.

***while* takrorlash operatori**

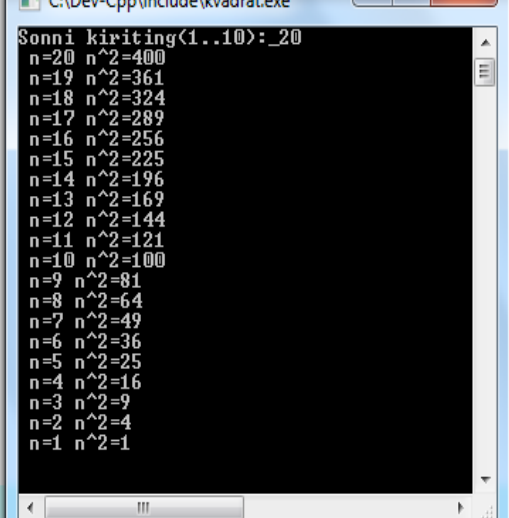
while takrorlash operatori, operator yoki blokni takrorlash sharti yolg'on (false yoki 0) bo'lguncha takror bajaradi. U quyidagi sintaksisga ega:

while (<ifoda>) <operator yoki blok>;

Agar <ifoda> rost qiymatli o'zgarmas ifoda bo'lsa, takrorlash cheksiz bo'ladi. Xuddi shunday, <ifoda> takrorlash boshlanishida rost bo'lib, uning qiymatiga takrorlash tanasidagi hisoblash ta'sir etmasa, ya'ni uning qiymati o'zgarmasa, takrorlash cheksiz bo'ladi. *while* takrorlash shartini oldindan tekshiruvchi takrorlash operatori hisoblanadi. Agar takrorlash boshida <ifoda> yolg'on bo'lsa, *while* operatori tarkibidagi <operator yoki blok> qismi bajarilmasdan cheklab o'tiladi. Ayrim hollarda <ifoda> qiymat berish operatori ko'rinishida kelishi mumkin. Bunda qiymat berish amali bajariladi va natija 0 bilan solishtiriladi. Natija noldan farqli bo'lsa, takrorlash davom ettiriladi. Agar rost ifodaning qiymati noldan farqli o'zgarmas bo'lsa, cheksiz takrorlash ro'y beradi. Masalan:

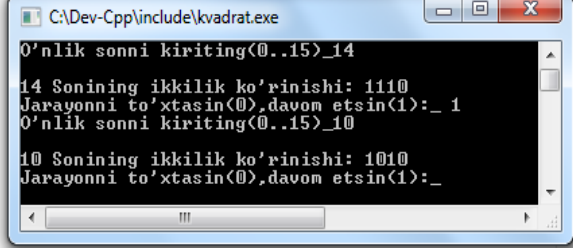
while(1); // cheksiz takrorlash Xuddi ***for*** operatoridek, ',' yordamida <ifoda> da bir nechta amallar sinxron ravishda bajarish mumkin. Masalan, son va uning kvadratlarini chop qiladigan dasturda ushbu holat ko'rsatilgan:

```
#include <iostream.h>
int main()
{
    int n,n2;
    cout<<"Sonni kiriting(1..10):_";
    cin>>n;
    n++;
    while (n--,n2=n*n,n>0)
        cout<<" n="<<n<<" n^2="<<n2<<endl;
    system("pause>>void");
    return 0;
}
```

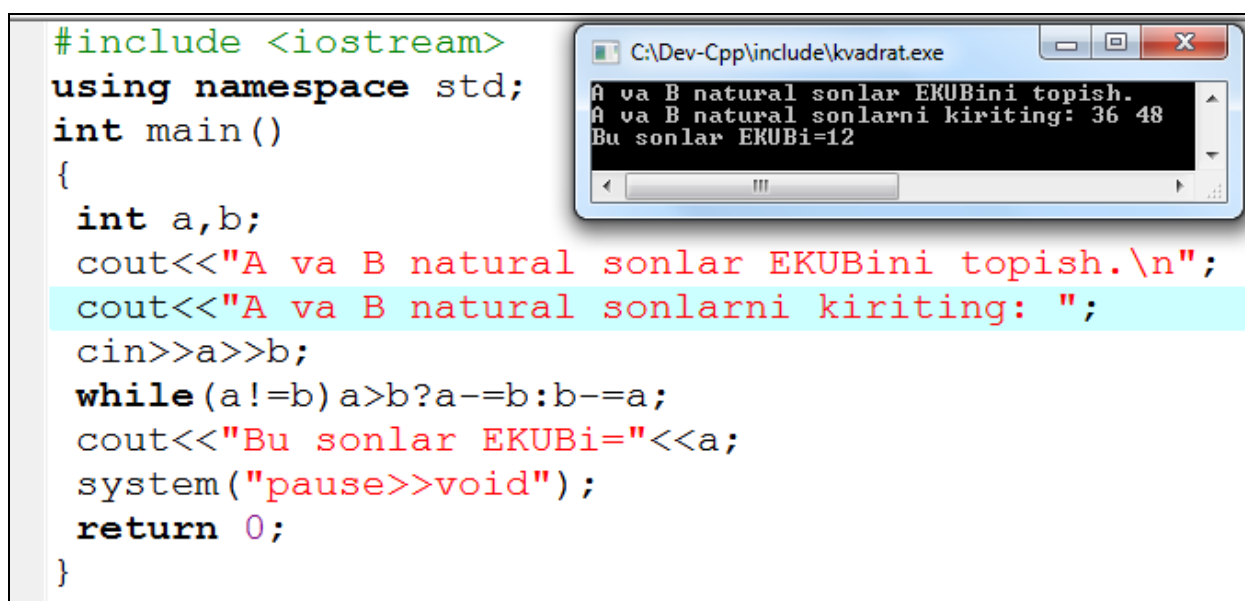


Dasturdagi takrorlash operatori bajarilishida n soni 1 gacha kamayib boradi. Har bir qadamda n va uning kvadrati chop qilinadi. Shunga e'tibor berish kerakki, shart ifodasida operatorlarni yozilish ketma-ketligining ahamiyati bor, chunki eng oxirgi operator takrorlash sharti sifatida qaraladi va n qiymati 0 bo'lganda takrorlash tugaydi. Keyingi dasturda berilgan o'nlik sonning ikkilik ko'rinishini chop qilish masalasini yechishda while operatorini qo'llash ko'rsatilgan.

```
#include <iostream.h>
int main()
{
    int sanagich=4;
    short son10,jarayon=1;
    while (jarayon) // cheksiz takrorlash
    {cout <<"0'nlik sonni kiriting(0..15)_"<<endl;
    cin >>son10;
    cout<<"\n"<<son10<<" Sonining ikkilik ko'rinishi: ";
    while (sanagich)
    {if(son10 & 8) //son10 & 00001000
        cout<<'1';
        else cout<<'0';
        son10=son10<<1; //razryadlarni 1 o'rin chapga surish
        sanagich--;
    }
    cout<<"\n";
    cout<<"Jarayonni to'xtasin(0),davom etsin(1):_ ";
    cin >> jarayon;
    sanagich=4;
}
return 0;
```



Dasturda ichma-ich joylashgan takrorlash operatorlari ishlatilgan. Birinchisi, sonning ikkilik ko'rinishini chop qilish jarayonini davom ettirish sharti bo'yicha amal qiladi. Ichki joylashgan ikkinchi takrorlash operatoridagi amallar - har qanday, 0 dan 15 gacha bo'lgan sonlar to'rtta razryadli ikkilik son ko'rinishida bo'lishiga asoslangan. Unda kiritilgan sonning ichki, ikkilik ko'rinishida uchinchi razryadida 0 yoki 1 turganligi aniqlanadi ("son10&8"). Shart natijasi natija 1 (rost) bo'lsa, ekranga '1', aks holda '0' belgisi chop etiladi. Keyingi qadamda son razryadlari chapga bittaga suriladi va yana uchinchi razryaddagi raqam chop etiladi. Takrorlash sanagich qiymati 0 bo'lguncha ya'ni to'rt marta bajariladi va boshqaruv ichki takrorlash operatoridan chiqadi. *while* takrorlash operatori yordamida samarali dastur kodi yozishga yana bir misol bu - ikkita natural sonlarning eng katta umumiy bo'luvchisini (EKUB) Evklid algoritmi bilan topish masalasini keltirishimiz mumkin:



The image shows a C++ program in a code editor and its execution output in a separate window. The code implements the Euclidean algorithm to find the Greatest Common Divisor (EKUB) of two numbers. The output window shows the program's execution with the input numbers 36 and 48, and the resulting EKUB of 12.

```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    cout<<"A va B natural sonlar EKUBini topish.\n";
    cout<<"A va B natural sonlarni kiriting: ";
    cin>>a>>b;
    while(a!=b) a>b?a-=b:b-=a;
    cout<<"Bu sonlar EKUBi="<<a;
    system("pause>>void");
    return 0;
}
```

C:\Dev-Cpp\include\kvadrat.exe

A va B natural sonlar EKUBini topish.
A va B natural sonlarni kiriting: 36 48
Bu sonlar EKUBi=12

Butun turdagi a va b qiymatlari oqimdan o'qilgandan keyin toki ularning qiymatlari o'zaro teng bo'lmaguncha takrorlash jarayoni ro'y beradi. Takrorlashning har bir qadamida a va b sonlarning kattasidan kichigi ayriladi. Takrorlashdan keyingi ko'rsatma vositasida a o'zgaruvchisining qiymati natija sifatida chop etiladi.

***do-while* takrorlash operatori**

do-while takrorlash operatori *while* operatoridan farqli ravishda oldin operator yoki blokni bajaradi, keyin takrorlash shartini tekshiradi. Bu qurilma takrorlash tanasini kamida bir marta bajarilishini ta'minlaydi. *do-while* takrorlash operatori quyidagi sintaksisga ega:

do <operator yoki blok>; *while* (<ifoda>);

Bunday takrorlash operatorining keng qo'llaniladigan holatlari - bu takrorlashni boshlamasdan turib, takrorlash shartini tekshirishning iloji bo'lmagan holatlar hisoblanadi. Masalan, birorta jarayonni davom ettirish yoki to'xtatish haqidagi so'rovga javob olish va uni tekshirish zarur bo'lsin. Ko'rinib turibdiki, jarayonni boshlamasdan oldin bu so'rovni berishning ma'nosi yo'q. Hech bo'lmaganda takrorlash jarayonining bitta qadami amalga oshirilgan bo'lishi kerak:

```
#include <iostream.h>
int main()
{
    char javob;
    do
    {
        ... // dastur tanasi
        cout<<"Jarayonni to'xtatish (N):_ ";
        cin>>javob;
    } while(javob !='N')
    return 0;
}
```

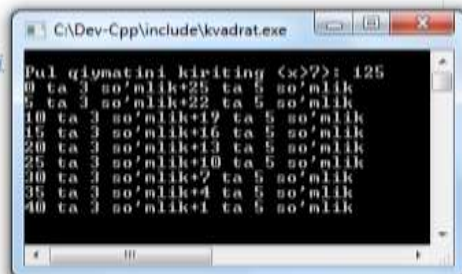
Dastur toki "Jarayonni to'xtatish (N):_ " so'roviga 'N' javobi kiritilmaguncha davom etadi. Bu operator ham cheksiz takrorlanishi mumkin:

do; while(1);

Masala. Har qanday 7 dan katta butun sondagi pul miqdorini 3 va 5 so'mliklarda berish mumkinligi isbotlansin. Qo'yilgan masala $p=3n+5m$ tenglamasi qanoatlantiruvchi m, n sonlar juftliklarini topish masalasidir (p -pul miqdori). Bu shartning bajarilishini m va n

o'zgaruvchilarining mumkin bo'lgan qiymatlarining barcha kombinasiyalarida tekshirish zarur bo'ladi.

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int Pul; //Pul- kiritiladigan pul miqdori
    unsigned n3,m5;   //n-3 so'mliklar,m-5 so'mliklar soni
    bool xato=false; //Pul qiymatini kiritishdagi xatolik
    do
    {
        if(xato) cout<<"Pul qiymati 7 dan kichik!";
        xato=true; // keyingi takrorlash xato hisoblanadi
        cout<<"\nPul qiymatini kiriting (x>7): ";
        cin>>Pul;
    }
    while(Pul<=7); //toki 7 katta son kiritilguncha
    n3=0;          //birorta ham 3 so'mlik yo'q
    do
    {
        m5=0;      //birorta ham 5 so'mlik yo'q
        do
        {
            if (3*n3+5*m5==Pul)
                cout<<n3<<" ta 3 so'mlik+"<<m5<<" ta 5 so'mlik\n";
            m5++; // 5 so'mliklar bittaga oshiriladi
        } while(3*n3+5*m5<=Pul);
        n3++; //3 s o'mliklar bittaga oshiriladi
    } while(3*n3<=Pul);
    system("pause>>void");
    return 0;
}
```



Dastur pul qiymatini kiritishni so'raydi (Pul o'zgaruvchisiga). Agar pul qiymati 7 sonidan kichik bo'lsa, bu haqda xabar beriladi va takror ravishda qiymat kiritish talab qilinadi. Pul qiymati 7 dan katta bo'lganda, 3 va 5 so'mliklarning mumkin bo'lgan to'la kombinasiyasini amalga oshirish uchun ichma-ich takrorlashlar amalga oshiriladi. Tashqi takrorlash n3 (3 so'mliklar miqdori) bo'yicha, ichki takrorlash esa m5 (5 so'mliklar miqdori) bo'yicha, toki bu miqdordagi pullar qiymati Pul qiymatidan oshib ketmaguncha davom etadi. Ichki takrorlashda m5 o'zgaruvchisining har bir qiymatida « $3*n3+5*m5==Pul$ » sharti tekshiriladi, agar u o'rinli bo'lsa, yechim varianti sifatida n3 va m5 o'zgaruvchilar qiymatlari chop etiladi.

break operatori

Takrorlash operatorlarining bajarilishida shunday holatlar yuzaga kelishi mumkinki, unda qaysidir qadamda, takrorlashni yakuniga yetkazmasdan takrorlashdan chiqish zarurati bo'lishi mumkin. Boshqacha aytganda, takrorlashni «uzish» kerak bo'lishi mumkin. Bunda break operatoridan foydalaniladi. break operatorini takrorlash operatori tanasining ixtiyoriy (zarur) joylariga qo'yish orqali shu joylardan takrorlashdan chiqishni amalga oshirish mumkin. E'tibor

beradigan bo'lsak, switch-case operatorining tub mohiyatiga ham break operatorini qo'llash orqali erishilgan. Ichma - ich joylashgan takrorlash va switch operatorlarida break operatori faqat o'zi joylashgan blokdan chiqish imkoniyatini beradi.

Quyidagi programmada ikkita ichma-ich joylashgan takrorlash operatoridan foydalangan holda foydalanuvchi tomonidan kiritilgan qandaydir sonni 3 va 7 sonlariga nisbatan qanday oraliqqa tushishi aniqlanadi. Tashqi takrorlashda "Son kiriting (0- to'xtash):_" so'rovi beriladi va javob javob_son o'zgaruvchisiga o'qiladi. Agar son noldan farqli bo'lsa, ichki takrorlash operatorida bu sonning qandaydir oraliqqa tushishi aniqlanib, shu haqida xabar beriladi va ichki takrorlash operatoridan chiqiladi. Tashqi takrorlashdagi so'rovga javob tariqasida 0 kiritilsa, programma o'z ishini tugatadi.

```
#include <iostream.h>
int main()
{
    int javob_son=0;
    do
    {
        while(javob_son)
        {
            if(javob_son<3)
                {cout<<"3 kichik !"; break;}
            if(3<=javob_son && javob_son <=7)
                {cout<<"3 va 7 oraligida !"; break;}
            if(javob_son>7)
                {cout<<"7 dan katta !"; break;}
        }
        cout<<"\nSon kiriting (0-to'xtash):_";
        cin>>javob_son;
    }
    while(javob_son !=0)
    return 0;
}
```

Amaliyotda break operatoridan cheksiz takrorlashdan chiqishda foydalaniladi.

```
for (;;)
```

```

{
    // 1- shart
    if (...)
    {
        ...
        break;
    }
    // 2- shart
    if (...)
    {
        ...
        break;
    }
    ...
}

```

Bu misolda cheksiz for takrorlashidan 1 yoki 2 - shart bajarilganda chiqiladi.

Masala. Ishorasiz butun sonlar ketma-ketligi 0 qiymati bilan tugaydi, 0 ketma-ketlik hadi hisoblanmaydi. Ketma-ketlikni kamaymaydigan holda tartiblangan yoki yo'qligi aniqlansin.

```

#include <iostream.h>
int main()
{
    unsigned int Ai_1=0,Ai;
    cout<<"Sonlar ketma-ketligini kiriting"
    cout<<(0-tugash alomati):\n ";
    cin>>Ai;      // ketma-ketlikning birinchi hadi
    while(Ai)
    {
        Ai_1=Ai;
        cin>>Ai;      // navbatdagi had
        if (Ai_1>Ai) break;
    }
    if(Ai_1)
    {
        cout<<"Ketma-ketlik tartiblangan";
        if(!Ai) cout<<" emas!";
        else cout<<"!";
    }
}

```

```

}
else cout<<"Ketma-ketlik bo'sh!";
return 0;
}

```

Dastur ishga tushganda, boshda ketma-ketlikning birinchi hadi alohida o'qib olinadi (A_i o'zgaruvchisiga). Keyin A_i qiymati nolga teng bo'lmaguncha takrorlash operatori amal qiladi. Takrorlash tanasida A_i qiymati oldingi qiymat sifatida A_{i-1} o'zgaruvchisida eslab qolinadi va navbatdagi had A_i o'zgaruvchisiga o'qiladi. Agar oldingi had navbatdagi haddan katta bo'lsa, break operatori yordamida takrorlash jarayoni uziladi va boshqaruv takrorlashdan keyingi shart operatoriga o'tadi. Bu yerdagi shart operatorlari mazmuni quyidagicha: agar A_{i-1} noldan farqli bo'lsa, ketma-ketlikning kamida bitta hadi kiritilgan bo'ladi (ketma-ketlik mavjud) va oxirgi kiritilgan had tekshiriladi. O'z navbatida agar A_i noldan farqli bo'lsa, bu holat hadlar o'rtasida kamaymaslik sharti bajarilmaganligi sababli hadlarni kiritish jarayoni uzilganini bildiradi va bu haqda xabar chop etiladi. Aks holda ketma-ketlikni kamaymaydigan holda tartiblangan bo'ladi.

continue operatori

continue operatori xuddi break operatoridek takrorlash operatori tanasini bajarishni to'xtatadi, lekin takrorlashdan chiqib ketmasdan keyingi qadamiga «sakrab» o'tishini tayinlaydi. continue operatorini qo'llanishiga misol tariqasida 2 va 50 sonlar oralig'idagi tub sonlarni topadigan programma matnini keltiramiz.

```

#include <iostream.h>
int main()
{
    bool bulinadi=false;
    for (int i=2; i<50; i++)
    {
        for (int j=2; j<i/2; j++)
        {
            if (i%j) continue;
            bulinadi=true;
            break;
        }
    }
}

```

```

    }
// break bajarilganda boshqaruv o'tadigan joy
    if (!bulinadi) cout <<i<<" ";
    bulinadi=false;
}

    return 0;
}

```

Keltirilgan programmada qo'yilgan masala ichma-ich joylashgan ikkita takrorlash operatorlari yordamida yechilgan. Birinchi takrorlash operatori 2 dan 50 gacha sonlarni hosil qilishga xizmat qiladi. Ichki takrorlash esa har bir hosil qilinayotgan sonni 2 sonidan toki shu sonning yarmigacha bo'lgan sonlarga bo'lib, qoldig'ini tekshiradi, agar qoldiq 0 sonidan farqli bo'lsa, navbatdagi songa bo'lish davom etadi, aks holda bulinadi o'zgaruvchisiga true qiymat berib, ichki takrorlash uziladi (son o'zining yarmigacha bo'lgan qandaydir songa bo'linar ekan, demak u tub emas va keyingi sonlarga bo'lib tekshirishga hojat yo'q). Ichki j bo'yicha takrorlashdan chiqqandan keyin bulinadi qiymati false bo'lsa (!bulinadi), i soni tub bo'ladi va u chop qilinadi.

goto operatori va nishonlar

Nishon - bu davomida ikkita nuqta (':') qo'yilgan identifikator. Nishon bilan qandaydir operator belgilanadi va keyinchalik, programmaning boshqa bir qismidan unga shartsiz o'tish amalga oshiriladi. Nishon bilan har qanday operator belgilanishi mumkin, shu jumladan e'lon operatori va bo'sh operatori ham. Nishon faqat funksiyalar ichida amal qiladi. Nishonga shartsiz o'tish goto operatori yordamida bajariladi. goto operatori orqali faqat uning o'zi joylashgan funksiya ichidagi operatorlarga o'tish mumkin. goto operatorining sintaksisi quyidagicha:

goto <nishon>;

Ayrim hollarda, goto operatorining «sakrab o'tishi» hisobiga xatoliklar yuzaga kelishi mumkin. Masalan,

```

int i=0;
i++; if(i)goto m;
int j;
m:j+=i;

```

operatorlarining bajarilishi xatolikka olib keladi, chunki `j` e'lon qilinmay qoladi. Shartsiz o'tish operatori programmani tuzishdagi kuchli va shu bilan birgalikda xavfli vositalardan biri hisoblanadi. Kuchliligi shundaki, uning yordamida algoritmnining «boshi berk» joylaridan chiqib ketish mumkin. Ikkinchi tomondan, bloklarning ichiga o'tish, masalan, takrorlash operatorlarini ichiga «sakrab» kirish kutilmagan holatlarni yuzaga keltirishi mumkin. Shu sababli, imkon qadar `goto` operatoridan foydalanmaslik kerak, ishlatilgan taqdirda ham quyidagi qoidaga amal qilish zarur: «blok ichiga, `if...else` va `switch` operatorlari ichiga, hamda takrorlash operatorlari tanasiga tashqaridan kirish mumkin emas».

Garchi, nishon yordamida programmaning ixtiyoriy joyiga o'tish mumkin bo'lsa ham, boshlang'ich qiymat berish e'lonlaridan sakrab o'tish man etiladi, lekin bloklardan sakrab o'tish mumkin.

Masalan:

```
...
goto B;      \ \ xato
float x=0.0;
goto B;      \ \ to'g'ri
{ int n=10; x=n*x+x; }
B: cout << "x=" << x;
...
```

Xususan, nishon yordamida ichki blokdan tashqi blokka va tashqi blokdan ichki blokka o'tishga C++ tili ruxsat beradi:

```
{ ...
    goto ABC:
    ...
    {int i=15;
        ...
        ABC:
        ...
        goto XYZ;
        int y=10;
        ...
        XYZ:
        ...
    }
```

```

        goto KLM;
        ... }
    ...
    int k=0;
    ...
    KLM:
    ...
}

```

Lekin, yuqorida keltirilgan misoldagi barcha o'tishlar mazmunan xato hisoblanadi. Quyidagi programmada ikkita natural sonlar eng katta umumiy bo'luvchini (EKUB) topish masalasidagi takrorlash jarayonini nishon va goto operatori vositasida amalga oshirish ko'rsatilgan:

```

int main()
{
    int a,b;
    cout<<"A va B natural sonlar EKUBini
topish.\n";
    cout<<"A va B natural sonlarni kiriting: "
    cin>>a>>b;
    nishon:
    if(a==b)
    {
        cout<<"Bu sonlar EKUBi: "<<a;
        return 0;
    }
    a>b?a-=b:b-=a;
    goto nishon;
}

```

dasturdagi nishon bilan belgilangan operatorlarda a va b sonlarni tengligi tekshiriladi. Agar ular teng bo'lsa, ixtiyoriy bittasi, masalan a soni EKUB bo'ladi va funksiyadan chiqiladi. Aks holda, bu sonlarning kattasidan kichigi ayiriladi va goto orqali ularning tengligi tekshiriladi. Takrorlash jarayoni a va b sonlar o'zaro teng bo'lguncha davom etadi.

Shuni qayd etish kerakki, bu masalani takrorlash operatorlari yordamida bajarish ancha samarali hisoblanadi.

Berilgan masala uchun algoritm.

Berilgan masalada uchta A, B va C misollar mavjud.

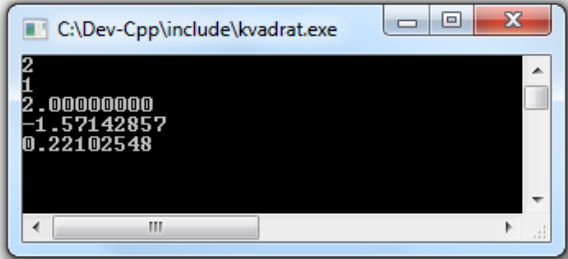
A misol: $k!$ hisoblash, ya'ni $1 \cdot 2 \cdot \dots \cdot k$ ko'paytma natijasini olish lozim. Bu yerda $A=1$ xaqiqiy o'zgaruvchi olamiz va k ta takrorlash natijasida 1 dan k gacha bo'lgan sonlarni A ga ko'paytiramiz.

B misol: $B=0$ haqiqiy o'zgaruvchi olamiz. $\frac{10n-30}{10n^2-3n+8}$ ifodaning natijasini $n=1$ dan toki $n=k$ gacha almashtirib hisoblaymiz va har birining natijasini B ga qo'shamiz.

C misol: Ushbu misol ichma-ich sikl bo'lganligi sababli $C=0$ va $C1=0$ haqiqiy qiymatli o'zgaruvchi olamiz. Tashqi siklning har bir holati uchun ichki sikl hisoblanib $C1$ ga qo'shiladi va natija C ga qo'shiladi.

Asosiy dastur matni.

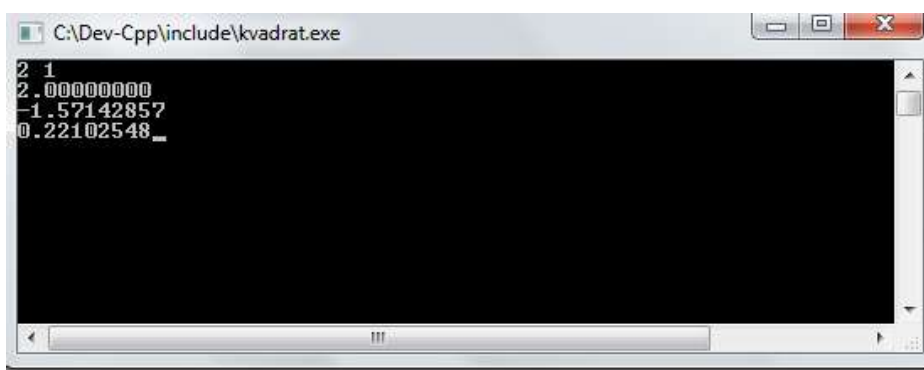
```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    double A=1,B=0,C=0,C1=0;
    __int64 i,k,p,n,m,r;
    cin>>k>>p;
    for (i=1;i<=k;i++)
        A*=i;
    for (n=1;n<=k;n++)
        B+=(10*n-30)*1.0/(10*n*n-3*n+8);
    for (r=1;r<=k;r++)
    {
        C1=0;
        for (m=1;m<=p;m++)
            C1+=sqrt(pow(r+m,2)+5*r)/(r+pow(m,3)+pow(2,m-r)+30);
        C+=C1;
    }
    cout.precision(8);
    cout<<fixed<<A<<"\n"<<B<<"\n"<<C;
    system("pause>>void");
    return 0;
}
```



Berilgan misol ishlanish natijalari tahlili.

k! faktorialni hisoblash **for (i=1;i<=k;i++) A*=i;** bu yerda **i=1** siklning initsializatsiya qismi, ya'ni i o'zgaruvchi 1 qiymatini o'zlashtiradi. **i<=k** siklning shart qismi, ushbu shart bajarilmaguncha davom etadi. **i++** siklning qadamlar qismi, ya'ni har safar A ga ko'paytirilib bir birlikka oshiriladi. Ikkinchi va uchinchi misolni hisoblashda ham sikldan huddi shu uslubda foydalanamiz.

Natijalar listingi.



The screenshot shows a window titled "C:\Dev-Cpp\include\kvadrat.exe" with a black background and white text. The output is as follows:

```
2 1
2.000000000
-1.57142857
0.22102548_
```

Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. C/C++ tilida masalalar va topshiriqlar, Yusupov R. A., Mirzanova Sh, Iskandarova S. N., Samarqand – 2013.
3. C/C++ и MS Visual C++ 2010 для начинающих, Пахомов Б. И. Санктпетербург – 2011.
4. <http://acm.tuit.uz>.
5. <http://cplusplus.com>.

Laboratoriya ishi №6

Mavzu: Bir o'lchamli massivlar.

I. Variantlar.

Quyidagi variantlarda berilgan massivlarga oid misollar dasturini tuzing.

№	Variantlar	№	Kirish	Chiqish
1	Bir o'lchamli sonli massivning o'rtacha qiymatidan kichik elementlarining o'rtacha qiymati hisoblansin.	1	6 58 22 17 84 50 53	19.50
		2	13 8 37 42 64 16 7 40 12 13 21 24 11 8	12.00
2	Bir o'lchamli sonli massiv [a,b] qismidagi elementlari massivning eng kichik elementiga bo'lib chiqilsin qolganlari o'zgartirishsiz qoldirilsin.	1	4 44 99 55 12 1 3	3.7 8.3 4.6 12.0
		2	2 15 8 1 1	1.9 8.0
3	Bir o'lchamli sonli massiv k – elementidan l – elementigacha bo'lgan elementlarining o'rtacha qiymati hisoblansin.	1	3 38 39 41 1 2	38.5
		2	10 6 93 73 62 26 28 65 74 53 59 8 9	63.5
4	Bir o'lchamli sonli massivning minimum elementini massivning oxirgi elementi bilan o'rin almashtirilsin.	1	4 74 0 1 33	74 33 1 0
		2	7 8 37 42 64 16 7 40	8 37 42 64 16 40 7
5	Bir o'lchamli sonli massiv [a,b] qismda yotmaydigan elementlarining o'rtacha qiymati hisoblansin.	1	6 51 49 9 76 56 78 1 4	67.00
		2	17 60 66 34 23 40 68 4 31 36 86 61 59 84 10 28 1 30 11 13	36.93
6	Bir o'lchamli sonli massiv elementlari kvadratlarining yig'indisi hisoblansin.	1	5 24 50 72 96 95	26501
		2	1 43	1849
7	Bir o'lchamli sonli massivning barcha elementlari massiv eng katta elementiga bo'lib chiqilsin.	1	9 37 23 51 47 12 23 90 85 98	0.38 0.23 0.52 0.48 0.12 0.23 0.92 0.87 1.00
		2	3 60 19 27	1.00 0.32 0.45
8	Bir o'lchamli sonli massivni barcha elementlari massiv eng kichik elementiga bo'lib chiqilsin.	1	9 69 48 58 45 57 55 88 89 85	1.53 1.07 1.29 1.00 1.27 1.22 1.96 1.98 1.89
		2	3 4 84 79	1.00 21.00 19.75
9	Bir o'lchamli sonli massiv M dan katta elementlarini ko'paytmalarini logarifmi hisoblansin.	1	5 26 18 32 68 81 6	18.23
		2	10 49 39 42 12 53 35 94 21 35 12 12	29.96
10	Bir o'lchamli sonli massiv K yoki M ga teng elementlari ko'paytmasi	1	5 7 11 83 18 31 31 3	31

	hisoblansin.	2	7 44 64 23 84 13 6 22 22 6	132
11	Bir o'lchamli sonli massiv M dan katta elementlari yig'indisi hisoblansin.	1	4 12 88 30 87 94	0
		2	9 9 72 18 48 75 32 29 78 25 76	78
12	Bir o'lchamli sonli massivni toq o'rindagi elementlari ko'paytmasini juft o'rindagi elementlari yig'indisiga bo'linsin.	1	2 32 8	4.00
		2	4 38 34 13 48	6.02
13	Bir o'lchamli sonli massiv manfiy elementlarini o'rtacha qiymati hisoblansin.	1	9 93 64 -90 74 62 -83 58 15 -37	-70.00
		2	10 63 89 -6 48 77 -19 16 73 -72 34	-32.33
14	Bir o'lchamli sonli massivni 2 ga va 5 ga bo'linadigan elementlarini ko'paytmasining sinusi topilsin.	1	6 44 34 42 83 43 64	0.02
		2	15 62 54 24 95 67 62 25 17 77 50 38 12 90 59 7	0.64
15	Bir o'lchamli sonli massivni M dan kichik elementlarini kvadratlari yig'indisi hisoblansin.	1	1 38 9	0
		2	14 85 15 57 68 18 67 7 45 69 21 1 5 98 34 92	28594
16	Bir o'lchamli sonli massiv elementlari massivning eng katta elementiga bo'lib chiqilsin.	1	5 91 51 75 85 29	1.00 0.56 0.82 0.93 0.32
		2	8 30 31 30 94 61 41 74 78	0.32 0.33 0.32 1.00 0.65 0.44 0.79 0.83
17	Bir o'lchamli sonli massiv toq o'rindagi elementlarini yig'indisi hisoblansin.	1	10 93 64 63 8 50 24 32 80 3 76	241
		2	2 85 88	85
18	Bir o'lchamli sonli massivning toq qiymatli elementlarini o'rtacha qiymati hisoblansin.	1	5 76 12 51 50 98	51.00
		2	13 23 87 77 4 14 57 91 16 80 7 45 78 46	55.29
19	Elementlari soni N ($1 \leq N \leq 10000$) ta bo'lgan butun sonli massiv berilgan. Ushbu massivning elementlarining qarama-qarshisini hamda necha marta ishora almashish amalga oshganligini aniqlovchi dastur tuzing. Massiv elementlarini qiymatlari [-1000,1000] oraliqda joylashgan.1	1	5 3 -3 9 -9 0	-3 3 -9 9 0 4
		2	10 0 1 2 3 4 5 0 7 8 -9	0 -1 -2 -3 -4 -5 0 -7 -8 9 8
20	Bir o'lchamli sonli massiv elementlarini qiymati [x,y] oraliqda yotmaydigan elementlari soni aniqlansin.	1	10 14 51 -83 42 85 -77 91 70 -98 54 50 99	5
		2	3 1 77 -58 20 97	2

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

Bir o'lchovli massivga oid quyidagi misolning dasturini tuzing.

Elementlari soni N ($1 \leq N \leq 10000$) ta bo'lgan butun sonli massiv berilgan. Ushbu massivning elementlarining qarama-qarshisini hamda necha marta ishora almashish amalga oshganligini aniqlovchi dastur tuzing. Massiv elementlarini qiymatlari $[-10^{-9}, 10^9]$ oraliqda joylashgan.

Bunda $N=5$ va $a_1 = 3, a_2 = -3, a_3 = 9, a_4 = -9, a_5 = 0$

Natija: $-3 \ 3 \ -9 \ 9 \ 0$.

Ishora almasishlar soni: 4

Ishdan maqsad.

Bir o'lchovli massivlar va ular ustida amallar bajarish ko'nikmalarini oshirish. Dasturini tuzish.

Nazariy qism. Bir o'lchovli massivlarni e'lon qilish va ular ustida amallar.

Xotirada ketma-ket (regulyar) joylashgan bir xil turdagi qiymatlarga massiv deyiladi. Odatda massivlarga zarurat, katta hajmdagi, lekin cheklangan miqdordagi va tartiblangan qiymatlarni qayta ishlash bilan bog'liq masalalarni yechishda yuzaga keladi. Faraz qilaylik, talabalar guruhining reyting ballari bilan ishlash masalasi qo'yilgan. Unda guruhning o'rtacha reytingini aniqlash, reytinglarni kamayishi bo'yicha tartiblash, konkret talabaning reytingi haqida ma'lumot berish va boshqa masala ostilarini yechish zarur bo'lsin. Qayd etilgan masalalarni yechish uchun berilganlarning (reytinglarning) tartiblangan ketma-ketligi zarur bo'ladi. Bu yerda tartiblanganlik ma'nosi shundaki, ketma-ketlikning har bir qiymati o'z o'rniga ega bo'ladi (birinchi talabaning reytingi massivda birinchi o'rinda, ikkinchi talabaniki - ikkinchi o'rinda va hakoza). Berilganlar ketma-ketligini ikki xil usulda hosil qilish mumkin. Birinchi yo'l - har bir reyting uchun alohida o'zgaruvchi aniqlash: $\text{Reyting}_1, \dots, \text{Reyting}_N$. Lekin, guruhdagi talabalar soni yetarlicha katta bo'lganda, bu

o'zgaruvchilar qatnashgan programmani tuzish katta qiyinchiliklarni yuzaga keltiradi. Ikkinchi yo'l - berilganlar ketma-ketligini yagona nom bilan aniqlab, uning qiymatlariga murojaatni, shu qiymatlarning ketma-ketlikda joylashgan o'rnining nomeri (indeksi) orqali amalga oshirishdir. Reytinglar ketma-ketligini Reyting deb nomlab, undagi qiymatlariga Reyting1,...,ReytingN ko'rinishida murojaat qilish mumkin. Odatda berilganlarning bunday ko'rinishiga massivlar deyiladi. Massivlarni matematikadagi sonlar vektoriga o'xshatish mumkin, chunki vektor ham o'zining individual nomiga ega va u fiksirlangan miqdordagi bir turdagi qiymatlardan - sonlardan iboratdir.

Demak, massiv - bu fiksirlangan miqdordagi ayrim qiymatlarning (massiv elementlarining) tartiblangan majmuasidir. Barcha elementlar bir xil turda bo'lishi kerak va bu tur element turi yoki massiv uchun tayanch tur deb nomlanadi. Yuqoridagi keltirilgan misolda Reyting - haqiqiy turdagi vektor deb nomlanadi.

Programmada ishlatiladigan har bir konkret massiv o'zining individual nomiga ega bo'lishi kerak. Bu nomni to'liq o'zgaruvchi deyiladi, chunki uning qiymati massivning o'zi bo'ladi. Massivning har bir elementi massiv nomi, hamda kvadrat qavsga olingan va element selektori deb nomlanuvchi indeksni ko'rsatish orqali oshkor ravishda belgilanadi. Murojaat sintaksisi:

$$< \text{massiv nomi} > [< \text{indeks} >]$$

Bu ko'rinishga xususiy o'zgaruvchi deyiladi, chunki uning qiymati massivning alohida elementidir. Bizning misolda Reyting massivining alohida komponentalariga Reyting[1],...,Reyting[N] xususiy o'zgaruvchilar orqali murojaat qilish mumkin. Boshqacha bu o'zgaruvchilar indeksli o'zgaruvchilar deyiladi. Massiv indeks sifatida butun son qo'llaniladi. Umuman olganda indeks sifatida butun son qiymatini qabul qiladigan ixtiyoriy ifoda ishlatilishi mumkin va uning qiymati massiv elementi nomerini aniqlaydi. Ifoda sifatida o'zgaruvchi ham olinishi mumkinki, o'zgaruvchining qiymati o'zgarishi bilan murojaat qilinayotgan massiv elementini aniqlovchi indeks ham o'zgaradi.

Shunday qilib, programmadagi bitta indeksli o'zgaruvchi orqali massivning barcha elementlarini belgilash (aniqlash) mumkin bo'ladi. Masalan, `Reyting[I]` o'zgaruvchisi orqali `I` o'zgaruvchining qiymatiga bog'liq ravishda `Reyting` massivining ixtiyoriy elementiga murojaat qilish mavjud.

Haqiqiy turdagi (`float`, `double`) qiymatlar to'plami cheksiz bo'lganligi sababli ular indeks sifatida ishlatilmaydi. C++ tilida indeks doimo 0 dan boshlanadi va uning eng katta qiymati massiv e'lonidagi uzunlikdan bittaga kam bo'ladi.

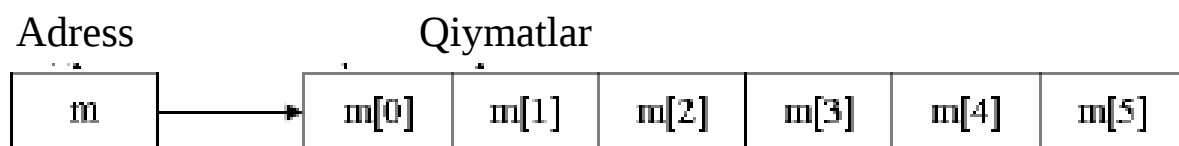
Massiv e'loni quyidagicha bo'ladi:

$\langle \text{tur} \rangle \langle \text{nom} \rangle [\langle \text{uzunlik} \rangle] = \{ \text{boshlang'ich qiymatlar} \}.$

Bu yerda $\langle \text{uzunlik} \rangle$ - o'zgarmas ifoda. Misollar:

`int m[6]={1,4,-5,2,10,3}; float a[4];`

Massiv statik va dinamik bo'lishi mumkin. Statik massivning uzunligi oldindan ma'lum bo'lib, u xotirada ma'lum adresdan boshlab ketma-ket joylashadi. Dinamik massivni uzunligi programma bajarilish jarayonida aniqlanib, u dinamik xotiradagi ayni paytda bo'sh bo'lgan adreslarga joylashadi. Masalan, **`int m[6];`** ko'rinishida e'lon qilingan bir o'lchamli massiv elementlari xotirada quyidagicha joylashadi:



Bir o'lchamli massivning xotirada joylashuvi

Massivning `i`- elementiga `m[i]` yoki `*(m+i)` - vositali murojaat qilish mumkin. Massiv uzunligini `sizeof(m)` amali orqali aniqladi. Massiv e'lonida uning elementlariga boshlang'ich qiymatlar berish mumkin va uning bir nechta variantlari mavjud.

1) o'lchami ko'rsatilgan massiv elementlarini to'liq inisializasiyalash:

`int t[5]={-10,5,15,4,3};`

Bunda 5 ta elementdan iborat bo'lgan `t` nomli butun turdagi bir o'lchamli massiv e'lon qilingan va uning barcha elementlariga boshlang'ich qiymatlar berilgan. Bu e'lon quyidagi e'lon bilan ekvivalent: **`int t[5]; t[0]=-10; t[1]=5; t[2]=15; t[3]=4; t[4]=3;`**

2) o'lchami ko'rsatilgan massiv elementlarini to'liqmas inisializasiyalash:

int t[5]={-10,5,15};

Bu yerda faqat massiv boshidagi uchta elementga boshlang'ich qiymatlar berilgan. Shuni aytib o'tish kerakki, massivning boshidagi yoki o'rtasidagi elementlariga qiymatlar bermasdan, uning oxiridagi elementlarga boshlang'ich qiymat berish mumkin emas. Agarda massiv elementlariga boshlang'ich qiymat berilmasa, unda kelishuv bo'yicha static va extern modifikatori bilan e'lon qilingan massiv uchun elementlarining qiymati 0 soniga teng deb, automatic massivlar elementlarining boshlang'ich qiymatlari noma'lum hisoblanadi.

3) o'lchami ko'rsatilmagan massiv elementlarini to'liq inisializasiyalash:

int t[]={-10,5,15,4,3};

Bu misolda massivni barcha elementlariga qiymatlar berilgan hisoblanadi, massiv uzunligi kompilyator tomonidan boshlang'ich qiymatlar soniga qarab aniqlanadi. Agarda massiv uzunligi berilmasa, boshlang'ich qiymati berilishi shart.

Massivni e'lon qilishga misollar:

char ch[4]={'a','b','c','d'}; //belgilar massivi

int in[6] = {10,20,30,40}; // butun sonlar massivi

char str[]="abcd"; //sitr uzunligi 5 ga teng, chunki uning oxiriga '\0' belgisi qo'shiladi

char str[]={ 'a','b','c','d'}; // yuqoridagi satrning boshqacha yozilishi

Masala. Berilgan massiv elementlarining har xil qiymatlilari soni toping. N – massiv elementlari soni ($1 \leq N \leq 100000$). ($-1000 \leq a[i] \leq 1000$).

Dastur matni:

```
vanilzorlar.cpp |
#include <iostream>
#include <stdio.h>
using namespace std;
int main(){
    int k,s=0,a[2001]={0},n,i;
    cin>>n;
    for (i=0;i<n;i++)
    {
        cin>>k;
        a[k+10]=1;
    }
    for (i=0;i<=2001;i++)
        s+=a[i];
    cout<<s;
    system("pause>>void");
    return 0;}

D:\acmp\vanilzorlar.exe
20
1 2 3 4 1 2 5 6 4 7 8 9 8 3 2 -1 -2 -1 -2 0
12
```

Berilgan masala uchun algoritm.

Berilgan masalada massiv elementlari qiymatlari ishoralarini almashtirish va almashtirishlar sonini aniqlash uchun bizga massiv elementlarining nechta 0 ga teng ekanligini aniqlash lozim bo'ladi. Chunki 0 qiymatli sonning ishorasi yo'q.

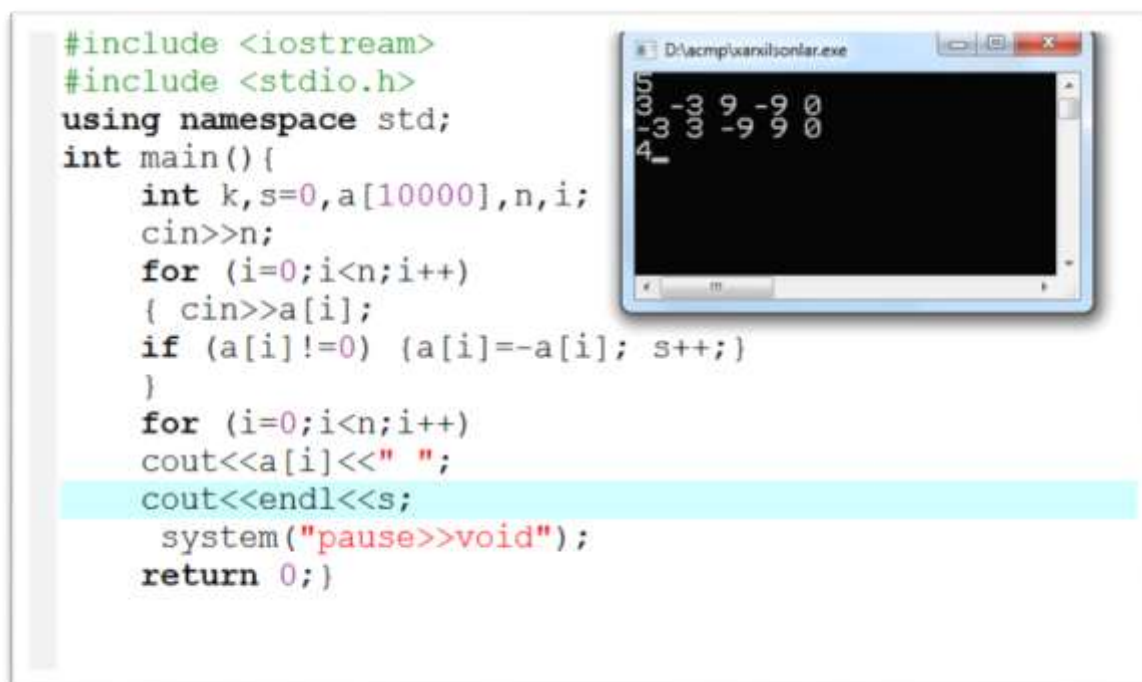
Asosiy dastur matni.

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main(){
    int k,s=0,a[10000],n,i;
    cin>>n;
    for (i=0;i<n;i++)
    { cin>>a[i];
      if (a[i]!=0) {a[i]=-a[i]; s++;}
    }
    for (i=0;i<n;i++)
        cout<<a[i]<<" ";
    cout<<endl<<s;
    system("pause>>void");
    return 0;}
```

Berilgan misol ishlanish natijalari tahlili.

Berilgan misolda massiv elementlarini kiritish jarayonida ushbu massiv elementlarining qiymati 0 dan farqli bo'lsa uning qiymatini qarama-qarshisiga almashtirilyapti va ushbu almashtirishlar shu joyda sanalayapti.

Natijalar listingi.



The image shows a C++ program on the left and its execution output on the right. The program reads an integer 'n', then 'n' integers into an array 'a'. It then iterates through the array, negating each non-zero element and incrementing a counter 's'. Finally, it prints the array elements and the value of 's'.

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main() {
    int k,s=0,a[10000],n,i;
    cin>>n;
    for (i=0;i<n;i++)
    { cin>>a[i];
    if (a[i]!=0) {a[i]=-a[i]; s++;}
    }
    for (i=0;i<n;i++)
    cout<<a[i]<<" ";
    cout<<endl<<s;
    system("pause>>void");
    return 0;}
```

The execution window shows the following input and output:

```
5
3 -3 9 -9 0
-3 3 -9 9 0
4
```

Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. C va C++ tili, Nazirov Sh. A., Kabulov R. V., Babajanov M. R., Raxmanov Q. S., Toshkent – 2013.
3. Решение сложных и олимпиадных задач по программированию, Долинский М. С., Санктпетербург – 2006.
4. <http://acm.tuit.uz>.
5. <http://algo.urgench-tuit.uz>.

Laboratoriya ishi №7

Mavzu: Ikki o'lchamli massivlar.

I. Variantlar

Ikki o'lchovli massivni e'lon qiling. Variantda ko'rsatilgan matritsalar ustida bajariladigan amallarning dasturini tuzing.

№	Variantlar
1	Uchga uch determantni hisoblang
2	A matritsaga B matritsani qo'shing
3	A matritsani transponirlang.
4	$N \times N$ birlik matritsa yarating
5	A matritsaning manfiy bo'lmagan elementlarini chop qiling.
6	A kvadrat matritsaning determantini hisoblovchi dastur tuzing
7	Bir o'lchovli massivni Pufakcha usulida saralang
8	A matritsaning har bir satr elementlarning maximumini chop qiling
9	A matritsaning bosh diagonalidan pastda yotgan elementlarini 0 (nol) bilan almashtiring
10	A kvadrat matritsaga teskari matritsani toping
11	A matritsaning elementlari yig'indisi eng katta bo'lgan satr elementlarini chop qiling
12	Bosh diagonalidan yuqori va past qismi simmetrik bo'gan matritsa hosil qiling
13	Elementlari soni tub bo'lmagan a massiv elementlaridan matritsa hosil qiling
14	O'lchamlari bir xil bo'lgan A matritsa bosh diagonal elementlarini B matritsa bosh diagonal elementlari bilan mos ravishda almashtiring
15	A va B matritsa barcha elementlari yig'indisi katta bo'lgan matritsa elementlarini chop qiling
16	$N \times N$ matritsani ilon izi shaklida chiqaring. Misol: $A[3,3] = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 7 & 8 & 9 \end{pmatrix}$
17	A matritsaning ikkinchi diagonalidan yuqorida yotgan

	elementlarining yig'indisini aniqlang
18	Berilgan n sonining kvadratigacha bo'lgan natural sonlarni NxN kvadrat matritsaga oxiridan boshlab ketma –ket joylashtiring.
19	Berilgan A matritsaning eng katta qiymati va eng kichik qiymati o'rinlarini almashtiring (agar ular soni bittadan ko'p bo'lsa birinchi uchragani bilan almashtiring)
20	A matritsaning ikki diagonalidan yuqorida joylashgan elementlari yig'indisini aniqlang

II. Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

A matritsaga B matritsani ko'paytiring.

Ishdan maqsad.

Matritsani matritsaga ko'paytirishni ikki o'lchovli massivlardan foydalanib dasturiy ta'minotni tuzish.

Nazariy qism. Ikki o'lchovli massivlarning ta'rifi, e'loni va qo'llanilishi.

C++ dasturlash tilida massivlar elementining turiga cheklovlar qo'yilmaydi, lekin bu turlar chekli o'lchamdagi ob'ektlarning turi bo'lishi kerak. Chunki kompilyator massivning xotiradan qancha joy egallashini hisoblay olishi kerak. Xususan, massiv komponentasi massiv bo'lishi mumkin natijada matritsa deb nomlanuvchi ikki o'lchamli massiv hosil bo'ladi. Ikki o'lchovli massivlar quyidagicha e'lon qilinadi:

$\langle \text{tur} \rangle \langle \text{nom} \rangle [\langle \text{uzunlik} \rangle] [\langle \text{uzunlik} \rangle]$

Misol: `int a[3][3]={ 1, 2, 3; 0, -1, 2; -2, 0, 1 }` ko'rinishi:

$$a[i], [j] = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & 2 \\ -2 & 0 & 1 \end{pmatrix}.$$

Berilgan masala uchun algoritm.

A matritsani B matritsaga ko'paytirish algoritmi. Matematikada berilgan A(m,n) matritsani B(n,k) matritsaga ko'paytirish imkoniyati mavjud. Ya'ni ikki matritsani bir biriga ko'paytirish vaqtida birinchi

matritsaning ustunlari soni ikkinchi matritsaning satrlari soniga teng bo'lishi kerak, aks xolda ushbu berilgan matritsalarini bir – biriga ko'paytirish mumkin emas.

Misol:

$$A[2,3] = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}. \quad B[3,3] = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}.$$

$C[2,3] = A[2,3] \times B[3,3]$. ya'ni

$$c[1,1] = a_{11} \cdot b_{11} + a_{12} \cdot b_{21} + a_{13} \cdot b_{31};$$

$$c[1,2] = a_{11} \cdot b_{12} + a_{12} \cdot b_{22} + a_{13} \cdot b_{32};$$

$$c[1,3] = a_{11} \cdot b_{13} + a_{12} \cdot b_{23} + a_{13} \cdot b_{33};$$

$$c[2,1] = a_{21} \cdot b_{11} + a_{22} \cdot b_{21} + a_{23} \cdot b_{31};$$

$$c[2,2] = a_{21} \cdot b_{12} + a_{22} \cdot b_{22} + a_{23} \cdot b_{32};$$

$$c[2,3] = a_{21} \cdot b_{13} + a_{22} \cdot b_{23} + a_{23} \cdot b_{33}.$$

Asosiy dastur matni.

```
[*] AxB.cpp
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int A[100][100], B[100][100], C[100][100];
5      int i, j, k, l, n, m;
6      cin >> n >> m >> k;
7      for (i=0; i<n; i++)
8          for (j=0; j<m; j++)
9              cin >> A[i][j];
10
11     for (i=0; i<m; i++)
12         for (j=0; j<k; j++)
13             cin >> B[i][j];
14     cout << endl << "natija: " << endl;
15     for (i=0; i<n; i++)
16         for (j=0; j<k; j++)
17         {
18             C[i][j] = 0;
19             for (l=0; l<m; l++)
20                 C[i][j] += A[i][l] * B[l][j];
21         }
22     for (i=0; i<n; i++) {
23         for (j=0; j<k; j++)
24             cout << C[i][j] << " ";
25         cout << endl;
26     }
27     return 0;
28 }
29
```

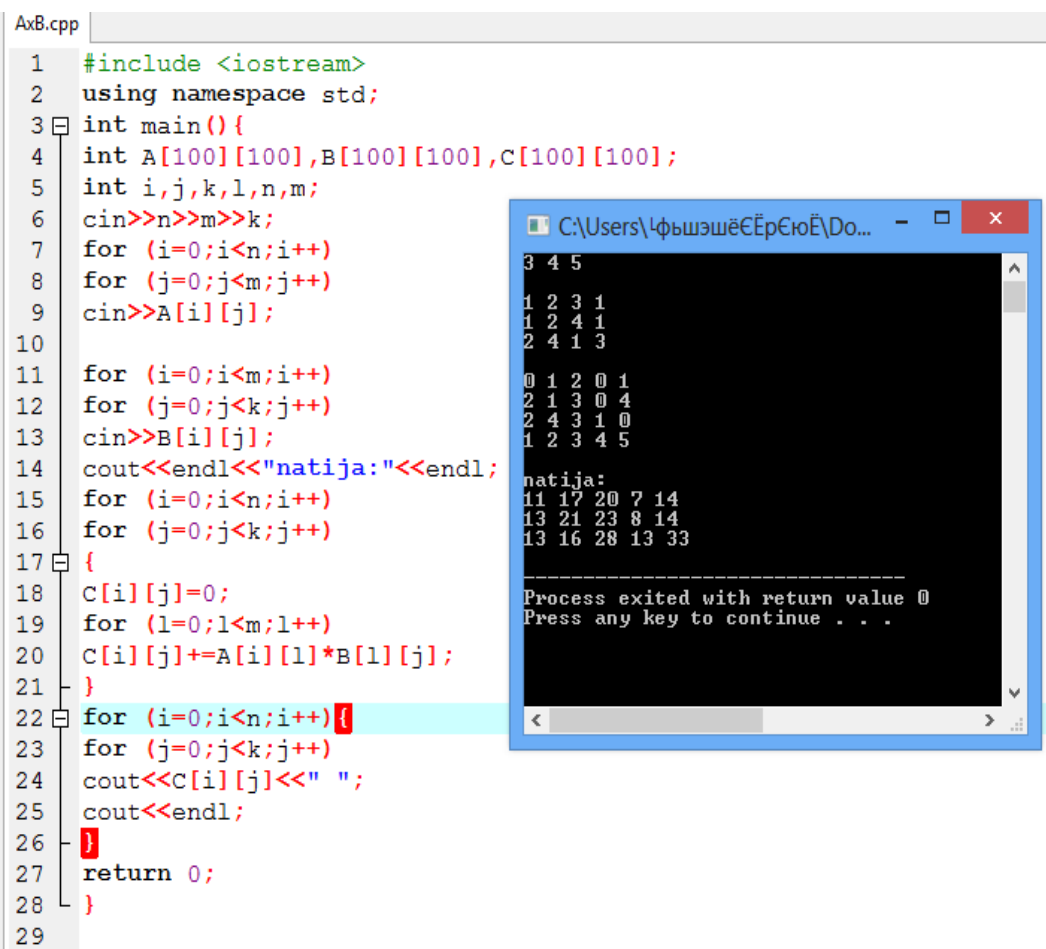
Dastur bajarilish natijalari tahlili.

Dasturning eng asosiy qismi:

```
for (i=0;i<n;i++)  
for (j=0;j<k;j++)  
{  
C[i][j]=0;  
for (l=0;l<m;l++)  
C[i][j]+=A[i][l]*B[l][j];  
}
```

Dasturning mana shu qismida berilgan A matritsani B matritsaga ko'paytirishdan hosil bo'lgan qiymatlarni natijaviy C matritsaga ta'minlagan.

Natijalar listingi.



```
AxB.cpp  
1 #include <iostream>  
2 using namespace std;  
3 int main() {  
4     int A[100][100], B[100][100], C[100][100];  
5     int i, j, k, l, n, m;  
6     cin >> n >> m >> k;  
7     for (i=0; i<n; i++)  
8         for (j=0; j<m; j++)  
9         cin >> A[i][j];  
10  
11     for (i=0; i<m; i++)  
12         for (j=0; j<k; j++)  
13         cin >> B[i][j];  
14     cout << endl << "natija:" << endl;  
15     for (i=0; i<n; i++)  
16         for (j=0; j<k; j++)  
17         {  
18             C[i][j]=0;  
19             for (l=0; l<m; l++)  
20                 C[i][j] += A[i][l] * B[l][j];  
21         }  
22     for (i=0; i<n; i++)  
23     for (j=0; j<k; j++)  
24     cout << C[i][j] << " ";  
25     cout << endl;  
26     return 0;  
27 }  
28  
29
```

```
3 4 5  
1 2 3 1  
1 2 4 1  
2 4 1 3  
0 1 2 0 1  
2 1 3 0 4  
2 4 3 1 0  
1 2 3 4 5  
natija:  
11 17 20 7 14  
13 21 23 8 14  
13 16 28 13 33  
-----  
Process exited with return value 0  
Press any key to continue . . .
```

Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. <http://acm.tuit.uz>.

Laboratoriya ishi №8

Mavzu: Ko'rsatkichlar.

I. Variantlar

Ko'rsatkichlar orqali o'zgaruvchilarni e'lon qiling. Variantda ko'rsatilgan misollar ustida bajariladigan amallarning dasturini tuzing.

№	Variantlar
1	n ta elementdan tashkil topgan massiv berilgan. Bu massivning juft o'rinlarida turgan elementlari yig'indisi hisoblansin. (Ko'rsatkichlardan foydalanib hisoblansin).
2	n ta elementdan tashkil topgan massiv berilgan. Bu massivning juft elementlari yig'indisi hisoblansin. (Ko'rsatkichlardan foydalanib hisoblansin).
3	n ta elementdan tashkil topgan massiv berilgan. Bu massivning elementlarini teskari tartibda chiqaring. (Ko'rsatkichlardan foydalaning).
4	n ta elementdan tashkil topgan massiv berilgan. Massivning eng katta elementi aniqlansin. (Ko'rsatkichlardan foydalaning).
5	n ta elementdan tashkil topgan massiv berilgan. Bu massivning eng kichik elementi aniqlansin. (Ko'rsatkichlardan foydalaning).
6	n ta elementdan tashkil topgan massiv berilgan. Bu massivning toq elementlari ko'paytmasi hisoblansin. (Ko'rsatkichlardan foydalaning)
7	n ta elementdan tashkil topgan massiv berilgan. Bu massivning juft o'ringa turgan juft elementlari soni aniqlansin. (Ko'rsatkichlardan foydalaning)
8	n ta elementdan tashkil topgan massiv berilgan. Eng katta 2 ta elementi aniqlansin. (Ko'rsatkichlardan foydalaning)
9	n ta elementdan tashkil topgan massiv berilgan. Bu massivning 2 ta eng kichik elementi aniqlansin. (Ko'rsatkichlardan foydalaning)
10	n ta elementdan tashkil topgan massiv berilgan. Bu massivning 3 ta eng katta elementi aniqlansin (Ko'rsatkichlardan foydalaning)
11	n ta elementdan tashkil topgan massiv berilgan. Bu massivning eng kichik 3 ta elementi aniqlansin. (Ko'rsatkichlardan

	foydalaning)
12	n ta elementdan tashkil topgan massiv berilgan.Bu massivning eng katta va eng kichik elementlari yig'indisi hisoblansin. (Ko'rsatkichlardan foydalaning)
13	n ta elementdan tashkil topgan massiv berilgan.Bu massivning eng katta va eng kichik elementlari ko'paytmasi hisoblansin. (Ko'rsatkichlardan foydalaning)
14	n ta elementdan tashkil topgan massiv berilgan.Bu massivning eng katta va eng kichik elementlari ayirmasi hisoblansin. (Ko'rsatkichlardan foydalaning)
15	n ta elementdan tashkil topgan massiv berilgan.Bu massivning tub elementlari soni aniqlansin. (Ko'rsatkichlardan foydalaning)
16	n ta elementdan tashkil topgan massiv berilgan.Bu massivning eng kichik elementi joylashgan manzilni chiqaring. (Ko'rsatkichlardan foydalaning)
17	n ta elementdan tashkil topgan massiv berilgan.Bu massivning elementlari joylashgan adreslarini chiqring. (Ko'rsatkichlardan foydalaning)
18	n ta elementdan tashkil topgan massiv berilgan. Eng katta 2 ta elementi joylashgan adreslar aniqlansin. (Ko'rsatkichlardan foydalaning)
19	n ta elementdan tashkil topgan massiv berilgan.Bu massivning 2 ta eng kichik elementi elementi joylashgan adreslar aniqlansin. (Ko'rsatkichlardan foydalaning)
20	n ta elementdan tashkil topgan massiv berilgan.Bu massivning 3 ta eng katta elementi joylashgan adreslar aniqlansin (Ko'rsatkichlardan foydalaning)

II.Uslubiy ko'rsatmalar

Masalaning qo'yilishi.

n ta elementdan tashkil topgan massiv berilgan.Bu massivning toq o'rinda turgan toq elementlari joylashgan adreslarni chiqaring (Ko'rsatkichlardan foydalaning).

Ishdan maqsad.

Ko'rsatkichlardan foydalanib berilgan masalani yechish.

Nazariy qism.

Dastur matnida o'zgaruvchi e'lon qilinganda, kompilyator o'zgaruvchiga xotiradan joy ajratadi. Boshqacha aytganda, programma kodi xotiraga yuklanganda berilganlar uchun, ular joylashadigan segmentning boshiga nisbatan siljishini, ya'ni nisbiy adresini aniqlaydi va obyekt kod hosil qilishda o'zgaruvchi uchragan joyga uning adresini joylashtiradi. Umuman olganda, programmadagi o'zgarmaslar, o'zgaruvchilar, funksiyalar va sinf obyektlar adreslarini xotiraning alohida joyida saqlash va ular ustidan amallar bajarish mumkin. Qiymatlari adres bo'lgan o'zgaruvchilarga ko'rsatkich o'zgaruvchilar deyiladi.

Ko'rsatkich uch xil turda bo'lishi mumkin:

- birorta obyektga, xususan o'zgaruvchiga ko'rsatkich;
- funksiyaga ko'rsatkich;
- void ko'rsatkich.

Ko'rsatkichning bu xususiyatlari uning qabul qilishi mumkin bo'lgan qiymatlarida farqlanadi. Ko'rsatkich albatta birorta turga bog'langan bo'lishi kerak, ya'ni u ko'rsatgan adresda qandaydir qiymat joylanishi mumkin va bu qiymatning xotirada qancha joy egallashi oldindan ma'lum bo'lishi shart.

Funksiyaga ko'rsatkich. Funksiyaga ko'rsatkich programma joylashgan xotiradagi funksiya kodining boshlang'ich adresini ko'rsatadi, ya'ni funksiya chaqirilganda boshqaruv ayni shu adresga uzatiladi. Ko'rsatkich orqali funktsiyani oddiy yoki vositali chaqirish amalga oshirish mumkin. Bunda funksiya uning nomi bo'yicha emas, balki funksiya ko'rsatuvchi o'zgaruvchi orqali chaqiriladi. Funktsiyani boshqa funksiya argument sifatida uzatish ham funksiya ko'rsatkichi orqali bajariladi. Funksiyaga ko'rsatkichning yozilish sintaksisi quyidagicha:

<tur> (* <nom>) (<parametrlar ro'yxati>);

Bunda <tur>- funksiya qaytaruvchi qiymat turi; *<nom> - ko'rsatkich o'zgaruvchining nomi; <parametrlar ro'yxati> - funksiya parametrlarining yoki ularning turlarining ro'yxati.

Obyektga ko'rsatkich. Biror obyektga ko'rsatkich (shu jumladan o'zgaruvchiga). Bunday ko'rsatkichda ma'lum turdagi (tayanch yoki hosilaviy turdagi) berilganlarning xotiradagi adresi joylashadi. Obyektga ko'rsatkich quyidagicha e'lon qilinadi:

<tur> *<nom>;

Bu yerda <tur> - ko'rsatkich aniqlaydigan adresdagi qiymatning turi, <nom> - obyekt nomi (identifikator). Agar bir turda bir nechta ko'rsatkichlar e'lon qilinadigan bo'lsa, har bir ko'rsatkich uchun '*' belgisi qo'yilishi shart:

int *i, j,*k;

float x,*y,*z;

Keltirilgan misolda i va k - butun turdagi ko'rsatkichlar va j - butun turdagi o'zgaruvchi, ikkinchi operatorida x - haqiqiy o'zgaruvchi va y,z - haqiqiy turdagi ko'rsatkichlar e'lon qilingan.

void ko'rsatkich. Bu ko'rsatkich obyekt turi oldindan noma'lum bo'lganda ishlatiladi. void ko'rsatkichining muhim afzalliklaridan biri - unga har qanday turdagi ko'rsatkich qiymatini yuklash mumkinligidir. void ko'rsatkich adresidagi qiymatni ishlatishdan oldin, uni aniq bir turga oshkor ravishda keltirish kerak bo'ladi. void ko'rsatkichni e'lon qilish quyidagicha bo'ladi:

void *<nom>;

Berilgan masala uchun algoritm.

n soni va massiv elementlarini kiritib olamiz. Massivning boshidan boshlab tekshiramiz: Agar element turgan index va element toq bo'lsa bu element joylashgan adresni chiqaramiz.

Asosiy dastur matni.

```
#include<iostream>
using namespace std;
int main () {
    int n,a[100],*p; //o'zgaruvchilarni e'lon qilish
    cin >> n;
    //n soni va
```

```

    for(int i = 1; i <= n; i++)cin >> *(a + i);
//massiv elementlarini kiritish
for(int i=1;i<=n;i++)//massivning birinchi
//hadidan boshlab
{
if(i%2==1&&*(a + i)%2==1) //shartni tekshiramiz
{
p=&*(a+i); //shart bajarilsa p ga element
//joylashgan adresni olamiz
cout << p << " ";//va p ni chiqaramiz
} }}

```

Dastur bajarilish natijalari tahlili.

Masalan:

4

2 3 5 7

2 – indeksi toq son lekin elementning o'zi juft son.

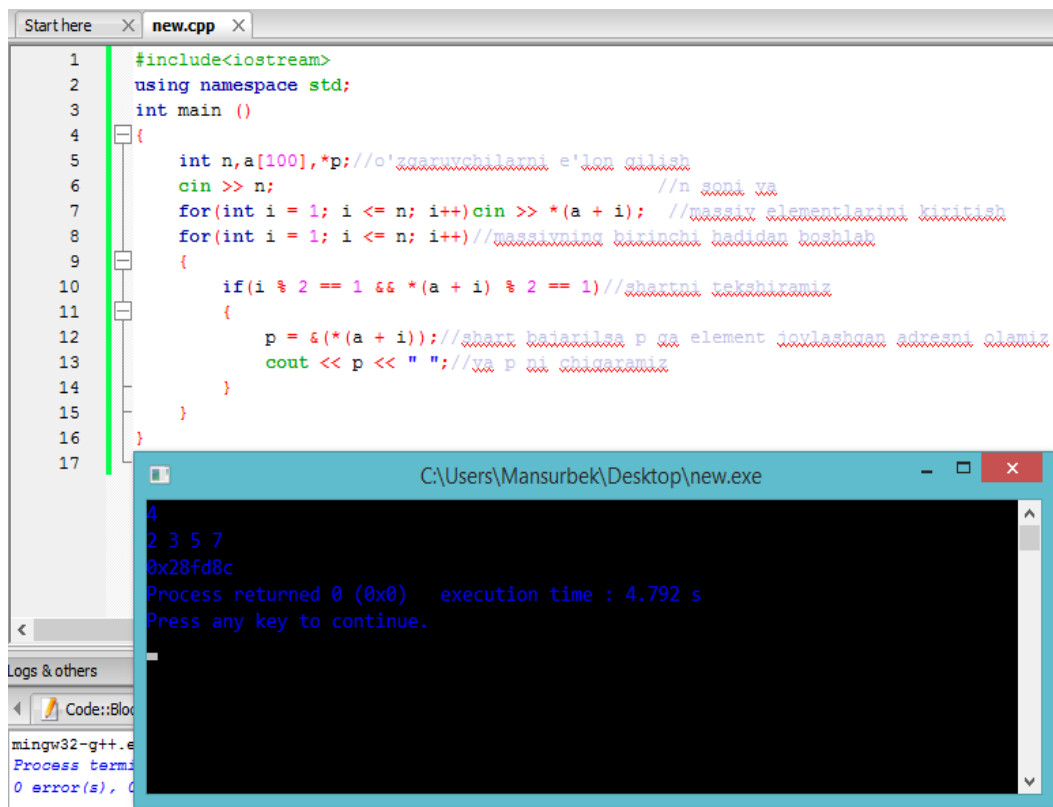
3 – indeksi juft element toq son.

5 – indeksi toq son va elementning o'zi ham toq son.

7 – indeksi juft son va element toq son.

Bu sonlardan faqat bittasi – 5 shartni qanoatlantiradi. Faqat shu elementning adresini chiqaramiz.

Natijalar listingi.



The image shows a C++ program in a code editor and its execution in a console window. The code in the editor is as follows:

```
1 #include<iostream>
2 using namespace std;
3 int main ()
4 {
5     int n,a[100],*p;//o'zgaruvchilarni e'lon qilish
6     cin >> n; //n soni va
7     for(int i = 1; i <= n; i++)cin >> *(a + i); //massiv elementlarini kiritish
8     for(int i = 1; i <= n; i++)//massivning birinchi hadidan boshlab
9     {
10         if(i % 2 == 1 && *(a + i) % 2 == 1)//shartni tekshiramiz
11         {
12             p = &*(a + i);//shart bajarilsa p ga element joylashgan adresni olamiz
13             cout << p << " "; //va p ni chiqaramiz
14         }
15     }
16 }
17
```

The console window shows the output of the program:

```
C:\Users\Mansurbek\Desktop\new.exe
4
2 3 5 7
0x28fd8c
Process returned 0 (0x0)   execution time : 4.792 s
Press any key to continue.
```

Foydalanilgan adabiyotlar.

1. C++ tilida programmalash asoslari, Sh. F. Madrahimov, S. M. Gaynazarov, Toshkent – 2009.
2. Решение сложных и олимпиадных задач по программированию, Долинский М. С., Санктпетербург – 2006.
3. <http://acm.tuit.uz>.
4. <http://algo.urgench-tuit.uz>.
5. <http://cplusplus.com>.