

**O‘ZBEKISTON RESPUBLIKASI RAQAMLI  
TEXNALOGIYALAR VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT  
AXBOROT TEXNOLOGIYALARI UNIVERSITETI  
SAMARQAND FILIALI**

**DASTIRIY INJINIRINGI KAFEDRASI**

**“Dasturlash 2” fanidan**

**2-AMALIY ISHI**

AKT-S20-03 guruh

**Topshirdi:** Saidova R.

**Tekshirdi:** Kayumov A.

**SAMARQAND-2024**

## Masalaning qo'yilishi.

Quyida berilgan tarmoqlanuvchi algoritmlarning dasturini tuzing.

$$Y = \begin{cases} \sqrt{k}, & \text{agar } sink \leq 0.2 \\ \frac{1}{\sqrt{k}}, & \text{agar } sink > 0.2 \end{cases}$$

$$\text{Bunda } k=7 \quad Y=0.378 \quad k=1.4 \quad Y=0.845$$

## 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:

$$\textit{If} (< \textit{shart} >) < \textit{operator} >;$$

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).

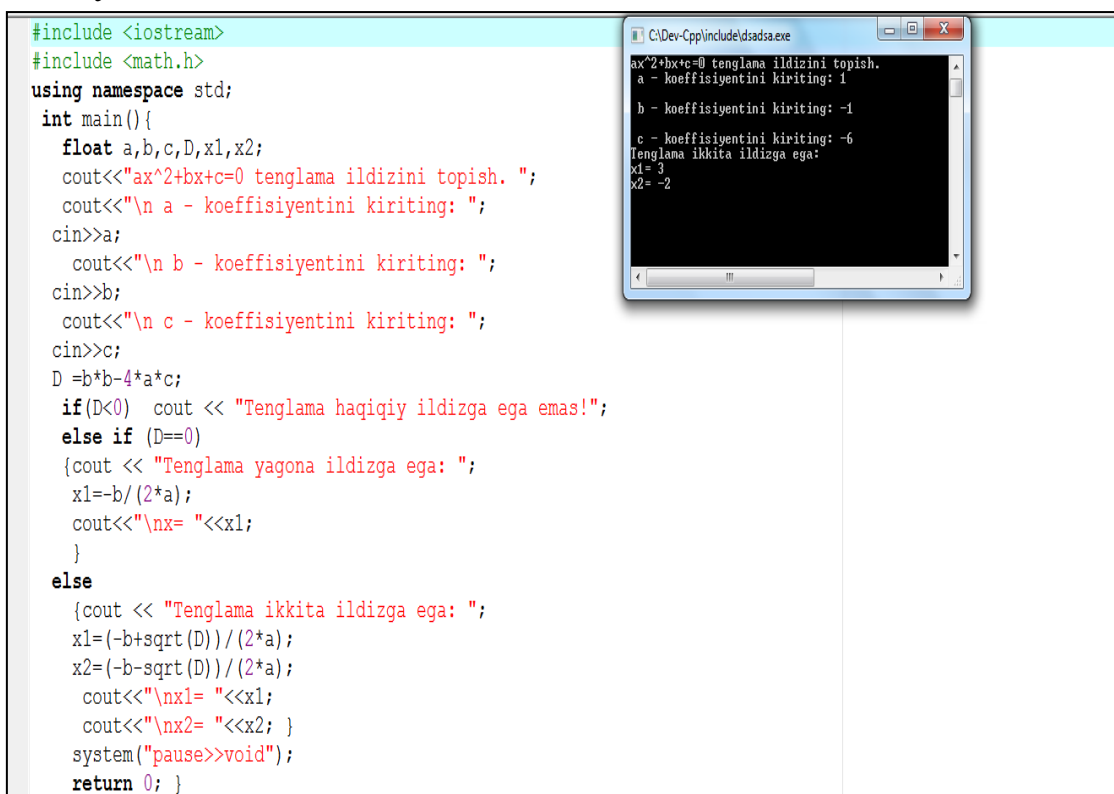
C++ tilining qurilmalari operatorlarni blok ko'rinishida tashkil qilishga imkon beradi. Blok - '{' va '}' belgi oralig'iga olingan operatorlar ketma-ketligi bo'lib, u kompilyator tomonidan yaxlit bir operator deb qabul qilinadi. Blok ichida e'lon operatorlari ham bo'lishi mumkin va ularda e'lon qilingan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

Bloktan 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 diskreminantni 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 - koeffisiyentini kiriting: ";
    cin>>a;
    cout<<"\n b - koeffisiyentini kiriting: ";
    cin>>b;
    cout<<"\n c - koeffisiyentini 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<<"\nx= ";<<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; }
```

The screenshot shows a C++ IDE window titled 'C:\Dev-Cpp\include\d\sd\sa.exe'. The main window displays the C++ code for solving a quadratic equation. An inset window shows the program's output: 'ax^2+bx+c=0 tenglama ildizini topish.', 'a - koeffisiyentini kiriting: 1', 'b - koeffisiyentini kiriting: -1', 'c - koeffisiyentini kiriting: -6', 'Tenglama ikkita ildizga ega:', 'x1= 3', and 'x2= -2'.

Dastur bajarilganda, birinchi navbatda tenglama koeffisiyentlari - a,b,c o’zgaruvchilar qiymatlari kiritiladi, keyin diskreminant - 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 x1 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 x1 va x2 o'zgaruvchilar qiymatlari chop etiladi. Shu bilan shart operatoridan chiqiladi va asosiy funksiyaning return ko'rsatmasini bajarish orqali dastur o'z ishini tugatadi.

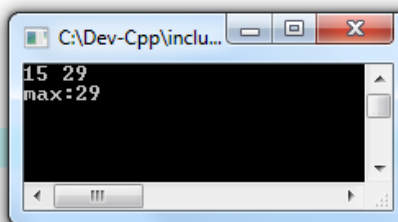
**?: shart amali.** Agar tekshirilayotgan shart nisbatan sodda bo'lsa, shart amalining "?:" ko'rinishini ishlatish mumkin:

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

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 ifodalari 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.**

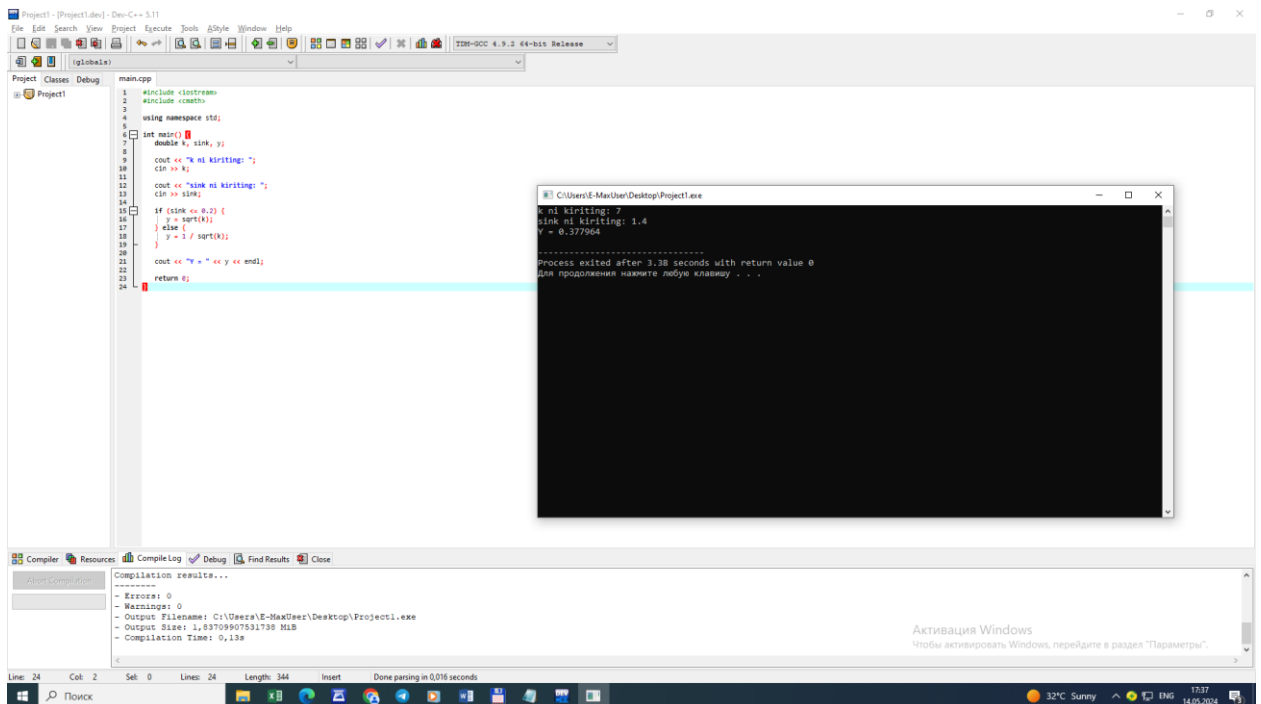
Yuqorida berilgan masalani hisoblash uchun birinchi navbatda uni matematik ko'rinishdan C++ dasturlash muhitiga o'tkazishda *if* shart operatoridan foydalanamiz.

$$Y = \begin{cases} \sqrt{k}, & \text{agar } sink \leq 0.2 \\ \frac{1}{\sqrt{k}}, & \text{agar } sink > 0.2 \end{cases}$$

### **Asosiy dastur matni:**

```
#include <iostream>
#include <cmath>
```

```
using namespace std;
int main() {
    double k, sink, y;
    cout << "k ni kiriting: ";
    cin >> k;
    cout << "sink ni kiriting: ";
    cin >> sink;
    if (sink <= 0.2) {
        y = sqrt(k);
    } else {
        y = 1 / sqrt(k);
    }
    cout << "Y = " << y << endl;
    return 0;
}
```



The screenshot displays a C++ IDE with the following components:

- Source Code (main.cpp):**

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main() {
7     double k, sink, y;
8
9     cout << "k ni kiriting: ";
10    cin >> k;
11    cout << "sink ni kiriting: ";
12    cin >> sink;
13
14    if (sink <= 0.2) {
15        y = sqrt(k);
16    } else {
17        y = 1 / sqrt(k);
18    }
19
20    cout << "Y = " << y << endl;
21    return 0;
22 }
```
- Output Window (C:\Users\E-Max\Desktop\Project1.exe):**

```
k ni kiriting: 7
sink ni kiriting: 1.4
Y = 0.377964
.....
Process exited after 3.38 seconds with return value 0
для продолжения нажмите любую клавишу . . .
```
- Compiler Output:**

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\E-Max\Desktop\Project1.exe
- Output Size: 1,83709907531730 Mib
- Compilation Time: 0,13s
```

The Windows taskbar at the bottom shows the date and time as 14.09.2024, 12:37, and the weather as 32°C Sunny.

