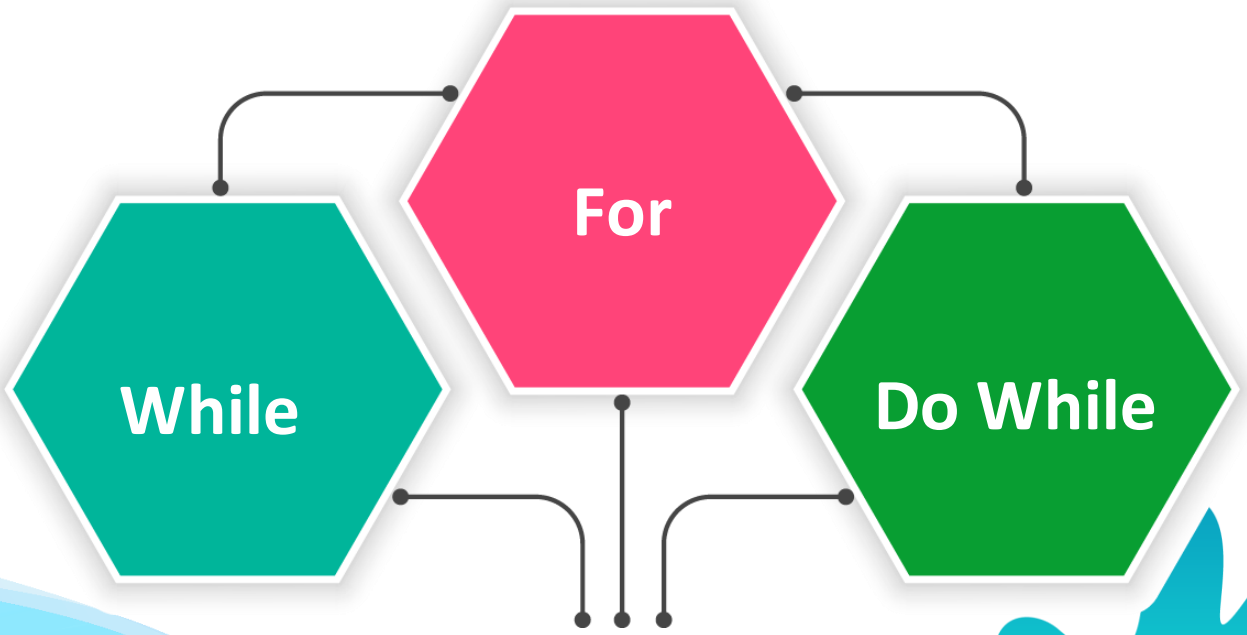
C++ dasturlash tilida takrorlanuvchi algoritimlar

Takrorlanuvchi algoritimlani dasturlash

Dastur kodinining ***bir qismining*** takroriy bajarilishi tsikl hisoblanadi. Tsikl ma'lum bir sharoitlarda dastur kodining bir qismini bir necha marta bajarishda ishlatiladi. Agar shart to'g'ri bo'lsa, tsikl davom etadi. Aks holda u to'xtatiladi. Agar shart har doim to'g'ri bo'lsa, bunday tsikl ***cheksiz*** tsikl deb ataladi.

C ++ da 3 xil tsikl operatorlari mavjud:



for operatori

For tsikli bu bir marta bajarilishi kerak bo'lgan shartlarni samarali yozishga imkon beruvchi takrorlashni boshqarish tuzilmasidir.

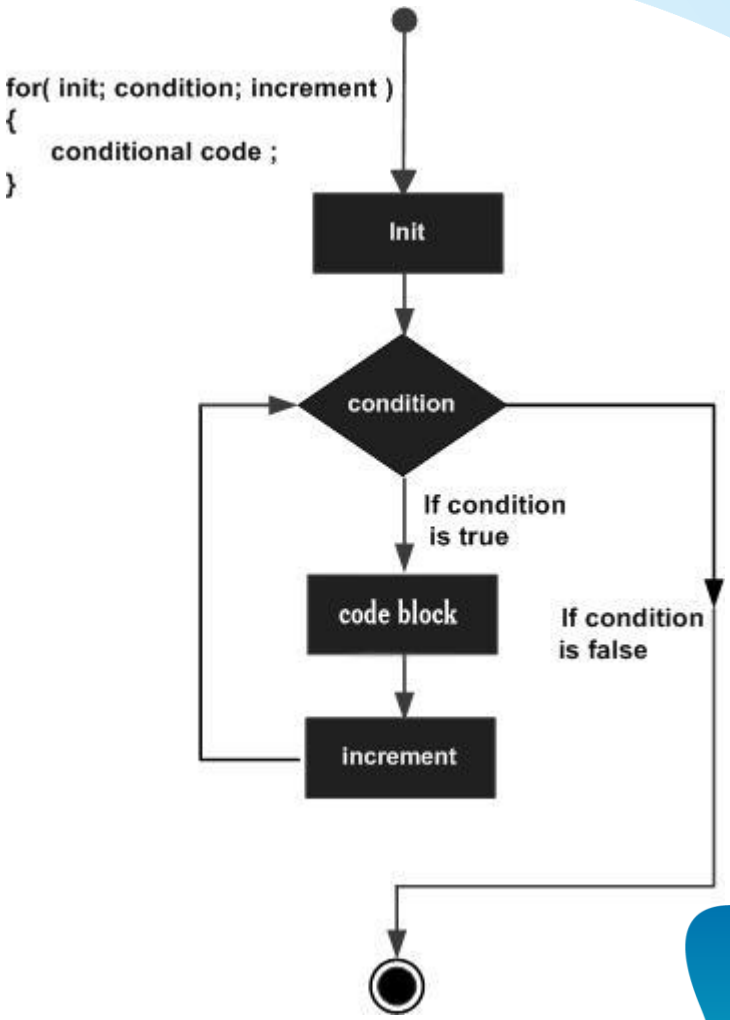
Ushbu tsiklning tuzilishi quyidagicha.

```
for (int i = 0; i < length; i++)
{
  Natija;
}
```

Bu yerda `int i = 0;` tsiklning boshlanishi. `i` o'zgaruvchining qiymatiga istalgan raqamni kiritishingiz mumkin.

`i < length;` bu tsiklning chegarasi. Ya'ni, kiritilgan raqam ushbu qiymatdan past bo'lgan diapazondagi qiymatlarni hisoblaydi.

`i++` бу C ++ tsiklning boshlang'ich raqamiga avtomatik ravishda birini qo'shadigan funktsiya.



for operatoridan foydalanish

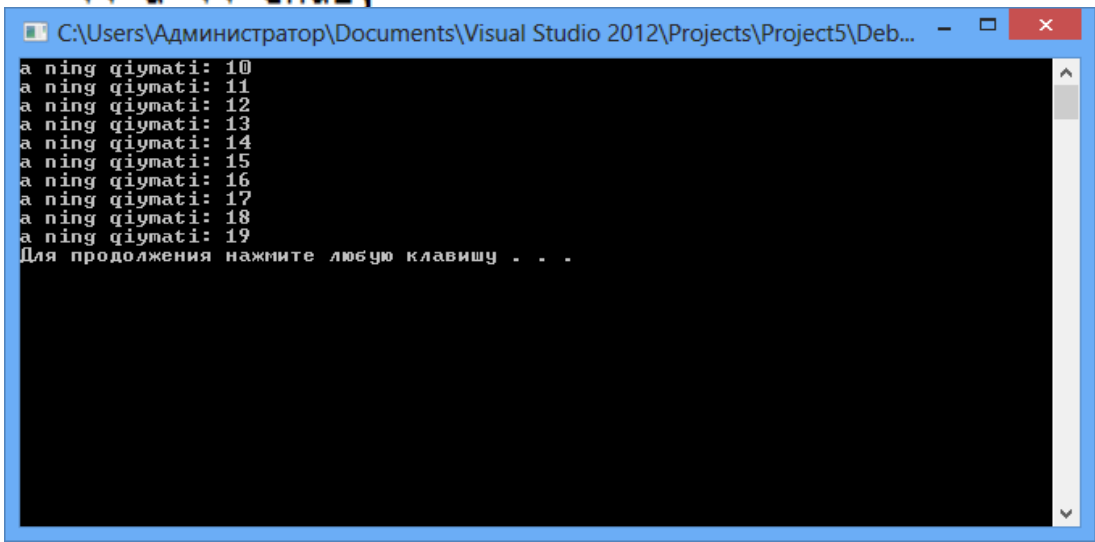
Tsiklni o'rganish uchun eng oson misol bu 1 dan N gacha bo'lgan (1 + 2 + 3 +… + N) musbat butun sonlarni yigindisini topish dasturini yaratish. Bu yigindini takrorlanish jarayoni orqali hisoblash uchun 1 dan N gacha sonlarni birma-bir qo’shib chiqish mumkin. Yigindining dastlabki qiymatini 0 ga tenglaymiz. Tsiklning har bir qadamida quyidagi amallar bajariladi:

s=0;
1-qadam. s=s+1=0+1=1;
2-qadam. s=s+2=1+2=3;
3-qadam. s=s+3=3+3=6;
4-qadam. s=s+4=6+4=10;
5-qadam. s=s+5=10+5=15;
……
i-qadam. s=s+i;
……
N-qadam. s=s+n;

Har bir qadamda bir xil harakatlar amalga oshiriladi, ya'ni to'plamning yangi qiymatini yaratish uchun oldingi bosqich qiymatiga keyingi natural raqam qo'shiladi.

for operatoriga misol

```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
  for( int a = 10; a < 20; a = a + 1 ) {
      cout << "a ning qiymati: " << a << endl;
   }
system("PAUSE");
return 0;
}
```

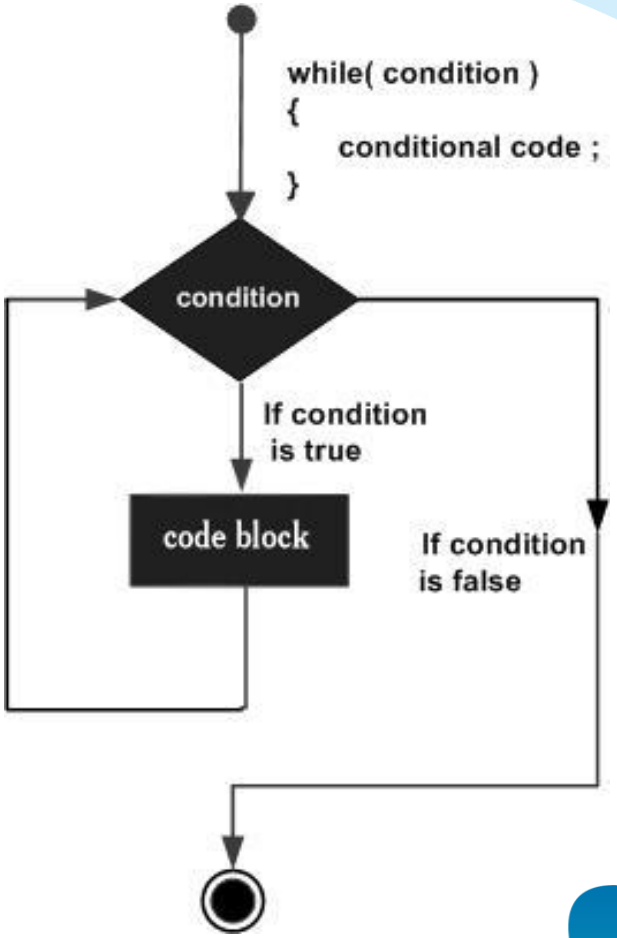


while operatori

while tsiklida shart oldindan qo’yiladi. Agar shart to'g'ri(***true***) bo'lsa tsikl tanasi bajariladi. Aks holda, tsikl to'xtaydi va keyingi bosqichga o'tadi.

Ushbu tsiklning tuzilishi quyidagicha.

```
while (true)
 {
  tsikl tanasi;
 }
```

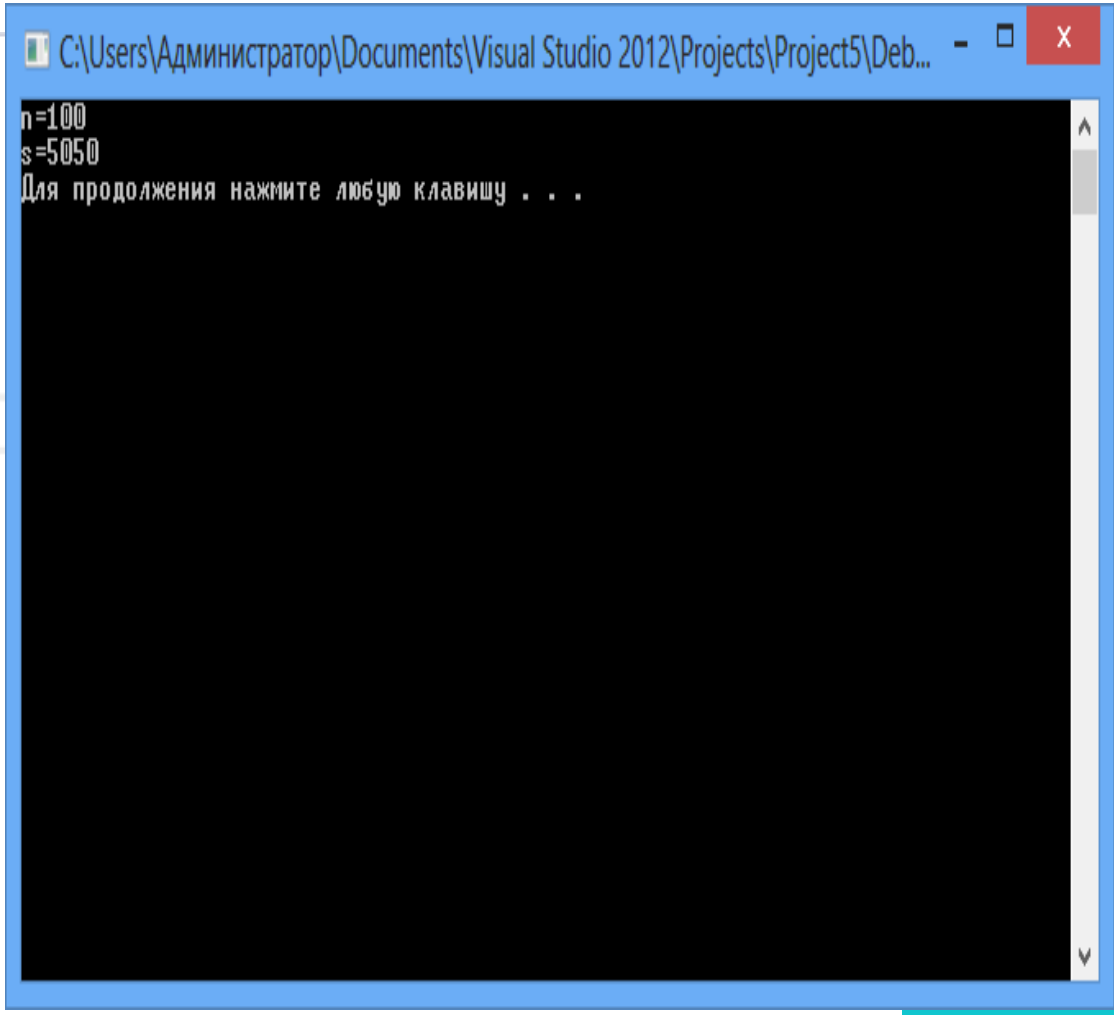


while ва **for** tsikllari orasidagi farq shundaki, **while** operatori odatda biron bir shart bajarilmaguncha davom etuvchi tsikldur. **for** operatori esa oldindan belgilangan sonli takrorlanishlar uchun ishlatiladi.

while operatoriga misol

1 dan ***n*** gacha sonlar yig'indisi topilsin.

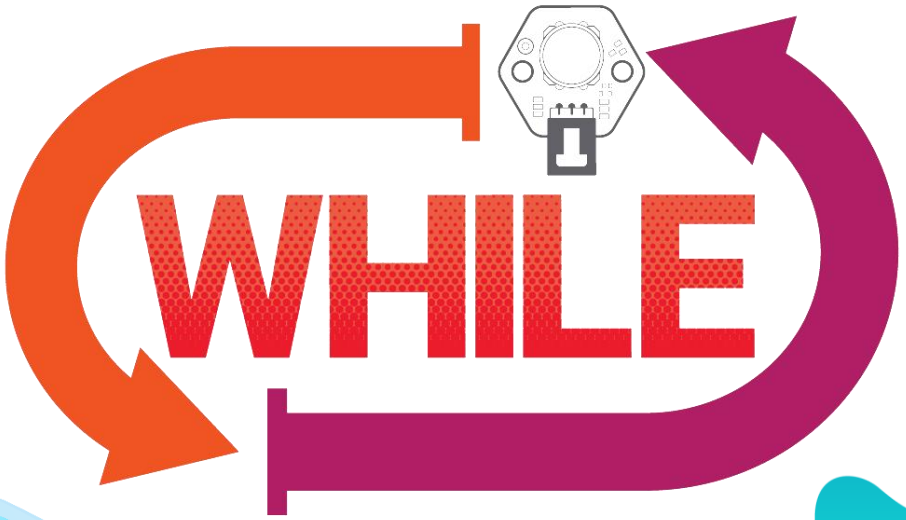
```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
int s = 0, i = 1, n;
cout << "n=";
cin >> n;
while (i <= n) {
s += i;
i++;
}
cout << "s=" << s << "\n";
system("PAUSE");
return 0;
}
```



Dastur tarifi

Dastur kodini boshidan ko'rib chiqaylik. Bizga uchta o'zgaruvchi kerak. Birinchi o'zgaruvchi **n** soni, ikkinchi o'zgaruvchi - hisoblash uchun ishlatiladigan **i** o'zgaruvchisi, uchinchi o'zgaruvchi - to'plamning qiymatini saqlash uchun ishlatiladigan **S** o'zgaruvchidir.

Tsiklni boshlashdan oldin biz **S** to'plamning qiymatini no’lga tenglashtiramiz. O'zgaruvchining standart qiymatini **1** ga o'rnatdik, chunki biz to'plamga **1** raqamini qo'shishimiz kerak. Agar **i≤ n** shart qanoatlantirilsa, u holda **i** ni to'plamga qo'shamiz (**s + = i** bu **s = s + i** degani) va **i** qiymatini bittaga ko'paytiramiz (**i ++** bu **i=i+1** degani).



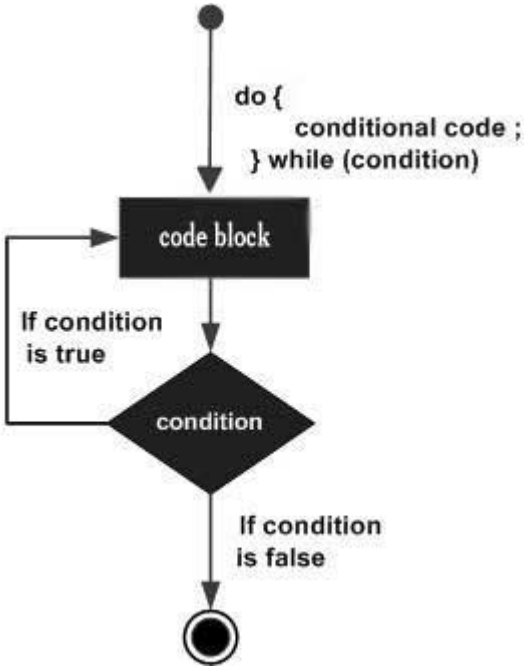
do while operatori

do while tsikli **while** tsikliga o'xshash, farqi shart tsikl oxirida tekshiriladi va shart bajariladimi yo’ki yo'qmi, kamida bir marta tsikl bajariladi.

Ushbu tsiklda o'zgaruvchining qiymati qanday bo'lishidan qat'iy nazar tsikl bir marta aylanadi.

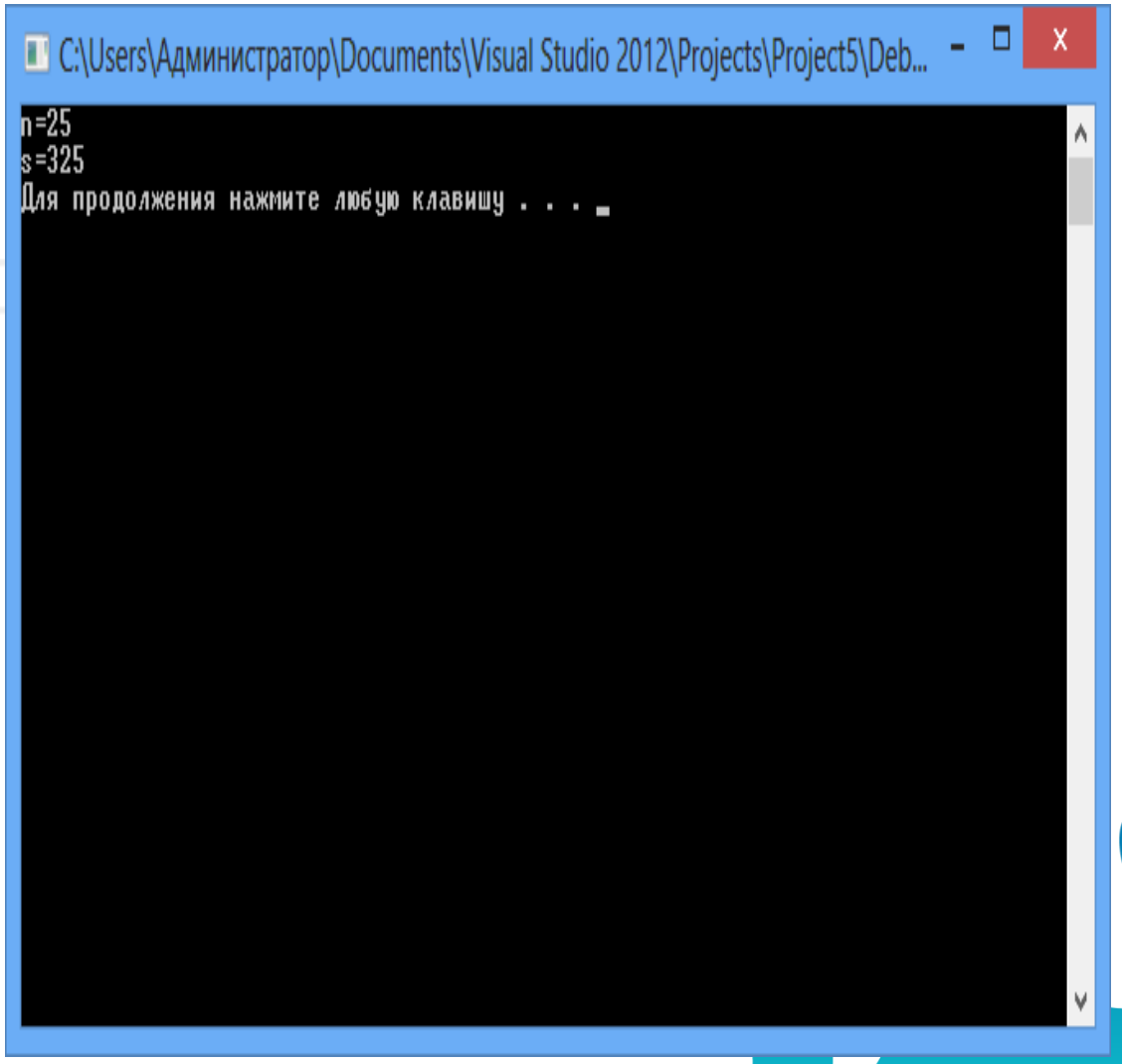
Misol uchun:

```
do {
cout << "n=";
cin >> n;
}
while (n < 1);
```



do while operatoriga misol

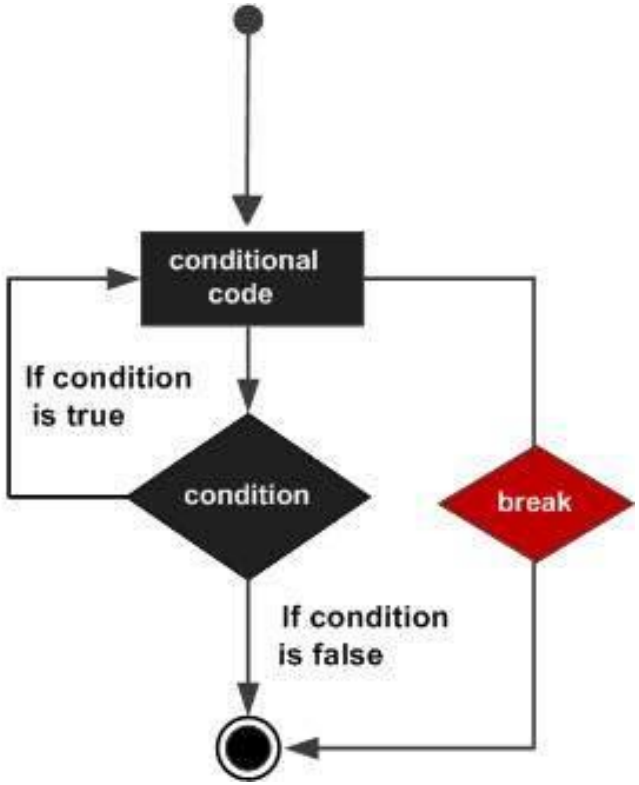
```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{

int s = 0, i = 1, n;
cout<<"n=";
cin>>n;
do {
s += i;
i++;
}
while (i <= n);
cout<<"s=" << s << "\n";

system("PAUSE");
return 0;
}
```



break operatori

break operatori tsiklni rost qiymatni bajaradimi yoki yo'qligidan qat'i nazar to'xtatish uchun ishlatiladi. Quyidagi misolda ko'rishimiz mumkin: **X** ning qiymati **1** dan **100** gacha **0,01** qadam bilan oshirish dasturi tuzilsin.

```
double x = 1;
for (; ;) {
if (x > 100.000001)
break;
Cout << x << " ";
x += 0.01;
}
```



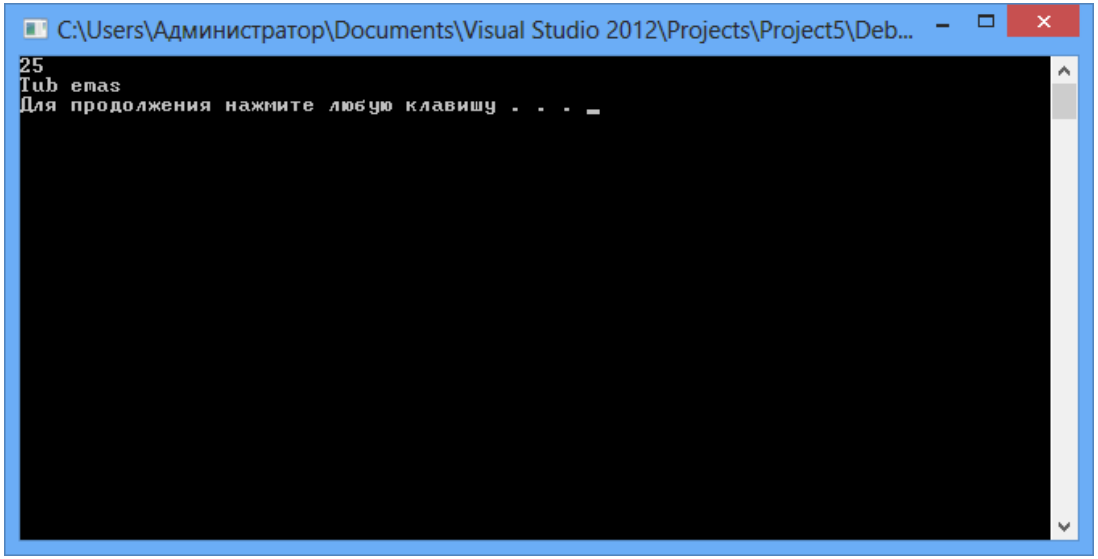
break operatoriga misol

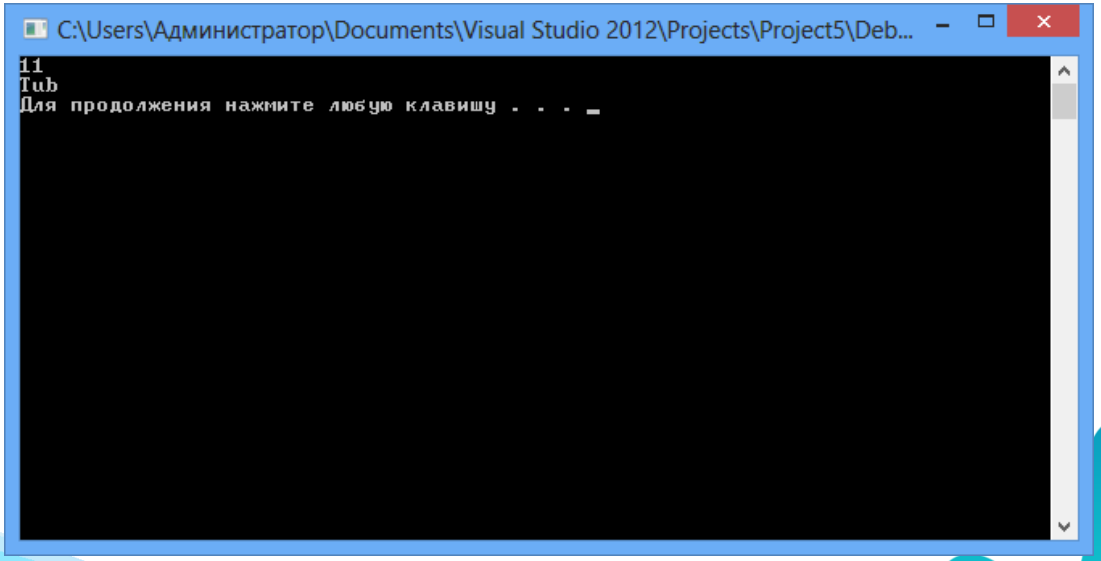
```
int main()
{

int n;
cin>>n;
bool is_prime = true;
for (int i = 2; i*i <= n; i++) {
if (n % i==0) {
is_prime = false;
break;
}
}
if (n==1)
{
is_prime = false;
}
if (is_prime)
{
cout<<"Tub" << "\n";
}
else
{
cout<<"Tub emas" << "\n";
}
system("PAUSE");
return 0;
}
```

Break Operatordan foydalanishning misollaridan biri berilgan sonning **tub** yoki **tub emasligini** aniqlaydigan dastur yo'zishdir.

Raqam ildiz ekanligini aniqlash uchun uning 2 dan $\sqrt{n}$ gacha bo'lgan raqamlarga bo'linishini tekshiramiz. Agar ulardan biri qo'ldiqsiz bo'linsa, demak bu tub son emas.

Natijasi

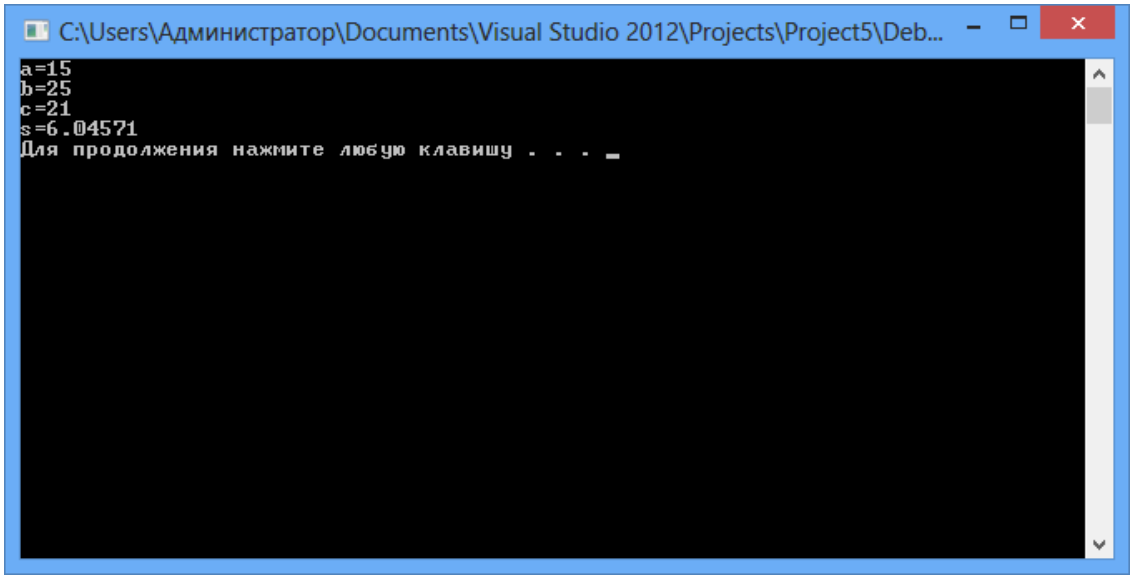

C:\Users\Администратор\Documents\Visual Studio 2012\Projects\Project5\Deb...
25
Tub emas
Для продолжения нажмите любую клавишу . . .


C:\Users\Администратор\Documents\Visual Studio 2012\Projects\Project5\Deb...
11
Tub
Для продолжения нажмите любую клавишу . . .

C ++ dasturlash tilidagi iterativ operatorlarning misollari

1-misol. Yeg’indini hisoblash.

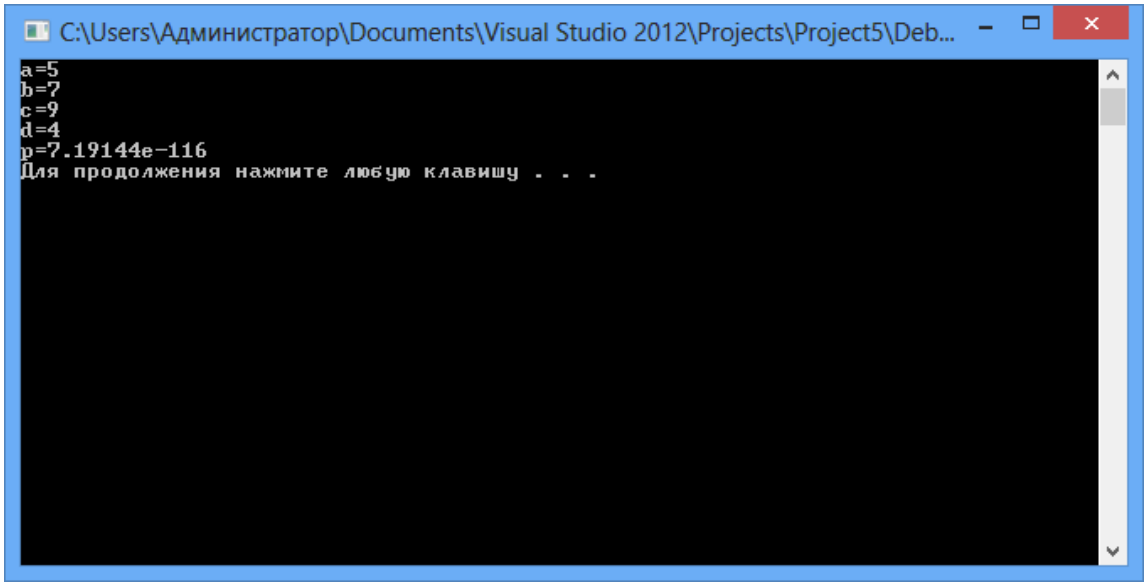
```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
int a,b,c,m;
cout<<"a=";
cin>>a;
cout<<"b=";
cin>>b;
cout<<"c=";
cin>>c;
double s=0;
for(m=1;m<=19;m++){
s=s+(3*pow(m,3.0)+4*a*m+5)/(b*pow(m
,3.0)+log(c*m-3));
}
cout<<"s="<<s<<endl;
system("PAUSE");
return 0;
}
```



$$S = \sum_{m=1}^{19} \frac{3m^3 + 4am + 5}{bm^3 + \ln(cm - 3)}$$

2-misol. Ko’paytmani hisoblash.

```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
int a,b,c,d,k;
cout<<"a=";
cin>>a;
cout<<"b=";
cin>>b;
cout<<"c=";
cin>>c;
cout<<"d=";
cin>>d;
double p=1;
for(k=1;k<=46;k++){
p *= a * k/(d * pow(k, 3.0)+7 * c *
k + 5*d);
}
cout<<"p="<<p<<endl;
system("PAUSE");
return 0;
}
```



$$P = \prod_{k=1}^{46} \frac{ak}{bk^3 + 7ck + 5d}.$$

Savol va javoblar