

```
if (condition) {
    statements
}
```

1

```
if (7 > 4) {
    cout << "Так";
}
```

2

```
// Виходи "Так"
```

```
if ( 10 == 10 ) {
    cout << "Так";
}
```

4

```
// Виходи "Так"
```

```
if (10 != 10) {
    cout << "Так";
}
```

5

```
int a = 55;
int b = 33;
if (a > b) {
    cout << "a is greater than b";
}
```

6

```
// Outputs "a is greater than b"
```

```
if (condition) {
    //statements
}
else {
    //statements
}
```

7

```
int mark = 90;
```

8

```
if (mark < 50) {
    cout << "You failed." << endl;
}
else {
    cout << "You passed." << endl;
}
```

```
// Outputs "You passed."
```

```
int mark = 100;
```

9

```
if (mark >= 50) {
    cout << "You passed." << endl;
    if (mark == 100) {
        cout << "Perfect!" << endl;
    }
}
else {
    cout << "You failed." << endl;
}
```

```
/*Outputs
You passed.
Perfect!
*/
```

Operator	Description	Example
>=	Greater than or equal to	7 >= 4 True
<=	Less than or equal to	7 <= 4 False
==	Equal to	7 == 4 False
!=	Not equal to	7 != 4 True

3

```
int age = 18;
if (age > 14) {
    if (age >= 18) {
        cout << "Adult";
    }
    else {
        cout << "Teenager";
    }
}
else {
    if (age > 0) {
        cout << "Child";
    }
    else {
        cout << "Something's wrong";
    }
}
```

10

```
int a = 10;
if (a > 4)
    cout << "Yes";
else
    cout << "No";
```

11

```
int mark = 90;
if (mark < 50) {
    cout << "You failed." << endl;
    cout << "Sorry" << endl;
}
else {
    cout << "Congratulations!" << endl;
    cout << "You passed." << endl;
    cout << "You are awesome!" << endl;
}

/* Outputs
Congratulations!
You passed.
You are awesome!
*/
```

12

```
while (condition) {
    statement(s);
}
```

13

```
int num = 1;
while (num < 6) {
    cout << "Number: " << num << endl;
    num = num + 1;
}

/* Outputs
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
*/
```

14

```
int num = 1;
while (num < 6) {
    cout << "Number: " << num << endl;
    num = num + 3;
}

/* Outputs
Number: 1
Number: 4
*/
```

15

```
int num = 1;
while (num < 6) {
    cout << "Number: " << num << endl;
    num++;
}

/* Outputs
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
*/
```

16

```
int num = 1;
int number;
int total = 0;

while (num <= 5) {
    cin >> number;
    total += number;
    num++;
}
cout << total << endl;
```

18

```
for (int a = 0; a < 10; a++) {
    cout << a << endl;
}

/* Outputs
0
1
2
3
4
5
6
7
8
9
*/
```

21

```
int num = 1;
int number;

while (num <= 5) {
    cin >> number;
    num++;
}
```

17

```
for (init; condition; increment) {
    statement(s);
}
```

19

```
for (int x = 1; x < 10; x++) {
    // some code
}
```

20

```
for (int a = 0; a < 50; a+=10) {
    cout << a << endl;
}

/* Outputs
0
10
20
30
40
*/
```

22

```
for (int a = 10; a >= 0; a -= 3) {
    cout << a << endl;
}

/* Outputs
10
7
4
1
*/
```

23

```
do {  
    statement(s);  
} while (condition);
```

 24

```
int a = 0;  
do {  
    cout << a << endl;  
    a++;  
} while(a < 5);  
  
/* Outputs  
0  
1  
2  
3  
4  
*/
```

 25

```
int a = 42;  
do {  
    cout << a << endl;  
    a++;  
} while(a < 5);  
  
// Outputs 42
```

 26

```
int a = 42;  
do {  
    cout << a << endl ;  
} while (a > 0);
```

 27

```
int age = 42;  
if (age == 16) {  
    cout << "Too young";  
}  
if (age == 42) {  
    cout << "Adult";  
}  
if (age == 70) {  
    cout << "Senior";  
}
```

 28

```
switch (expression) {  
    case value1:  
        statement(s);  
        break;  
    case value2:  
        statement(s);  
        break;  
    ...  
    case valueN:  
        statement(s);  
        break;  
}
```

 29

```
int age = 42;  
switch (age) {  
    case 16:  
        cout << "Too young";  
        break;  
    case 42:  
        cout << "Adult";  
        break;  
    case 70:  
        cout << "Senior";  
        break;  
}
```

 30

```
int age = 25;  
switch (age) {  
    case 16:  
        cout << "Too young";  
        break;  
    case 42:  
        cout << "Adult";  
        break;  
    case 70:  
        cout << "Senior";  
        break;  
    default:  
        cout << "This is the default case";  
}  
  
// Outputs "This is the default case"
```

 31

```
int age = 42; fall-through 32  
switch (age) {  
    case 16:  
        cout << "Too young" << endl;  
    case 42:  
        cout << "Adult" << endl;  
    case 70:  
        cout << "Senior" << endl;  
    default:  
        cout << "This is the default case" << endl;  
}  
/* Outputs  
Adult  
Senior  
This is the default case  
*/
```

Operator	Name of Operator	Form	
&&	AND Operator	y && y	
	OR Operator	x y	
!	NOT Operator	! x	33

AND

Left Operand	Right Operand	Result	
false	false	false	
false	true	false	
true	false	false	
true	true	true	34

OR (||)

Left Operand	Right Operand	Result	
false	false	false	
false	true	true	
true	false	true	
true	true	true	37

```
int age = 20;
if (age > 16 && age < 60) {
    cout << "Accepted!" << endl;
}

// Outputs "Accepted"
```

35

```
int age = 20;
int grade = 80;

if (age > 16 && age < 60 && grade > 50) {
    cout << "Accepted!" << endl;
}
```

36

```
int age = 16;
int score = 90;
if (age > 20 || score > 50) {
    cout << "Accepted!" << endl;
}

// Outputs "Accepted!"
```

38

```
int age = 10;
if ( !(age > 16) ) {
    cout << "Your age is less than 16" << endl;
}

// Outputs "Your age is less than 16"
```

39

Стандартні функції

```
#include <cmath>
```

підключить
математическую
библиотеку

abs (x) — модуль цілого числа
fabs (x) — модуль натурального числа
sqrt (x) — квадратний корень
sin (x) — синус кута, заданого **в радіанах**
cos (x) — косинус кута, заданого **в радіанах**
exp (x) — експонента e^x
ln (x) — натуральний логарифм
pow (x, y) — x^y : приведення числа x в степінь y
floor (x) — округлення «вниз»
ceil (x) — округлення «вверх»

```
float x;  
x = floor(1.6) ; // 1  
x = ceil(1.6) ; // 2
```

```
x = floor(-1.6) ; //-2  
x = ceil(-1.6) ; //-1
```

Генератор випадкових чисел

```
#include <cstdlib>
```

Генератор на відрізку [0,RAND_MAX]:

```
int X, Y;  
X = rand(); // псевдовипадкове число  
Y = rand()  // це вже інше число!
```

Цілі числа на відрізку [a,b]:

```
int X, Y;  
X = a + rand() % (b - a + 1);  
Y = a + rand() % (b - a + 1);
```

[0,b-a]

Функція `rand()` один раз генерує випадкові числа, а при наступних запусках програми всього лише відображає згенеровані перший раз числа. Така особливість функції `rand()` потрібна для того, щоб можна було правильно налагодити розроблювану програму. При налагодженні програми, внівши якісь зміни, необхідно упевнитися, що програма спрацьовує правильно, а це можливо, якщо вхідні дані залишилися ті ж, тобто згенеровані числа. Коли програма успішно налагоджена, потрібно, щоб при кожному виконанні програми генерувалися випадкові числа. Для цього потрібно скористатися функцією `srand()` зі стандартної бібліотеки C. Функція `srand()` отримавши цілий позитивний аргумент типу `unsigned` або `unsigned int` (без знакове ціле) виконує рандомізацію, таким чином, щоб при кожному запуску програми функція `srand()` генерувала випадкові числа. Програма, використовує функцію `srand()` для рандомізації генератора випадкових чисел `rand()`

```
1  #include <iostream>
2  #include <stdio.h>
3  #include <locale.h>
4  #include <stdlib.h>
5  #include <ctime>
6  using namespace std;
7
8  int main() {
9      setlocale (LC_ALL, "Rus");
10     int X, Y;
11     srand( time( 0 ) );
12     X = rand()% 100; // псевдовипадкове число
13     Y = rand()% 100;
14     cout << X << "\t" << Y;
15     return 0;
16 }
```