

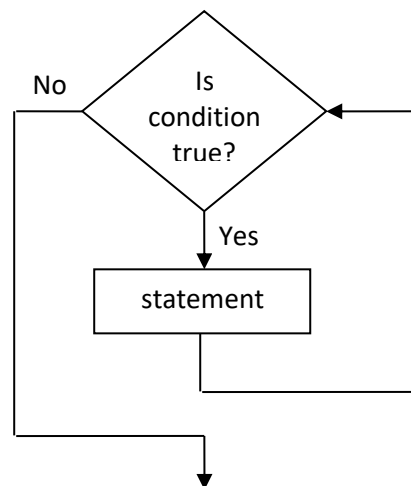
Iteration (Repetition) statements

1. Iteration (Repetition) statements

1) while statement 2) do/while statement 3) for statement

1.1 while statement

while (condition)
statement;



The statement within the loop must modify variables in the condition; otherwise, the value of the condition will never change, and will never be able to exit the loop (i.e. **infinite loop**). Infinite loop is generated if the condition in a loop is always true.

Example: Write a C++ program that computes the sum of consecutive integer numbers $1 + 2 + 3 + \dots + n$.

```
#include <iostream.h>
void main()
{
    int n , i = 1;
    long sum = 0;
    cout << "Enter a positive integer number: ";
    cin >> n;

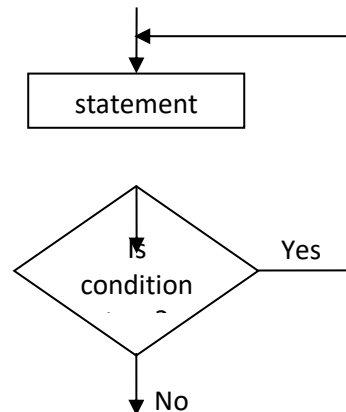
    while (i <= n)
        sum += i++;
    cout << "The sum of the first " << n
        << " integers is " << sum;
}
```

Example: Write a C++ program that computes the sum of ten numbers input by the user. Use while loop.

```
#include <iostream.h>
void main()
{
    int number , sum = 0 , i = 1 ;
    while( i <= 10 )
    {
        cout << "Enter an integer number: ";
        cin >> number;
        sum += number;
        i++;
    }
    cout<<"Sum = " << sum << endl;
}
```

1.2 do/while statement

```
do
    statement;
while (condition);
```



Example: Consecutive integer numbers $1 + 2 + 3 + \dots + n$. Use do/while loop.

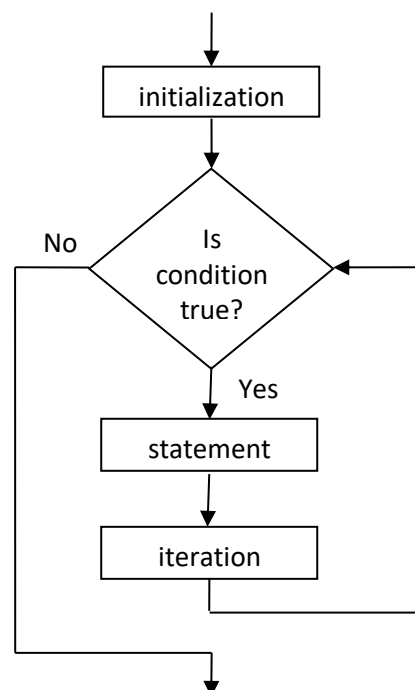
```
#include <iostream.h>
void main()
{
    int n , i = 1;
    long sum = 0;
    cout << "Enter a positive integer: ";
    cin >> n;
    do
        sum += i++;
    while (i <= n);
    cout << "The sum of the first " << n
        << " integers is " << sum;
}
```

Example: Write a C++ program that computes the sum of integer numbers input by the user. The program should stop when the user enters zero.

```
#include <iostream.h>
void main()
{
    int number;
    long sum = 0;
    do
    {
        cout<<"Enter an integer number: ";
        cin >> number;
        sum += number;
    }while(number);    // or while(number != 0);
    cout << "Sum = " << sum;
}
```

1.3 for statement

```
for ( initialization ; condition ; iteration )
    statement ;
```



Examples:

1. increasing

```
for ( int i = 1 ; i <= 100 ; i++ )
```

2. decreasing

```
for ( int i = 100 ; i >= 1 ; --i )
```

3. increasing by 7

```
for ( int i = 7 ; i <= 77 ; i += 7 )
```

4. decreasing by 5

```
for ( int i = 20 ; i >= 2 ; i -= 5 )
```

Example: Write a C++ program that prints the numbers from 1 to 20.

```
#include <iostream.h>
void main()
{
    for ( int i = 1 ; i <= 20 ; i++ )
        cout << i << " ";
    cout << endl;
}
```

Example: Write a C++ program that computes the sum of ten integer numbers input by the user. Use for loop.

```
#include <iostream.h>
void main()
{
    int number;
    long sum = 0;
    for ( int i = 1 ; i <= 10 ; i++ )
    {
        cout << "Enter an integer number: ";
        cin >> number;
        sum += number;
    }
    cout << "Sum = " << sum << endl;
}
```

Example: Write a C++ program that computes the factorial of an integer number.

```
#include <iostream.h>
void main()
{
    int number;
    long fact = 1;
    cout << "Enter a positive integer number: ";
    cin >> number;
    for ( int i = number ; i > 1 ; i-- )
        fact *= i;
    cout << "The factorial is " << fact << endl;
}
```

1.4 Nested for statements

```
cout<<"i\tj\n";
for (int i = 1 ; i <= 3 ; i++)
    for (int j = 1 ; j <= 3 ; j++)
        cout << i << "\t" << j << "\n";
```

The output is:

i	j
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

Example: Write a C++ program that prints the multiplication table.

```
#include <iostream.h>
#include <iomanip.h>
void main()
{
    cout << "\t\tMultiplication Table\n\n\n ";
    for ( int i = 1 ; i <= 10 ; i++ )
        cout << setw(5) << i;
    for ( i = 1 ; i <= 10 ; i++ )
    {
        cout<< "\n\n" << i << setw(5);
        for ( int j = 1 ; j <= 10 ; j++ )
            cout << i*j << setw(5);
    }
}
```

1.5 break and continue statements

The `break` statement is used to exit immediately from the loop in which it is contained.

The `continue` statement is used to skip the remaining statements in the body of the loop and then continue with the next iteration of the loop.

Example: break statement

```
#include <iostream.h>
void main()
{
    for ( int i = 1 ; i <= 10 ; i++ )
    {
        if ( i == 5 )
            break;
        cout<< i << " ";
    }
    cout << "\nBroke out of loop at i = " << i
        << endl;
}
```

Example: continue statement

```
#include <iostream.h>
void main()
{
    for ( int i = 1 ; i <= 10 ; i++ )
    {
        if ( i == 5 )
            continue;
        cout<< i << " ";
    }
    cout << "\nUsed continue to skip printing 5" << endl;
}
```


6. What is the output of the following C++ program?

```
#include <iostream.h>
void main()
{
    for(int c = 7 ; c <= 16 ; c++)
        switch(c % 10)
        {
            case 0: cout<<" , "; break;
            case 1: cout<<"OFTEN "; break;
            case 2:
            case 8: cout<<"IS "; break;
            case 3: cout<<"NOT "; break;
            case 4:
            case 9: cout<<"DONE "; break;
            case 5: cout<<"WELL "; break;
            case 6: cout<<" . "; break;
            case 7: cout<<"WHAT "; break;
            default: cout<<" bad number. ";
        }
    cout<<endl;
}
```

7. Write a program that calculates the value of (pi) from the following series. Stop calculation when the value of (pi) exceeds 8.7235.

$$\text{Pi} = 4 + 4/3 + 4/5 + 4/7 + 4/9 + \dots$$

8. Write a program that converts a positive integer number into binary.
9. Write a program that converts a positive integer number into octal.
10. Write a C++ program that reads several integer numbers input by the user and finds the smallest number. The user should first enter a value that specifies the number of integer values remaining to be entered.
- Ex:** n=9
10, 8, 4, 33, 6, 91, 44, 22, 89
Smallest is 4
11. An integer number is said to be a prime if it is divisible only by 1 and itself. Write a C++ program that determines if a number is a prime and use this program to determine and print all the prime numbers between 10 and 30.

12. Assume $j=0$, what is the new value of j at the end of each of the following loops?

- ```
for(int i = 8 ; i >= 0 ; i = i - 3)
 j = j + 1;
```
- ```
for(int i = 0 ; i <= 8 ; i = i + 2 )  
    j = j + 1;  
    i = i + 1;
```

13. If ($i=0$) and ($g=5$), what are the new values of i and g after the following program segment?

```
while ((i <= 4) && (g > 0))  
{  
    i = i + 1;  
    g = g - 1;  
}
```

14. Replace the following `for` loop with a corresponding `while` loop.

```
float a = 8;  
for (int i = 0 ; i < 10 ; i ++)  
{  
    if (a == 0) continue;  
    cout << 1/a << endl;  
    a = a - 1;  
}
```

15. What is the output of each of the following C++ code segments:

- a)

```
int x = 0;  
while (x < 10)  
    cout << x++ << endl;  
cout << "Done\n";
```
- b)

```
char ch;  
for (ch = 'A' ; ch <= 'F' ; ch = ch + 1)  
    cout << ch;  
cout << endl;
```

16. Write a C++ program that reads a positive integer number and computes the sum of its decimal digits.

Ex: 7354

Sum is 19