OL Academy

ITCS110/CSC103

Computer Programming for Scientists and Engineers

مدرس المقرر

أحمد كريم

Part 1

Chapter 5 : Control Structures II Repetition

The while Loop

The loop is repeated while the controlling boolean expression is true, when the expression is false the repetition ends.

Syntax

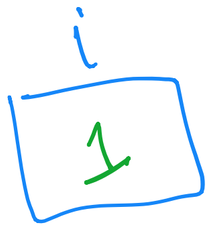
```
while (expression)
   Statement
```

or

```
while (expression)
{
   First_Statement
   Second_Statement
        ...
        ...
   Last_Statement
}
```

Infinite Loop

- A loop which repeats without ever ending is called an **infinite loop**.
- if the controlloing boolean expression never become false a loop will repeat without ending.



Example (1)

```
int i=2;
while (i<=4)
{
    cout<<i;
    i--;
}
```

2 <= 4 true
1 <= 4 true
0 <= 4 true
-1 <= 4 true
⋮
∞

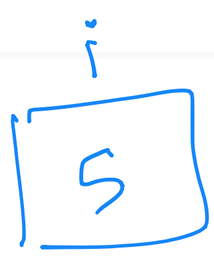
Output
2
1
0
-1
⋮
-∞

Case 1 | Counter-Controlled while Loops

If you know exactly how many pieces of data need to be read, the **while** loop becomes a counter-controlled loop.

```
counter = 0;
while (counter < N)
{
      .....
      .....
      counter++;
      .....
      .....
}
```

Example (2)



(A)

```
int i = 0;
while (i<5)
{
      cout<<i<<endl;
      i++;
}
```

5<5 ~~true~~ false

Output
0
1
2
3
4

(B) Second Version (without Braces)

```
int i = 0;
while (i<5)
     cout<<i++<<endl;
```

0<5
1<5
2<5
3<5
4<5
} true

5<5 false

Output
0
1
2
3
4

Case 2 | Sentinel-Controlled while Loops

Sentinel variable is tested in the condition and loop ends when sentinel is encountered.

```
cin >> varible;
while (variable != sentinel)
{
        .....
        .....
        cin >> varible;
        .....
        .....

}
```

body

Example (3)

num

1

```
int num = 9;
while (num != 0)
{
      cout<< num % 2 << '\t';
      num /= 2;
}
```

0 != 0 false

num = num/2

Output
1 0 0 1

9 % 2 = 1

9 / 2 = 4

4 % 2 = 0

4 / 2 = 2

2 % 2 = 0

2 / 2 = 1

1 % 2 = 1

2 ⟌ 1

1

1 / 2 = 0

Test Yourself

Suppose that the input is 0 5 6 4 9 8 -1. What is the output of the following code?

```
int num;
int sum;
cin >> num;
sum = num;
while (num != -1)
{
    sum = sum + num * (num - 1);
    cin >> num;
}
cout << "Sum = " << sum << endl;
```

sum = 0

num: -1

sum: 190

-1 != -1

(0 != -1) true

num == -1

output: Sum = 190

Sum = 0 + 0(-1)
= 0 + (5)(4) = 20
= 20 + (6)(5) = 50
= 50 + (4)(3) = 62
= 62 + (9)(8) = 134
= 134 + (8)(7) = 190

Case 3 | Flag-Controlled while Loops

A flag-controlled while loop uses a bool variable to control the loop.

```
bool found = false;
while (!found)
{
      .....
      .....
      if (expression)
        found = true;
      .....
      .....
}
```

while (true)
{
if (
break;
}

Example (4)

Assume the number entered by the user = 5, 4, 2, 9, 15

```
bool exit = false;
int num, sum = 0;
while (!exit)
{
    cout <<"Enter an number: ";
    cin>> num;
    sum += num;
    if (sum > 20)
        exit = true;
}
cout<< " The sum = " << sum << endl;
```

Output

```
Enter an number:
Enter an number:
Enter an number:
Enter an number:
Enter an number:
 The sum = 35
```

exit: false → true

num: 15

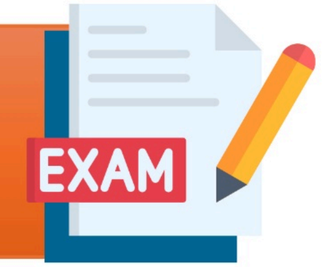
sum: 35

!false → true

!true → false

35 > 20 true

Test Yourself




```
int i=1, n=5;
while (i<n)
{
    if (n%i == 0)
        cout<<"?";
    else
        cout<<"#";
    i++;
}
```

false

5 < 5 true

5 % 4 == 0

i	n
1	5

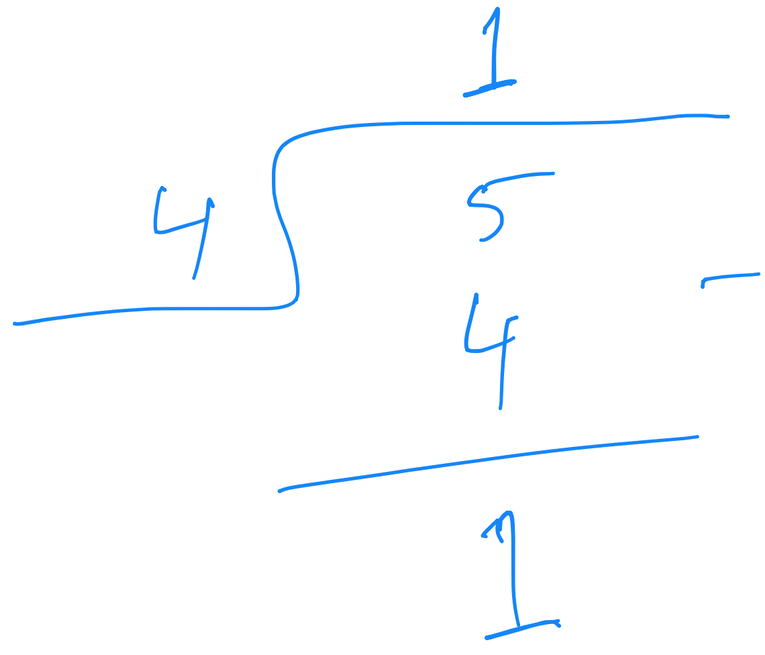
a) ?#?#?

b) ?###

c) ??##?

d) ?#?#

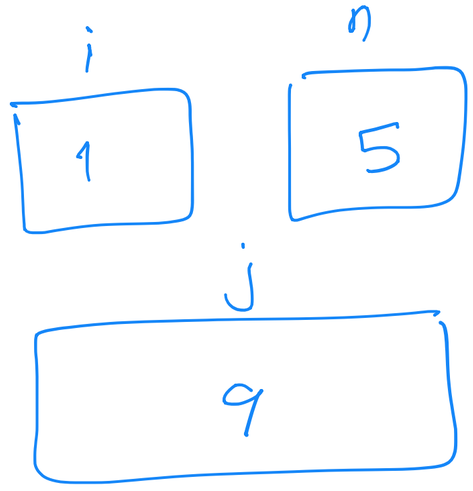
?###



Test Yourself


```
int i=1, n=5;
int j=12;
while (j >= 10)
{
    cout << 2 * j ;
    j-- ;
}
```

9 >= 10 ~~true~~ false



(a) 201816

(b) 1062

(c) 242220

(d) 181410

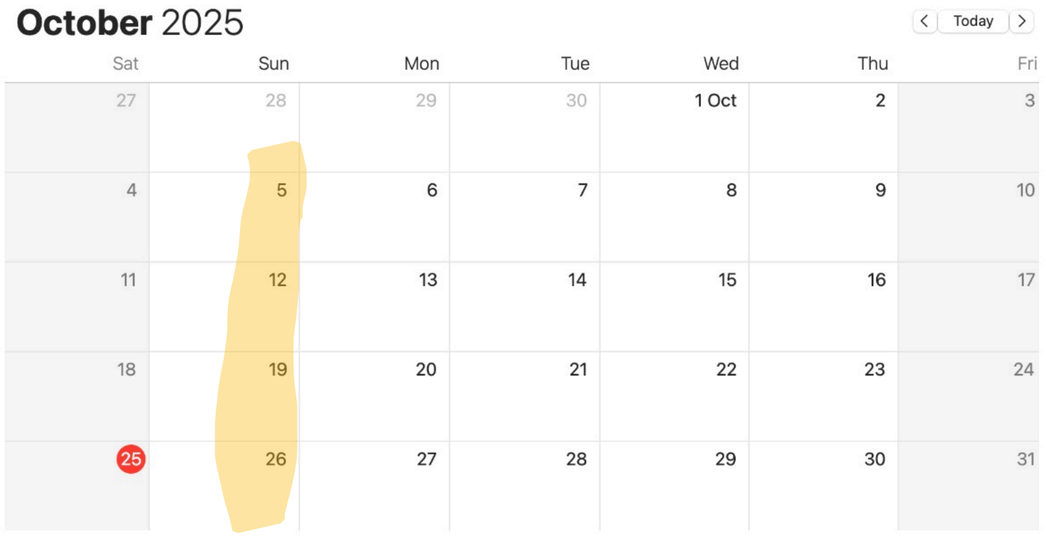
242220

1

Write a C++ program using a **Loop** to print the date of all Sundays in a month, if the date of the first Sunday is given. First prompt the user to input the number of days in the month, then prompt the user to input the date of the first Sunday in that month. After that find all Sundays in the given month and output. Assume all inputs are valid. Your program must follow the given Sample Input/Output.

Sample Input/Output

```
Enter number of days in the month: 31
Enter the date of first Sunday: 5
All Sundays in the month will be: 5 12 19 26
```



```
#include <iostream>
using namespace std;

int main ()
{
    int daysInMonth, SundayDate;
    cout << "Enter number of days in the month: ";
    cin >> daysInMonth;
    cout << "Enter the date of first Sunday: ";
    cin >> SundayDate;
```

```
Cout << "All Sundays in the month will be:";
while (sundayDate <= daysInMonth)
{
    Cout << sundayDate << " ";
    sundayDate += 7;
}
return 0;
}
```

(33 <= 31)

false

5 12 19 26