

```
...  
...  
i = 1;  
do{  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
}while(num < 1 || num > 20);  
...  
...  
do{  
    cout << "i = " << i << "\n", i;  
    i++;  
}while(i <= num;  
...
```

# Control Statement FOR & WHILE & DO-WHILE

## 04 CONTROLS

# Controls

## ❑ What is a Control?

- ❑ Statement used to enable some instructions to be executed repeatedly until some logical condition has been satisfied.

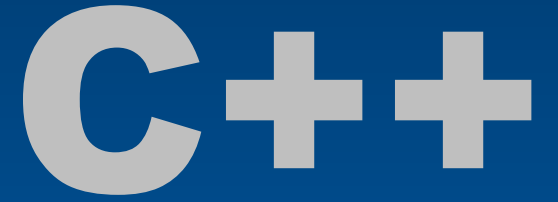
### ❑ Branching:

- ❑ if
- ❑ switch

### ❑ Looping:

- ❑ for
- ❑ while
- ❑ do - while

```
cout << "Hello World";  
cout << "Hello World";  
cout << "Hello World";  
...  
for( i=1 ; i<=3 ; i++ ){  
    cout << "Hello World";  
}  
...  
int i=1;  
do{  
    cout << i;  
    i++;  
}while(i<=3);
```



```
...  
...  
i = 1;  
do{  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
}while(num < 1 || num > 20);  
...  
...  
do{  
    cout << "i = " << i << "\n", i;  
    i++;  
}while(i <= num;  
...
```

# Control Statement FOR

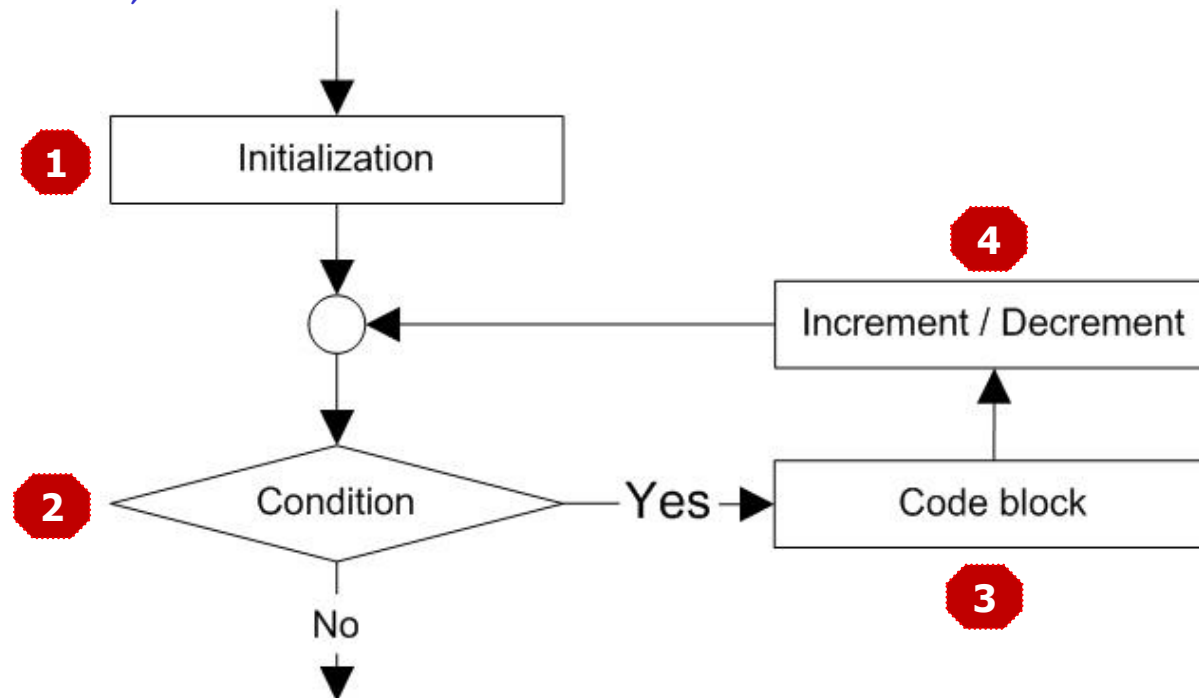
## 04 CONTROLS

# Control -- Looping : FOR

## □ Looping: for

### □ General form of *for* statement:

```
for(1initializations; 2condition; 4increments/decrements) {  
    statement; 3  
    statement;  
}
```



# Example

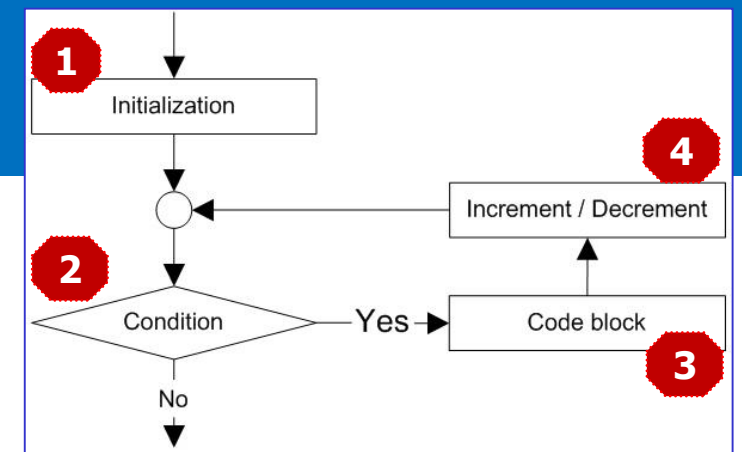
## Looping: for (cont.)

### Example:

`i = 3`

```
...  
→ int i;  
→ for(i = 1; FALSE; i++)  
  {  
→   cout << i << endl;  
→  }
```

Answer: `1`  
`2`



`1`                      `2`                      `4`

```
for(initializations; condition; increments/decrements) {  
    statement; 3  
    statement;  
}
```

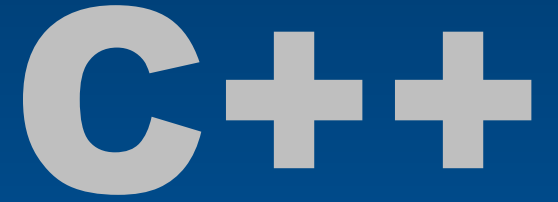
# Control -- Looping : NESTED FOR

- Looping: for
  - Form of *nested for* statement:

```
      1          2 12          11
for(initializations; condition; increments/decrements) {
      3 13          4 7 10 14          6 9 16
    for(initializations; condition; increments/decrements) {
      statement;
      statement 5 8 15
    }
}
```

```
for(i=0; i < 3; i++){
    cout << "i = " << i << endl;
    for(j=0; j < 2; j++){
        cout << "j: " << j << endl;
    }
}
```

i = 0	j = 0
	j = 1
i = 1	j = 0
	j = 1
i = 2	j = 0
	j = 1



```
...
...
i = 1;
do{
    cout << "Enter a number from 1 - 20: ";
    cin >> num;
}while(num < 1 || num > 20);
...
...
do{
    cout << "i = " << i << "\n", i;
    i++;
}while(i <= num;
...
```

# Control Statement

## CONTINUE & BREAK & GOTO

# 04

## CONTROLS

# Control -- Looping : CONTINUE

## ❑ Continue Statement

❑ Used to bypass the remainder of the current pass through a loop

❑ Example:

```
...
1. int  n, count;
2. float x, sum = 0;
3. cout << "How many number?: ";
4. cin >> n;
5. for(count = 1; count <= n; ++count){
6.     cout << "x = ";
7.     cin >> x;
8.     if(x < 0)
9.         continue;
10.    sum += x;
11. }
12. cout << "The sum of positive number is: " << fixed <<
    setprecision(2) << sum;
```

```
How many number?: 6
x = 1
x = -1
x = 2
x = -2
x = 4
x = -5
The sum of positive number is: 7.00
—
```



# Control -- Looping : BREAK & GOTO

## ❑ Break Statement

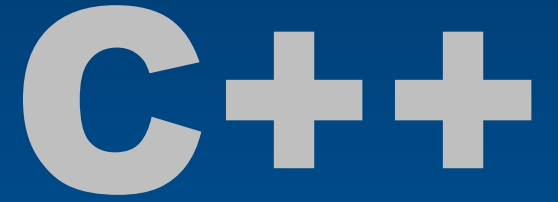
### ❑ Used to terminate loops

```
...  
1. char choice;  
2. choice = toupper(choice);  
3. switch (choice) {  
4.     case 'R':  
5.         cout << "RED";  
6.         break;  
7.     case 'B':  
8.         cout << "BLUE";  
9.         // break;  
10.        goto default;  
11.    default:  
12.        cout << "ERROR";  
13.        break;  
14. }  
...
```

R  
RED\_

B  
BLUE  
ERROR\_

x  
ERROR\_



```
...  
...  
i = 1;  
do{  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
}while(num < 1 || num > 20);  
...  
...  
do{  
    cout << "i = " << i << "\n", i;  
    i++;  
}while(i <= num;  
...
```

# Control Statement

## WHILE

# 04

## CONTROLS

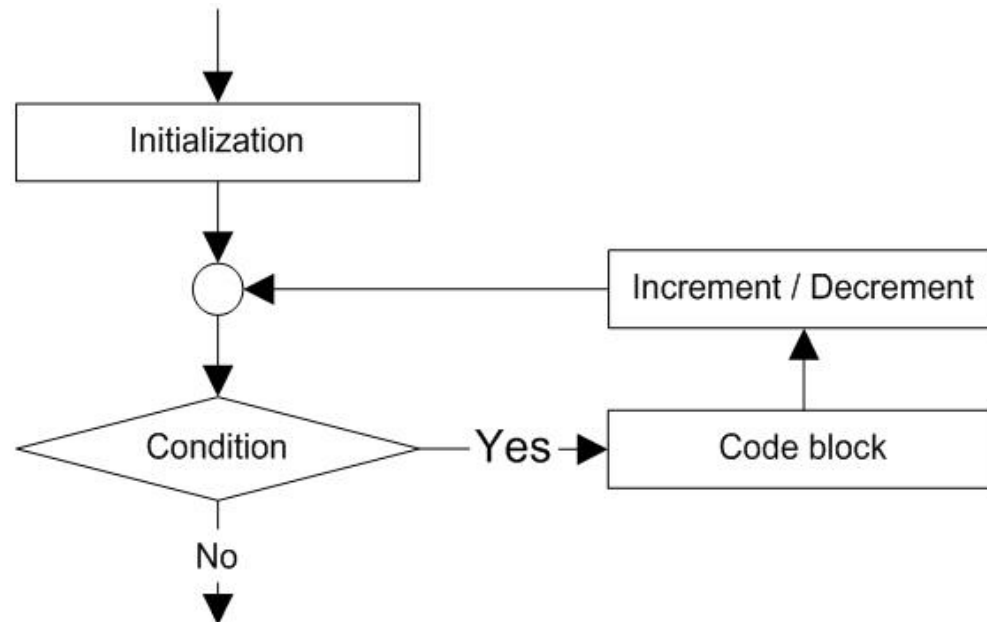
# Control -- Looping : WHILE

## □ Looping: while

### □ General form of *while* statement:

```
for(initializations; condition; increments/decrements) {  
    statement;  
    statement;  
}
```

```
initialization;  
while(condition) {  
    statement;  
    statement;  
    increment/decrement;  
}
```



# Example

- Looping: while (cont.)

- Example:

```
...  
int i = 0;  
  
while (i < 100) {  
    ++i;  
    cout << i << " Hello World!" << endl;  
}
```

# Example

## ❑ Looping: while (cont.)

### ❑ Example:

```
...  
int i = 0;  
  
while (i < 100) {  
    ++i;  
    cout << i << " Hello World!" << endl;  
}
```

=

```
...  
int i;  
  
for (i = 0; i < 100; ++i) {  
    cout << i+1 << " Hello World!" << endl;  
}
```

# Example (Cont.)

- Looping: while (cont.)

- Example:

```
...  
int i = 0;  
  
while (i < 100) {  
    //++i;  
    cout << i << " Hello World!" << endl;  
}
```

# Example (Cont.)

## □ Looping: while (cont.)

### □ Example:

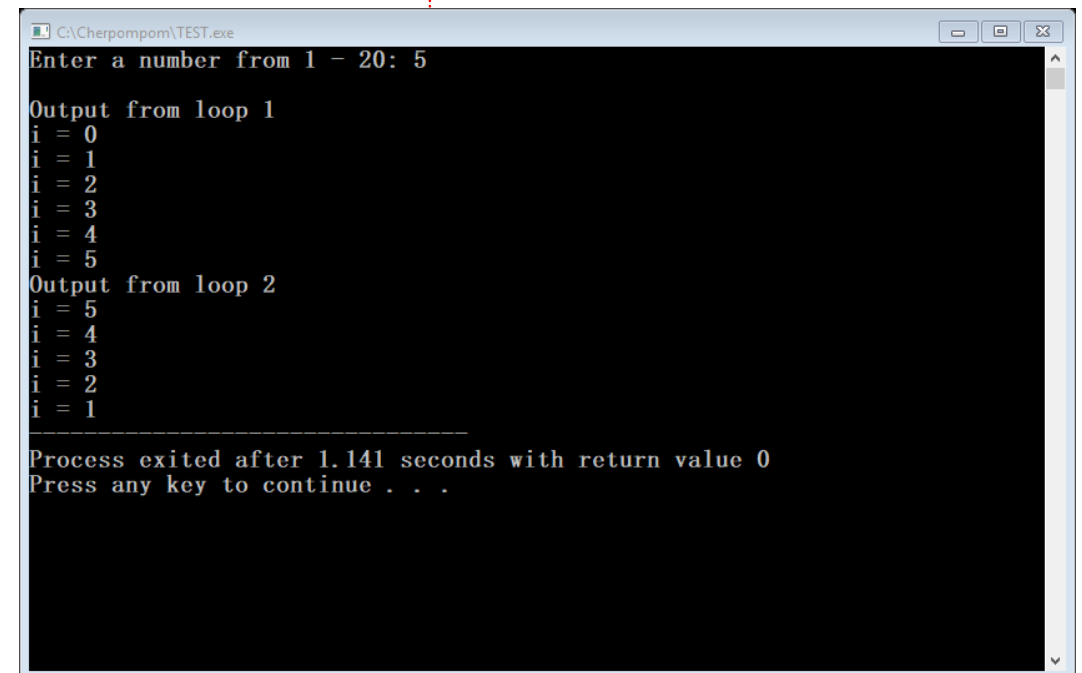
```
...
int i=0, num;

cout << "Enter a number from 1 - 20: ";
cin >> num;

cout << "\nOutput from loop 1";
while (i <= num) {
    cout << "\ni = " << i;
    i++;
}

i--;

cout << "\nOutput from loop 2";
while (i >= 1) {
    cout << "\ni = " << i;
    i--;
}
...
```



```
C:\Cherpompom\TEST.exe
Enter a number from 1 - 20: 5

Output from loop 1
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5

Output from loop 2
i = 5
i = 4
i = 3
i = 2
i = 1

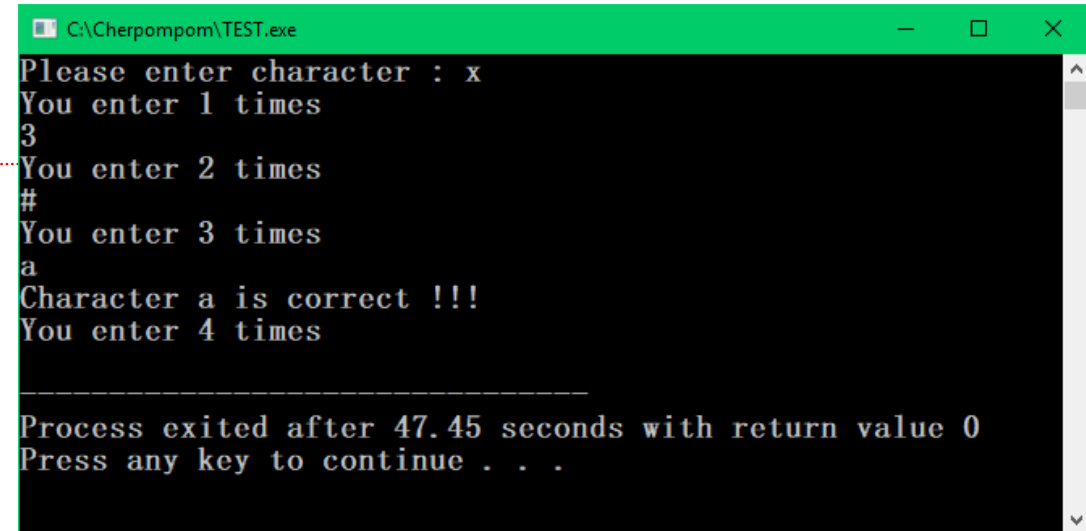
-----
Process exited after 1.141 seconds with return value 0
Press any key to continue . . .
```

# Example (Cont.)

## ❑ Looping: while (cont.)

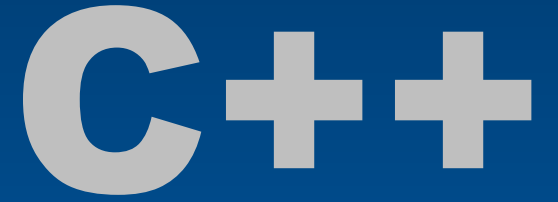
### ❑ Example:

```
...  
char ch;  
int count=0;  
  
cout << "Please enter character : ";  
  
while (ch!='8' && ch!='*' && ch!='a' )  
{  
    cin >> ch;  
    count++;  
    if( !(ch!='8' && ch!='*' && ch!='a') ){  
        cout << "Character " << ch << " is correct !\n";  
    }  
    cout << "You enter " << count << " times\n";  
}...
```



```
C:\Cherpompom\TEST.exe  
Please enter character : x  
You enter 1 times  
3  
You enter 2 times  
#  
You enter 3 times  
a  
Character a is correct !!!  
You enter 4 times  
-----  
Process exited after 47.45 seconds with return value 0  
Press any key to continue . . .
```





```
...  
...  
i = 1;  
do{  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
}while(num < 1 || num > 20);  
...  
...  
do{  
    cout << "i = " << i << "\n", i;  
    i++;  
}while(i <= num;  
...
```

# Control Statement

## DO-WHILE

# 04

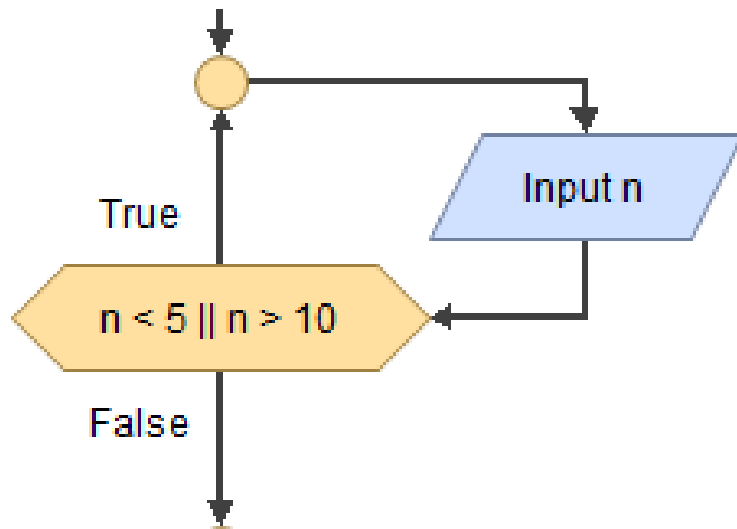
## CONTROLS

# Control -- Looping : DO-WHILE

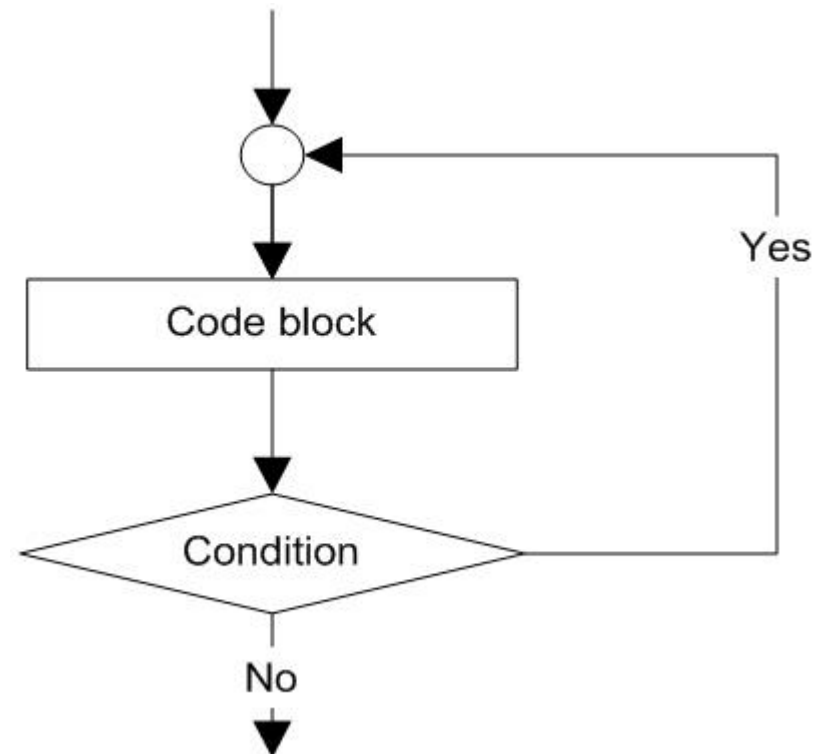
## □ Looping: do-while

### □ General form of *do-while* statement:

```
do{  
    statement 1;  
    statement 2;  
}while (condition);
```



(ในโปรแกรม flowgorithm)



# Control -- Looping : DO-WHILE

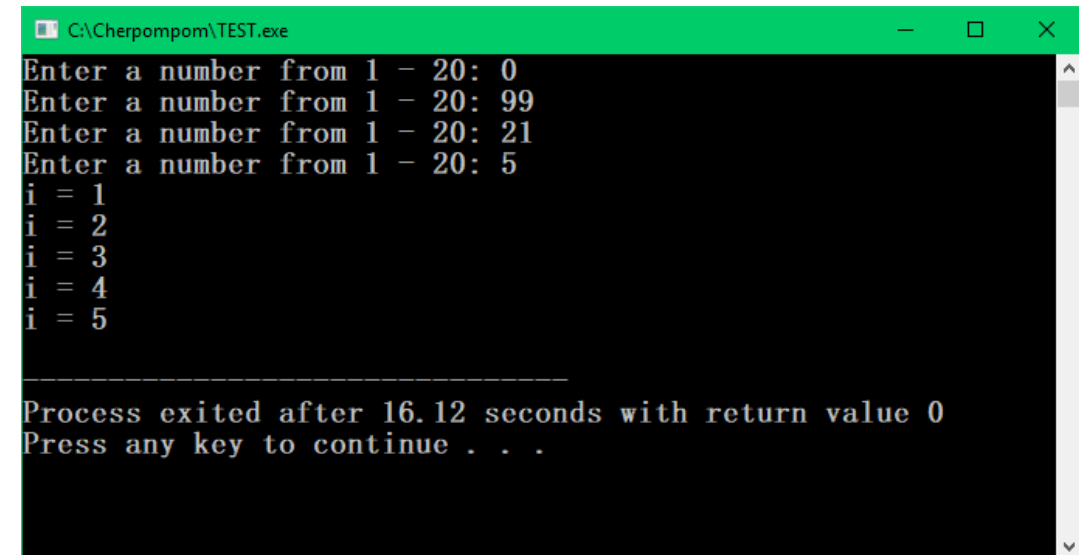
## □ Looping: do-while

### □ Example:

`num >= 1 && num <= 20`

-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

```
...  
int i=1, num;  
  
do {  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
} while(!(num >= 1 && num <= 20));  
  
do {  
    cout << "i = " << i << endl;  
    i++;  
} while(i <= num);  
...
```

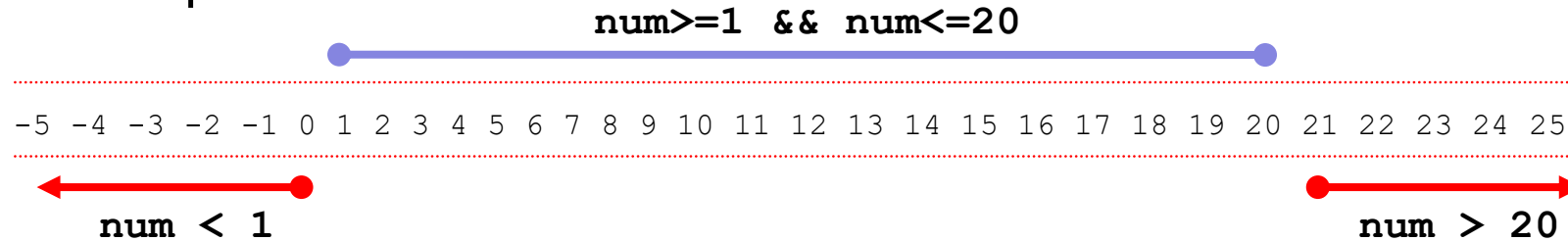


```
C:\Cherpompom\TEST.exe  
Enter a number from 1 - 20: 0  
Enter a number from 1 - 20: 99  
Enter a number from 1 - 20: 21  
Enter a number from 1 - 20: 5  
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
-----  
Process exited after 16.12 seconds with return value 0  
Press any key to continue . . .
```

# Control -- Looping : DO-WHILE

## □ Looping: do-while

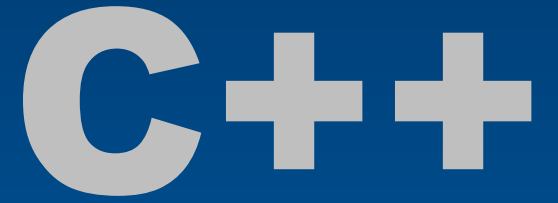
### □ Example:



```
...  
int i=1, num;  
  
do {  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
} while(num < 1 || num > 20);  
  
do {  
    cout << "i = " << i << endl;  
    i++;  
} while(i <= num);  
...
```

ทำ ... ในขณะที่ num  
ไม่อยู่ในช่วงที่ต้องการ  
do {  
 ...  
} while(!(num >= 1 && num <= 20));

ทำ ... ในขณะที่ num  
อยู่ในช่วงที่ไม่ต้องการ  
do {  
 ...  
} while(num < 1 || num > 20);



```
...  
...  
i = 1;  
do{  
    cout << "Enter a number from 1 - 20: ";  
    cin >> num;  
}while(num < 1 || num > 20);  
...  
...  
do{  
    cout << "i = " << i << "\n", i;  
    i++;  
}while(i <= num;  
...
```

## Example of a C++ program

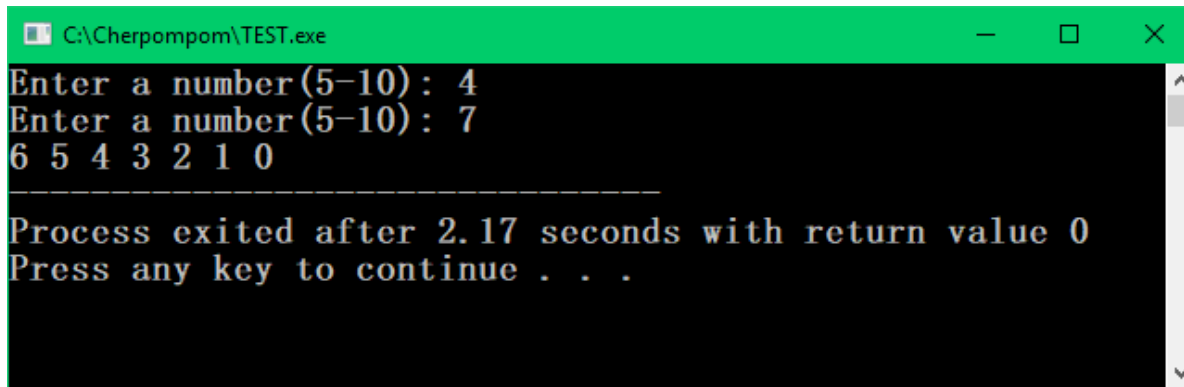
# 04

## CONTROLS

# Example of a C++ program

## □ Example 1

- จงเขียนโปรแกรมเพื่อรับข้อมูลเลขจำนวนเต็ม  $n$  โดยที่  $n$  จะต้องมามีค่าตั้งแต่ 5 ถึง 10 เท่านั้น หากไม่อยู่ในช่วง ให้ทำการรับค่าใหม่จนกว่าจะถูกต้อง จากนั้นจึงแสดงเลขที่น้อยกว่าค่าดังกล่าวจนถึง 0 ดังตัวอย่างการทำงานของโปรแกรม



```
C:\Cherpompom\TEST.exe
Enter a number(5-10): 4
Enter a number(5-10): 7
6 5 4 3 2 1 0
-----
Process exited after 2.17 seconds with return value 0
Press any key to continue . . .
```

5 10

ทำ ... ในขณะที่  $n$  ไม่อยู่ในช่วงที่ต้องการ

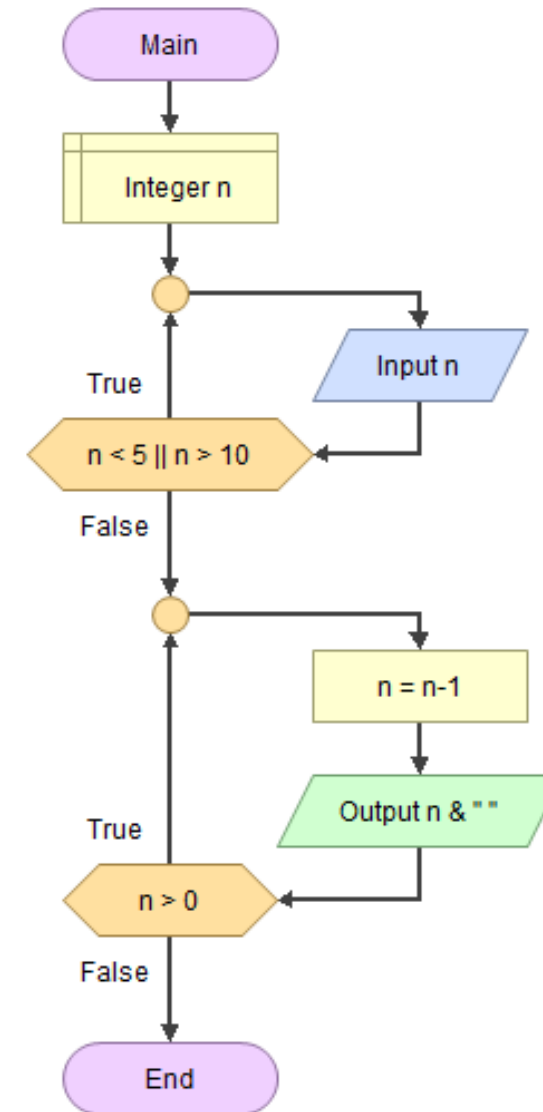
```
do { ...
} while(!(n >= 5 && n <= 10));
```

ทำ ... ในขณะที่  $n$  อยู่ในช่วงที่ไม่ต้องการ

```
do { ...
} while(n < 5 || n > 10);
```

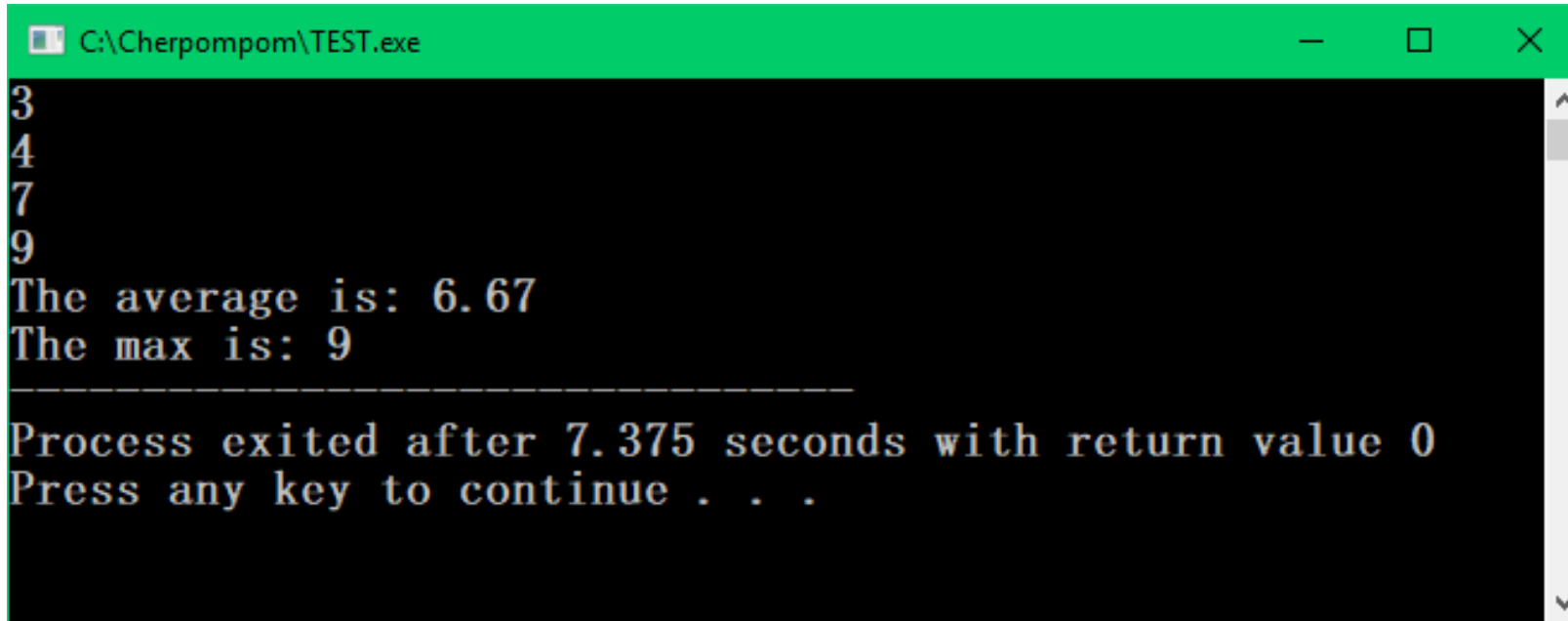
# Example of a C++ program

```
...  
int n, i;  
  
do {  
    cout << "Enter a number(5-10): ";  
    cin >> n;  
} while(n < 5 || n > 10);  
  
do {  
    n--;  
    cout << n << " ";  
} while(n > 0);  
...
```



# Example of a C++ program

- ❑ Example 2
- ❑ จงเขียนโปรแกรมเพื่อรับข้อมูลจากผู้ใช้ว่าต้องการป้อนข้อมูลกี่จำนวน จากนั้นทำการรับค่าของจำนวนเหล่านั้น แล้วพิมพ์ค่าเฉลี่ยและค่าที่มากที่สุดออกทางจอภาพ ดังตัวอย่างการทำงานของโปรแกรม



```
C:\Cherpompom\TEST.exe
3
4
7
9
The average is: 6.67
The max is: 9
-----
Process exited after 7.375 seconds with return value 0
Press any key to continue . . .
```



# Example of a C++ program

```
...
int n, i=1, x, max=-1;
float sum=0;

cin >> n;
while(i<=n){
    cin >> x;
    sum+=x;
    if(max<x)
    {
        max=x;
    }
    i++;
}

cout << "The average is: " << fixed << setprecision(2) << sum/n;
cout << "\nThe max is: " << max;
...
```