GECG10069 (561085) F25: Introduction to Programming (C++)

Lab 6 : Repetition Statements

What you will learn from Lab 6

In this lab, you will practice loops and learn how to avoid and debug infinite loops.

TASK 6-1 : FOR V.S. WHILE V.S. DO WHILE

- ✔ **for loop**: Best when the number of iterations is known; initialization, condition, and update are all in one line.
- ✔ **while loop**: Condition-driven; the body may not execute at all if the condition is false at the beginning.
- ✔ **do while loop**: Executes the body at least once before checking the condition.
  - ➢ Don't forget the semicolon in the end.

```
#include <iostream>
using namespace std;
int main() {
    const int BEGIN = 1, END = 5;
    // --- for loop ---
    cout << "for loop: ";
    for (int i = BEGIN; i <= END; i++) {
        cout << i << " ";
    }
    cout << endl;

    // --- while loop ---
    cout << "while loop: ";
    int j = BEGIN;
    while (j <= END) {
        cout << j << " ";
        j++;
    }
    cout << endl;

    // --- do while loop ---
    cout << "do while loop: ";
    int k = BEGIN;
    do {
        cout << k << " ";
        k++;
    } while (k <= END);
    cout << endl;

    return 0;
}
```

TASK 5-2 : INFINITE LOOP

- ✔ If the loop condition never becomes false, the loop runs forever.

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

int main() {
    int i = 1;

    // Wrong condition → infinite loop
    cout << "Infinite loop example:" << endl;
    while (i <= 5) {
        cout << i << " ";
        // i++;  // ⚠ Forgot to update i
    }
    cout << endl;

    return 0;
}
```

- ✔ The output result on OneCompiler should look like the screenshot below.

```
Output:

Infinite loop example:
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
RangeError [ERR_CHILD_PROCESS_STDIO_MAXBUFFER]: stdout maxBuffer length exceeded
```

- ✔ Infinite for loop

  - ➢ Wrong comparison operator

```cpp
for (int i = 0; i > 5; i++) {  // should be i < 5
    cout << i << " ";
}
```

  - ➢ Wrong update direction (++ vs --)

```cpp
for (int i = 0; i < 5; i--) {  // should be i++
    cout << i << " ";
}
```

  - ➢ Missing update expression

```cpp
for (int i = 0; i < 5; ) {  // forgot update(i++)
    cout << i << " ";
}
```

- Logic error in condition

```
for (int i = 1; i != 10; i += 2) {  // never equals 10
    cout << i << " ";
}
```

- Unintended assignment instead of comparison

```
for (int i = 0; i = 5; i++) {  // i = 5 is assignment, always true
    cout << i << " ";
}
```

✔ Add cout statements inside the loop to trace variable values and check whether they update as expected.

EXERCISE 6-1 : COUNT ZEROS UNTIL STOP

Description -

Write a program that keeps reading integers from the user one by one.

If the input is **'0'**, keep counting.

If the input is **'1'**,
print the number of zeros entered before the first **'1'** and stop the program.

Input :

- A single line containing only **'0'** or **'1'** separated by spaces.
- The input will always contain at least one **'1'**.

Reminder & Hint :

- The input might contain several **'1'**,
  but you have to print the answer right after you read the first **'1'**.

- You can use a loop '**`while(true)`**' with '**`cin`**' inside.
  Use '**`break`**' when you need to leave the loop.

Constraints :

- *length (input) < 30*
- *All the input number ∈ {0,1}*
- *The input consists of exactly **one** line*

Sample Testcases -

Sample Input - 1
0 0 1 1 0 0 1 0
Sample Output - 1
2

Sample Input - 2
1 0 0 0 0 0 0 1 0 0 1
Sample Output - 2
0

Sample Input - 3
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1
Sample Output - 3
13

EXERCISE 6-2 : INVERTED PYRAMID

Description -

Write a program to print an **inverted pyramid** composed of numbers.
Please print it based on the input number **`n`** as shown in the sample input / output.

Input - only one integer (**`int`**) :

- **`n`:** the number of the pyramid levels

Constraints :

- $n \in$ *integer (***`int`***)*
- $0 \leq n \leq 9$

Sample Testcases -

Sample Input - 1
9
Sample Output - 1
123456789876543 1234567876543 12345676543 1234565432 12345432 123432 1232 121 1

Sample Input - 2
0
Sample Output - 2 (Print NOTHING)