EEEC10012(515102) F22: Introduction to Computers and Programming

Lab 3 : Flow of Control (II) - Repetition

What you will learn from Lab 3

In this laboratory, you will understand how to use repetition (`for` and `while`) to control the flow of programs.

TASK 3-1 : FLOW CONTROL STATEMENT - `for`

Please predict the result of program `lab3-1`, and then execute this program to compare the result on screen with your prediction.

```
//File: lab3-1.cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "Open the refrigerator..." << endl;
    int count = 0;
    for (int idx = 0; idx < 10;  idx += 2)
    {
        cout << "Counter in loop " << count++ << "\t";
        cout << "Number of object: " << idx << endl;
    }
    cout << "...close the refrigerator." << endl;

    return 0;
}
```

An example for multiple structures.

```
//File: lab3-2.cpp
#include <iostream>
using namespace std;

int main()
{
    for(int i = 1; i <= 10; i++)
    {
        for(int j = 1; j <= i; j++)
            cout << j << " ";
        cout << endl;
    }

    return 0;
}
```

TASK 3-2 : FLOW CONTROL STATEMENT - `while`

✔ Program `lab3-3` is designed to print out the values from 1 to 10. Execute the program below and observe the results. If the results are different from the design, please modify the program properly.

```
//File: lab3-3.cpp
#include <iostream>
using namespace std;

int main()
{
    int i;
    i = 1;
    while (i < 10)
    {
        cout << i << endl;
        i++;
    }

    return 0;
}
```

- [?] Consider `i++` is removed from the program, what will you observe?
  - ✧ Note: `Ctrl + C` is the way to terminate the running program. (Don' use `Ctrl + Z`)
- [?] Rewrite file `lab3-3.cpp` by replacing `while` with `for` .

[?] Execute program `lab3-4` and record your result.

```
//File: lab3-4.cpp
#include <iostream>
using namespace std;

int main()
{
    int i;
    i = 10;
    do
    {
        cout << i << endl;
        i++;
    } while (i < 10);

    return 0;
}
```

- [?] Please modify the program by replacing `do … while` structure with `while` .
- [?] Note that the `;` (semicolon) is required in `do … while` structure.

? Execute the program `lab3-5` and record your result.

```
//File: lab3-5.cpp
#include <iostream>
using namespace std;

int main()
{
    int i;
    i = 10;
    while (i < 10)
    {
        cout << i << endl;
        i++;
    }

    return 0;
}
```

? Note: Observe carefully the differences between results from two programs using `while (lab3-5.cpp)` and `do … while(lab3-4.cpp)` structures.

TASK 3-3 : LOOPS - `break` and `continue`

? Execute program `lab3-6` and `lab3-7`, compare the results.

```
//File: lab3-6.cpp
#include <iostream>
using namespace std;

int main()
{
    int idx = 0;

    while(idx < 10)
    {
        idx++;
        if(idx == 5)
            continue;
        cout << idx << " ";
    }
    cout << endl;

    return 0;
}
```

```
//File: lab3-7.cpp
#include <iostream>
using namespace std;

int main()
{
    int idx=0;

    while(idx < 10)
    {
        idx++;
        if(idx == 5)
            break;
        cout << idx << " ";
    }
    cout << endl;

    return 0;
}
```

TASK 3-4 : DRAWING PICTURES

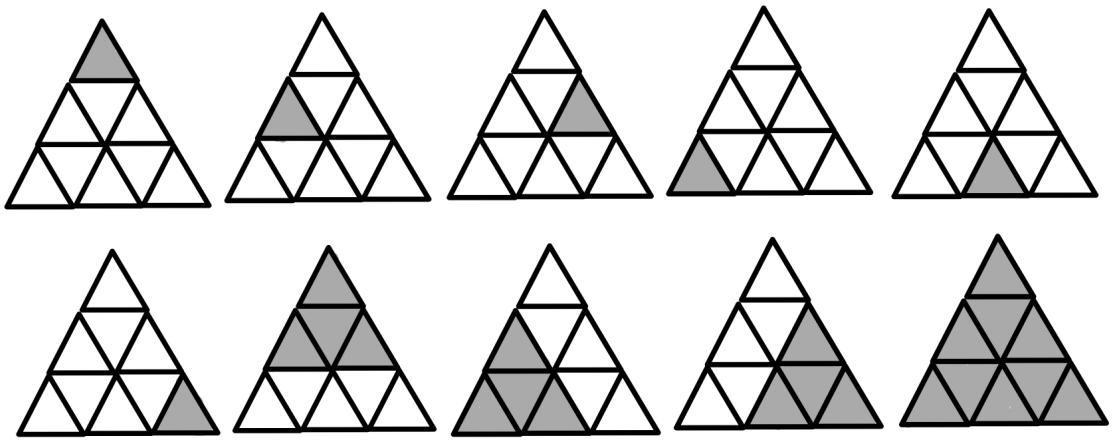
? Print out a rectangle with `for` loops.

```
//File: lab3-8.cpp
#include <iostream>
using namespace std;

int main()
{
    int a;
    cout << "Please enter the size of rectangle:";
    cin >> a;
    for(int i = 1; i <= a; i++)
    {
        if( i == 1 || i == a )
        {
            for(int j = 1; j <= a; j++)
                cout << "*";
        }
        else
        {
            for(int j = 1; j <= a; j++)
               if( j == 1 || j == a )
                   cout << "*";
               else
                   cout << " ";
        }
        cout << endl;
    }

    return 0;
}
```

EXERCISE 3-1 : RICHARD FEYNMAN

✔ Please count the number of different non-inverted equilateral triangles in a big equilateral triangle lattice which length is N.

- For example, there are 10 different equilateral triangles in a equilateral triangle lattice which length is 3.



- The program keeps running until the input value is 0.
- The input value is the side length of the big equilateral triangle lattice.
- If you want to stop your program, you must press the "Ctrl+C". (Don't use "Ctrl+Z".)
- The testing input N is an integer. ($0 \leq N \leq 100$)

```
TA_Nicholas@ICP:~/lab3$ ./ex3-1
Length = 1
1
Length = 3
10
Length = 10
220
Length = 0
TA_Nicholas@ICP:~/lab3$
```

Demo command: `$ /home/share/lab3/demo_lab3-1`

EXERCISE 3-2 : POKEMON GOGO

✔ In this exercise, you will encounter various wild Pokemon and need to fight with your only Pokemon.

- There are three input value. All of them are integers. The first one is the enemy's health point(HP). The second one is the enemy's attack power(ATK). The last one is the skill point(SP).
- The health point(HP) will reduce the value of the enemy's ATK after enemy attacks.
- Your Pokemon will attack first.
- Your Pokemon and the wild Pokemon need to attack by turns.
- If the Pokemon's health point(HP) is less than or equal to zero, it will lose the battle.
- Skill point(SP) means the number of the rest of the times that the Pokemon can attack. After the Pokemon attack, its SP will reduce 1. When the Pokemon is going to attack and its SP is less than or equal to 0, it will escape from the battle.
- Numerical value of your Pokemon: HP = 50, ATK = 9 - 12 - 5 - 3 (change each attack, 9->12->5->3->9……), SP = 15

```
TA_Nicholas@ICP:~/lab3$ ./ex3-2
Here comes an enemy!!!!
HP = 20
ATK = 8
SP = 20
You win the battle!
Rest of your HP is 42
TA_Nicholas@ICP:~/lab3$ ./ex3-2
Here comes an enemy!!!!
HP = 100
ATK = 27
SP = 5
You lose the battle!
Rest of your HP is -4
TA_Nicholas@ICP:~/lab3$ ./ex3-2
Here comes an enemy!!!!
HP = 10000
ATK = 2
SP = 30
You escape from the battle!
Rest of your HP is 20
TA_Nicholas@ICP:~/lab3$ ./ex3-2
Here comes an enemy!!!!
HP = 10000
ATK = 4
SP = 10
Your enemy escapes from the battle!
Rest of your HP is 10
TA_Nicholas@ICP:~/lab3$
```

Demo command: `$ /home/share/lab3/demo_lab3-2`

EXERCISE 3-3 : THE WAY TO THE GREAT COLLEGE

✔ Please print out the number stairs.

- ? The input value N is an integer ($0 \leq N \leq 5000$)
- ? The number stairs is composed of the numbers from 1 to N.
- ? Each of the number are followed by one space.
- ? The number stairs will go right 1 step and go down 1 step, then go right 2 step and go down 2 step, and so on.

```
[TA_Nicholas@ICP:~/lab3$ ./ex3-3
If you want to go fast, you go alone. If you want to go far, go together~~
How far you want to go?
[10
1 2
  3 4 5
      6
      7 8 9 10
[TA_Nicholas@ICP:~/lab3$ ./ex3-3
If you want to go fast, you go alone. If you want to go far, go together~~
How far you want to go?
[25
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
[TA_Nicholas@ICP:~/lab3$ ./ex3-3
If you want to go fast, you go alone. If you want to go far, go together~~
How far you want to go?
[120
1 2
  3 4 5
```

```
79 80 81 82
         83
         84
         85
         86
         87
         88
         89
         90
         91 92 93 94 95 96 97 98 99 100 101
                                         102
                                         103
                                         104
                                         105
                                         106
                                         107
                                         108
                                         109
                                         110
                                         111 112 113 114 115 116 117 118
```

Demo command: `$ /home/share/lab3/demo_lab3-3`