CS2310 Computer Programming
2017/18 Semester B
Department of Computer Science
City University of Hong Kong
Assignment One
Due Date: 2 March 2018 23:59

All questions should be submitted via Canvas under Assignments page

This assignment consists of 5 pages, including this cover page. It contains 2 questions named as Q1 to Q2.

In Q1-Q2, DO NOT INCLUDE any library other than <iostream> in your solution. Each solution of Q1-Q2 using any function/facility in other libraries will receive ZERO mark.

Solution Submission Instruction:

- You are required to submit a complete C++ program in one file via **Canvas** under the Assignments page.
- In addition to the running samples given for your own test, we will use a more comprehensive set of test cases for grading your solution. Therefore, reproducing the same results as in the samples provided to you do NOT mean you will get 100% correct in our grading tests. Try to test your program thoroughly before final submission. Happy coding! ☺

Grading:

Solution correctness (90%) and style (10%).

- **Correctness**: The correctness is based on how many test cases are correct with your program. If your program cannot be successfully compiled and tested by us, you will get **ZERO** mark.
- **Style**: We will check your source code for indentation, variable naming, comments, etc.

Plagiarism Check:

The course will use multiple code plagiarism detection software to check every solution the whole class submits. These software tools include the following two tools. Students are suggested to keep important immediate versions and paper draft of their solutions in case that students are asked to show evidences of self-effort and originality in developing their solutions of each question. All suspicious cases will be forwarded to the CS Departmental Disciplinary Committee for further actions.

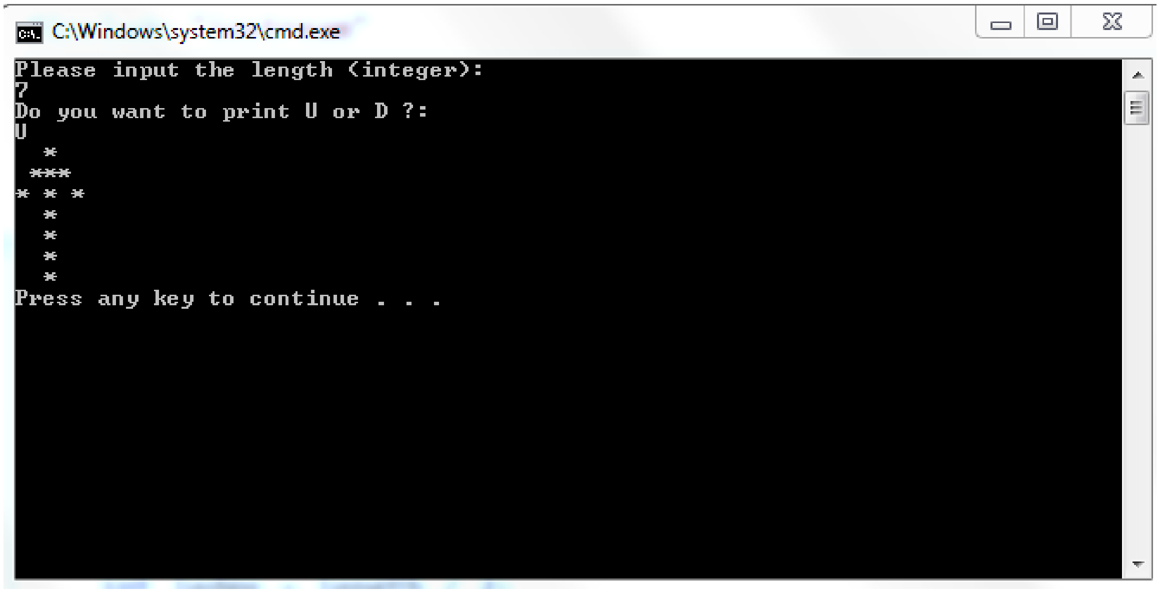
- PASS: https://pass3.cs.cityu.edu.hk/index.jsp and
- MOSS: https://theory.stanford.edu/~aiken/moss .

Q1 Write a program that meets all the following requirements:

1. Receive two inputs:
   a. An integer representing the size of the arrow ("↑" or "↓") to print. In particular, it represents the length of the central line to be printed out. Note that you can assume the input integer is always not less than 4.
   b. A character that indicates the direction of the arrow, i.e., "↑" or "↓". If the character is 'U', then output the shape of "↑"; if the character is D, output the shape of "↓". For the input other than 'U' and 'D', the program will print "Invalid input!" and terminate.
2. We assume that the arrow is made up of the symbol '*'. The length of an arrow is measured by the number of '*'. The number of '*' on each edge of the arrow is defined as $\lfloor 1/2 \cdot \text{length} \rfloor$, e.g., if length = 7, size(edge)=3 (as shown below); if length = 9, size(edge) = 4.

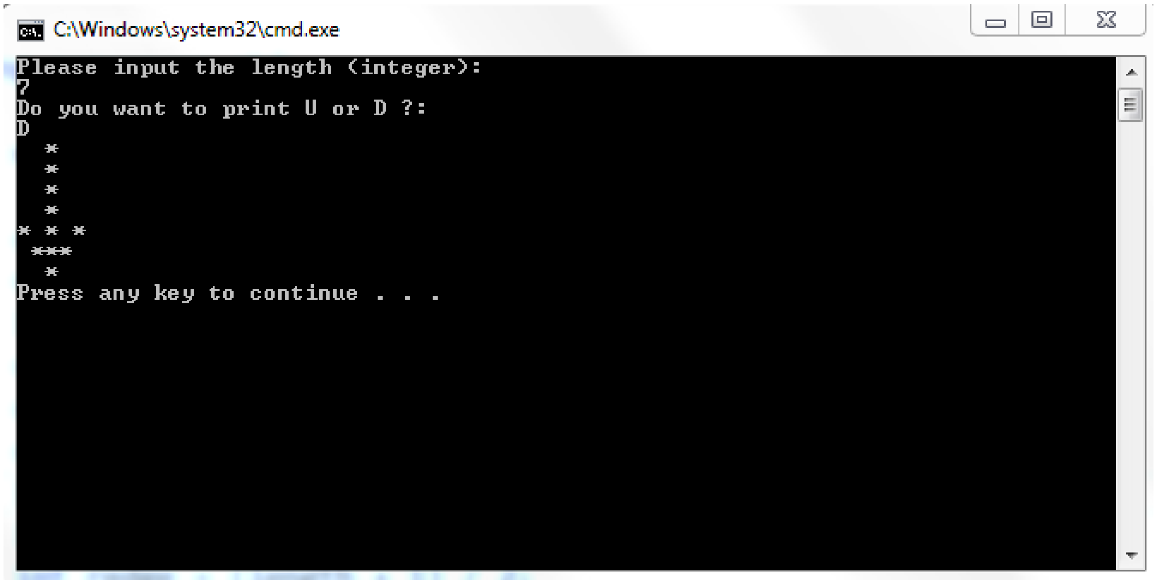
Some running samples are given below.

Sample 1:



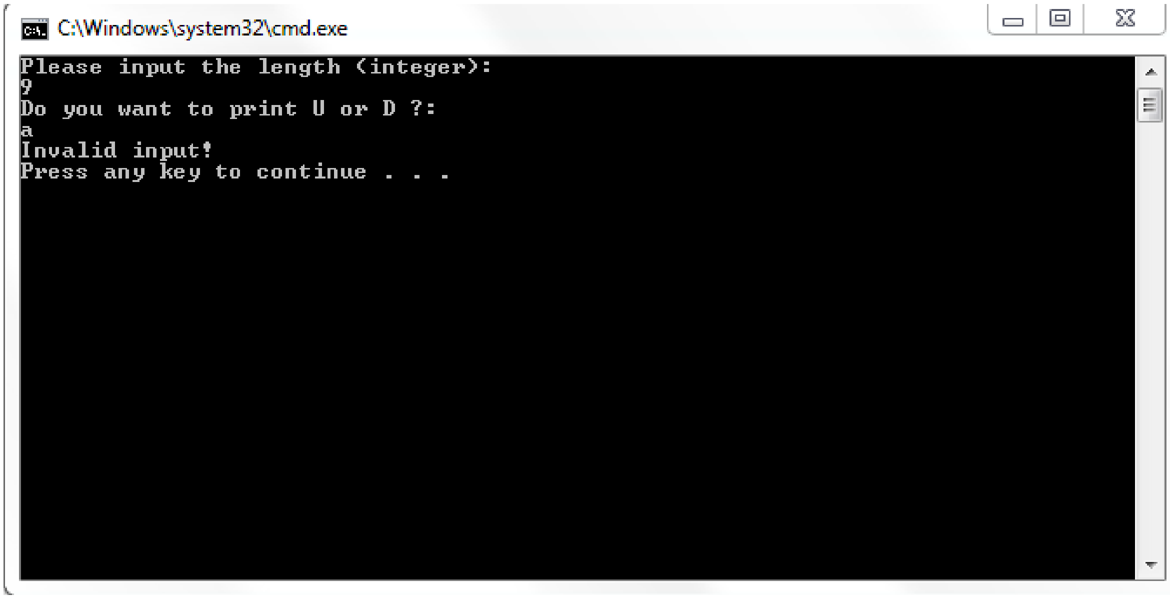
```
Please input the length (integer):
7
Do you want to print U or D ?:
U
  *
 ***
* * *
  *
  *
  *
  *
Press any key to continue . . .
```

Sample 2:



```
Please input the length (integer):
7
Do you want to print U or D ?:
D
  *
  *
  *
  *
* * *
 ***
  *
Press any key to continue . . .
```

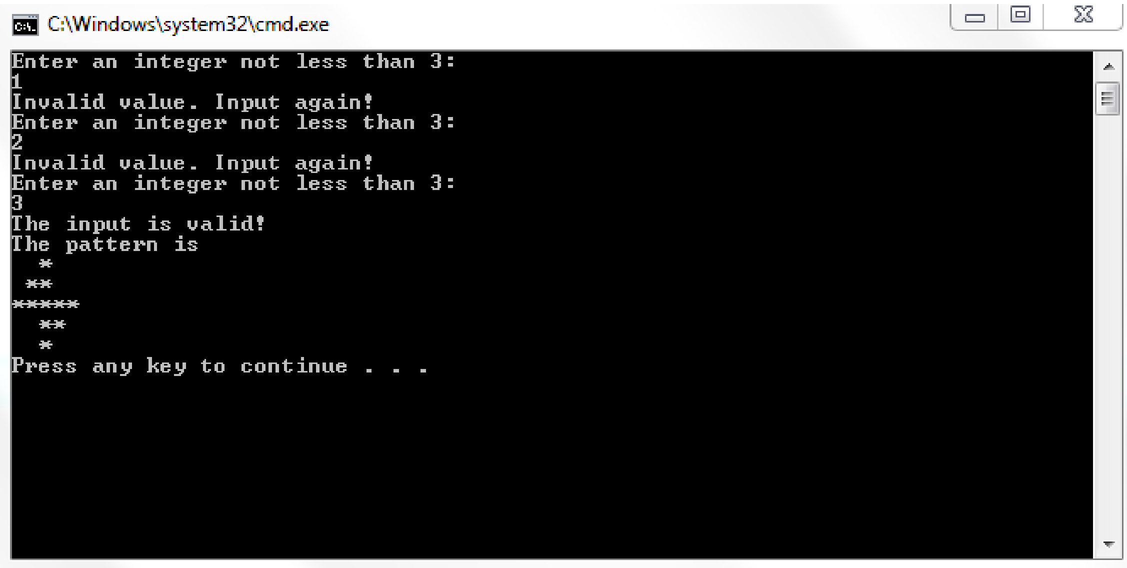
Sample 3:



```
Please input the length (integer):
9
Do you want to print U or D ?:
a
Invalid input!
Press any key to continue . . .
```

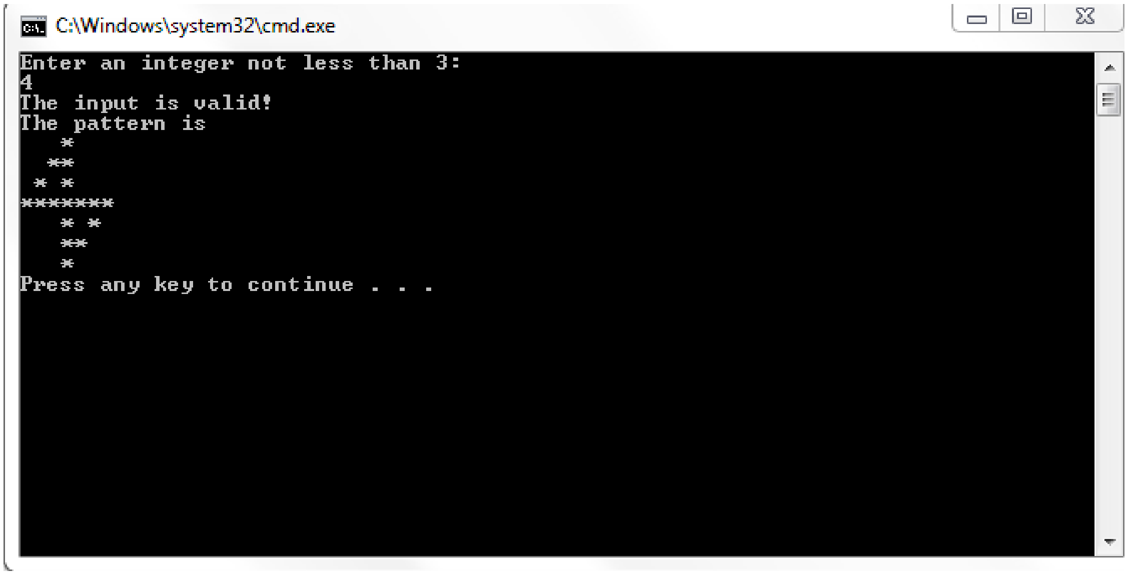
Q2. Write a program to print the pattern made up of two hollow triangles. The program receives as input an integer and then prints the pattern based on the integer. Note that the input integer value is required to be not less than 3. So, the program should keep doing validity check until the input integer is a valid one. This integer represents the number of rows occupied by a hollow triangle. Some running samples are given below:

Sample 1:



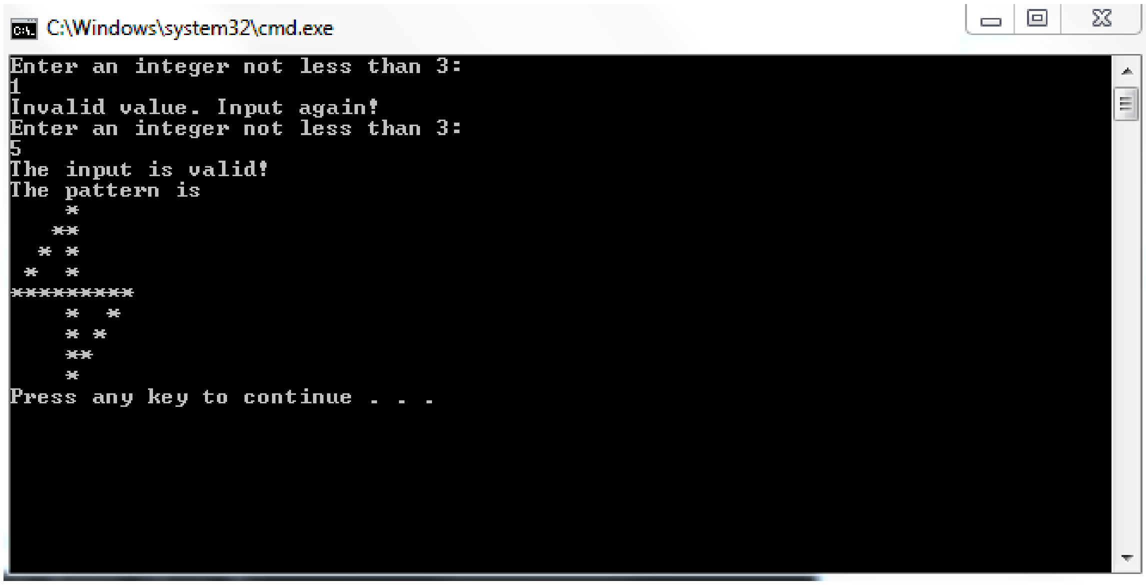
```
Enter an integer not less than 3:
1
Invalid value. Input again!
Enter an integer not less than 3:
2
Invalid value. Input again!
Enter an integer not less than 3:
3
The input is valid!
The pattern is
  *
 **
*****
  **
  *
Press any key to continue . . .
```

Sample 2:



```
Enter an integer not less than 3:
4
The input is valid!
The pattern is
   *
  **
 * *
*******
   * *
   **
   *
Press any key to continue . . .
```

Sample 3:



```
Enter an integer not less than 3:
1
Invalid value. Input again!
Enter an integer not less than 3:
5
The input is valid!
The pattern is
    *
   **
  * *
 *  *
*********
    *  *
    * *
    **
    *
Press any key to continue . . .
```