COSC 600
Programming Assignment 2

Problem 1 (Maximum Subsequence Sum Program)

Run four algorithms in the text book for N= 5000, 10000, 50000, 100000, 200000 which should be generated by random function in Java between -5000 and +5000.
Compare your analysis with the actual running(execution) times.
(If your program is running more than 15 minutes, you can terminate it.)

Problem 2 (Josephus Problem-joseph.cpp)

This problem is an exercise in the implementation of queue. We will look at the implementation of a circular array. Then we will use the implementation to solve a problem called the Josephus problem.
Josephus Flavius was a famous historian of the first century at the time of the Second Temple destruction. During the war he got trapped in a cave with a group of 39 soldiers surrounded by Romans. The legend has it that preferring suicide to capture, the people decided to form a circle and, proceeding clockwise around it, to kill every seventh(It will be given as an input) person until only one was left, who must then commit suicide. Josephus, an accomplished mathematician, quickly found the safe spot in the circle (24th) to be the last to go. But when the time came, instead of killing himself he joined the Roman side. The problem rightfully raises the question of how someone might be able to quickly compute the correct place to stand.

In this assignment you are to simulate the Josephus problem.

Input data (interactively input by keyboard):

How many soldiers? ➔ 10

Type 10 soldiers name:

Andy

Shawna

Jianjia

Adam

Jacob

Wesley

Zahari

Thomas

Chris

Ben

The program should simulate the Josephus problem by repeatedly removing the n-th name from the list and displaying it. At the end, display the name of the survivor. For the example above, your output should be:

```
Enter the position ➔ 3

Eliminating order:
1. Jianjia (3)
2. Wesley(6)
3. Chris(9)
4. Shawna(2)
5. Zahari(7)
6. Andy(1)
7. Thomas(8)
8. Jacob(5)
9. Ben(10)
```

(Note: The number of the parenthesis is an original spot number.)

```
The survivor is Adam(4).
```

Problem 3. (Longest sequence)

Find the longest increasing sequence of numbers in a 15 x 15 array. For example, if the array, 4x4, contains

```
97  47  56  36
35  57  41  13
89  36  98  75
25  45  26  17
```

then the longest increasing sequence of numbers is the sequence of length eight consisting of 17, 26, 36, 41, 47, 56, 57, 97. Note that there are no duplicates in the increasing sequence.

Design and write a program in Java to solve the same problem using a stack and run your program with given test data file.

Run your programs for an input data.

NOTE: A 15 x 15 matrix is generated by a random function. And the values are between 0 and 1000.

Output: Print the complete longest increasing sequence of numbers with their positions. (It is a sample output for the 4x4 example above.)

```
17  (3,3)
26  (3,2)
36  (2,1)
41  (1,2)
47  (0,1)
56  (0,2)
57  (1,1)
97  (0,0)

The length of the sequence is 8.
```