Test name: INFO 5100 SPRING 2018 MID TERM Part B

Total Questions : 4 + 1
Total Points : 30 + 10
Deadline : 17 March, 8: 00PM, PST.
Extra credit will be added only if Total score is less than 100. Submit it to the github in the same repository of your Assignments.

Question 1 of 5 6 pts

Given an unsorted array of integers, find the length of
longest continuous increasing subsequence (subarray). 10 pts

Example 1:

Input: [1,3,5,4,7]

Output: 3

Explanation: The longest continuous increasing subsequence is [1,3,5], its length is 3.

Even though [1,3,5,7] is also an increasing subsequence, it's not a continuous one where 5 and 7 are separated by 4.

Example 2:

Input: [2,2,2,2,2]

Output: 1

Explanation: The longest continuous increasing subsequence is [2], its length is 1.

Note: Length of the array will not exceed 10,000.

```
class Solution {
   public int findLongestLength(int[] nums){
        //your code
 }
}
```

Question 2 of 5 7 pts

Given an array of integers sorted in ascending order, find the starting and ending position of a given target value.

Your algorithm's runtime complexity must be in the order of O(log n).

If the target is not found in the array, return [-1, -1].

For example,
Given [5, 7, 7, 8, 8, 10] and target value 8,
return [3, 4].

```
class Solution {

    public int[] searchForRange(int[] nums, int target) {



}

}
```

Question 3 of 5 7 pts

Write a function rotate(ar[], d, n) that rotates arr[] of size n by d elements.

Rotation of the above array by 2 will make array


```
class RotateArray{
Public void rotatedArray(int nums[], int d, int n){
}


}
```

Question 4 of 5 10 pts

Given an array of integers and a number x, find the smallest subarray with sum greater than the given value.

Examples:

arr[] = {1, 4, 45, 6, 0, 19}

x = 51

Output: 3

Minimum length subarray is {4, 45, 6}

arr[] = {1, 10, 5, 2, 7}

x = 9

Output: 1

Minimum length subarray is {10}

arr[] = {1, 11, 100, 1, 0, 200, 3, 2, 1, 250}

x = 280

Output: 4

Minimum length subarray is {100, 1, 0, 200}

arr[] = {1, 2, 4}

x = 8

Output : Not Possible

Whole array sum is smaller than 8.

```
class SmallestSubArraySum{

        public static int smallestSubWithSum(int arr[], int n, int x){




}

}
```

Question 5 of 5 10 pts

Extra Credit

A Maze is given as N*N binary matrix of blocks where source block is the upper left most block i.e., maze[0][0] and destination block is lower rightmost block i.e., maze[N1][N1]. A rat starts from source and has to reach destination. The rat can move only in two directions: forward and down.

In the maze matrix, 0 means the block is dead end and 1 means the block can be used in the path from source to destination. Your function should take the maze as input and return an arrayList of the resulting path. If no path is found return empty list.

Example:

```
 {1, 0, 0, 0}
 {1, 1, 1, 1}
 {0, 1, 0, 0}
 {1, 1, 1, 1}
```

Output:

[[0, 0], [1, 0], [1, 1], [2, 1], [3, 1], [3, 2], [3, 3]]

```
class Cell{
   int x,y;
Cell(int x, int y){
this.x = x;
this.y = y;
}
public String toString(){
return "["+this.x +", "+this.y+ "]";
}
}
class Solution{
public ArrayList<Cell> findPath(int[][] maze){
//your code
}
}
```