Test name: INFO 5100 FALL 2017 MID TERM Part B

Total Questions : 4 + 1
Total Points : 30 + 10
Deadline : THU, 26 OCT, 11:59PM, 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

Write a method named reverseEvenIndices that takes an integer array as input and outputs an array such that all the values with odd indices remain in the same position.
However, elements with even indices should be output in reverse order. That is, the first element with even index should be swapped with the last element with even index, the second even-indexed element with the second-to-last even-indexed element, and so on.

Note that zero is an even index.

Example inputs and outputs are as follows:

```
Input: {9, 4, 8, 7, 5, 1, 3} Output: {3, 4, 5, 7, 8, 1, 9}
Input: {6, 4, 1, 0, 3, 2} Output: {3, 4, 1, 0, 6, 2}
Input: {1, 2, 3} Output: {3, 2, 1}
```

```
public int[] reverseEvenIndices(int[] nums){
            // your code
}
```

Question 2 of 5

7 pts

You have a total of n coins that you want to form in a staircase shape, where every k-th row must have exactly k coins.

Given n, find the total number of full staircase rows that can be formed.

n is a non-negative integer and fits within the range of a 32-bit signed integer.

Example 1:

```
n = 5

The coins can form the following rows:
¤
¤ ¤
¤ ¤

Because the 3rd row is incomplete, we return 2.
```

Example 2:

```
n = 8

The coins can form the following rows:
¤
¤ ¤
¤ ¤ ¤
¤ ¤

Because the 4th row is incomplete, we return 3.
```

```
public int arrangeCoins(int n){
        //your code
}
```

Question 3 of 5

7 pts

Given a non-empty integer array of size n, find the minimum number of moves required to make all array elements equal, where a move is incrementing n - 1 elements by 1.

Example:

```
Input:
[1,2,3]

Output:
3

Explanation:
Only three moves are needed
(remember each move increments two elements):

[1,2,3]  =>  [2,3,3]  =>  [3,4,3]  =>  [4,4,4]
```

```
public int minMoves(int[] nums){
//your code
}
```

Question 4 of 5

10 pts

Given n dice each with m faces, numbered from 1 to m, find the number of ways to get sum X. X is the summation of values on each face when all the dice are thrown. Your function should take, number of faces, number of dice and required sum as input and return a number of possible ways.

```
public int countNumberOfPossibleWays(int m, int n, int x){
        // your code
}
```

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[N-1][N-1]. 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
}
}
```