Assignment 3 – Lab questions – Coding (CS 202)

Both Coding Questions are of 4 marks each.

1) ISRO sent a rover on the moon.

Surface of Moon is a rectangular grid of blocks containing n rows (numbered from 0 to n-1 from north to south) and m columns (numbered from 0 to m-1 from west to east).
Let (i, j) denote the block in the i-th row and j-th column.
The rover begins on the block (0, 0).

The rover can be maneuvered around on the surface of the moon by sending it a program, which is a string of characters representing movements in the four directions. The rover executes characters of the string in order:
N : travel one block in north direction.
E : travel one block in east direction.
W : travel one block in west direction.
S : travel one block in south direction.

There is also a special instruction X(Y), where X is a number between 2 and 9 inclusive and Y is a non-empty subprogram.
This denotes that the robot should repeat the subprogram Y a total of X times. For example:
2(SWN) is equivalent to SWNSWN.
3(W2(S)) is equivalent to WSSWSSWSS.
NNN4(W)2(SS) is equivalent to NNNWWWWSSSS.

Suppose the Moon is a torus, the first and last columns are adjacent, so moving east from column m-1 (last column) will move the rover to column 0 (first row) and moving south from row n-1 (last row) will move the rover to row 0 (first row).
Similarly, moving west from column 0 will move the rover to column m-1 and moving north from row 0 will move the rover to row n-1.
Given a program that the robot will execute, determine the final position of the robot after it has finished all its movements.

Input
First line contains two numbers n and m (the number of rows and column respectively).
Second line contains a string S : the program sent to the rover.

Output
Output two numbers r and c. (where Block (r, c) is the final block rover finishes in).

Limits
1 <= n, m <= 1,000,000,000
0 <= Length of S <= 1,000,000

Sample 1 :-
Input :-
10 8
SSSEEE

Output :-
3 3

Sample 2 :-
Input :-
2 1
SSSEEE

Output :-
1 0

Sample 3 :-
Input :-
10 8
N8(S)N6(E)N

Output :-
5 6

Explanation :-
the given string is equivalent to NSSSSSSSSNEEEEEEN.

Sample 4 :-
Input :-
4 4
2(3(NW)2(W2(EE)W))

Output :-
2 2

Explanation :-
the given string is equivalent to NWNWNWWEEEEWWEEEEWNWNWNWWEEEEWWEEEEW.

2) You are given an array of integers Arr, you need to find the difference between maximum and minimum in every subarray of size K.

Input
First line contains two numbers N (size of Arr) and K (size of sub-array).
Second line contains N space separated integers.

Output
Output N-K+1 space separated integers (where i'th integer is
max(Arr[i], Arr[i+1], ------, Arr[i+K-1]) - min(Arr[i], Arr[i+1], ------, Arr[i+K-1])).

Limits
1 <= K <= N <= 1,000,000
0 <= Arr[i] <= 1,000,000,000
Expected time complexity -> O(n)
Expected space complexity -> O(n)

Sample 1 :-
Input
8 3
1 3 -1 -3 5 3 6 7
Output
4 6 8 8 3 4

Explanation :-

	Max	Min	Diff
[1 3 -1] -3 5 3 6 7	3	-1	4
1 [3 -1 -3] 5 3 6 7	3	-3	6
1 3 [-1 -3 5] 3 6 7	5	-3	8
1 3 -1 [-3 5 3] 6 7	5	-3	8
1 3 -1 -3 [5 3 6] 7	6	3	3
1 3 -1 -3 5 [3 6 7]	7	3	4

Sample 2 :-
Input
30 5
28 20 9 28 4 23 0 27 0 7 10 24 15 3 29 29 26 21 7 25 19 10 3 24 21 12 4 18 10 11
Output

24 24 28 28 27 27 27 27 24 21 26 26 26 26 22 22 19 18 22 22 21 21 21 20 17 14

Practice Questions(No marks associated with these problems)

Pam's Date:

(Dipanshu)

Michael has got to deal with the linked list in the office. However, he messed up things a little bit and added the next pointer of the last node to some other node in the list or so he says while he addresses his problem to Pam and asks her, to find the index of the node to which this last node points, over the weekend. However, Pam has plans with Jim this weekend. Help her solve this problem, by writing a program that returns the index (0 based) of the node in the list to which the last node was pointing to.
(**Note**: Since memory is nothing to be wasted, solve by not using any additional memory and in Linear time complexity.)

Since Dwight and Angela will be checking for the structure of Node. **Pam has to follow the given template and must make changes only to the Solve( ) function given below.**

Space complexity = O(1)
Time complexity = O(N) where N is the length of the longest Linked List.

Constraints:
$1<=N<=10^6$

```
#include <iostream>
using namespace std;

class Node {
    public:
    int data;
    Node* next;
    Node (int n) {
        data = n;
        next = nullptr;
    }
};



int Solve(Node *head){
    int link=-1;
```

```
    //
    //please write your code in here only.
    //make the link store the index of the node to which the last
node points.
    //
    return link;
}

Node* input(){
    int n;cin>>n;
    int x;cin>>x;
    Node* head=new Node(x);
    Node *ptr = head;
    for(int i=0;i<n-1;i++){
        int x;cin>>x;
        ptr->next = new Node(x);
        ptr = ptr->next;
    }
    int y;cin>>y;
    Node *ptr2 = (y>=0?head:nullptr);
    for(int i=0;i<y;i++){
        ptr2 = ptr2->next;
    }
    ptr->next = ptr2;
    return head;
}


int main() {
    Node* head = input();
    cout<<Solve(head)<<endl;
    return 0;
}

// Sample input
// 5
// 1 2 3 4 5
// 2
// output - pointer to node with value 3
```

→ **Given a singly linked list, determine if it's palindrome .**

KTH ORDER STATISTIC

Given an array **A** of **N** distinct numbers and a number. The challenge is to find **N^{0.5}**-th smallest number in the array, i.e., order statistic.

Input
First line contains one number N, Assume N to be a perfect square.
Second line contains N space separated integers. (A[1], A[2], -----, A[N]).

Output
Output one integer (Kth smallest element).

Limits
1 <= K <= N <= 1,000,000
0 <= Arr[i] <= 1,000,000,000
Expected time complexity -> O(n)
Expected space complexity -> O(n)

Sample 1 :-
Input
9
1 2 3 4 5 6 7 8 11
Output
3