CSE3032 - Competitive Programming
WIN SEM (2022-2023) AMR
Class Number: AP2022236001007
Slot: L11+L12+L19+L20
ASSIGNMENT - 8
Last Date for Submission: Thursday (08-04-2023) @ 12.40PM

Name: Chilamakuru Sai Venkata Yaswanth

Reg No: 20BCI7076

Write the program using (C / C++ / Java / Python) to solve the following problems.

Concept: Recursion and Dynamic Programming

S.No	Problem Name	Statement
1	Magic Index	A magic index in an array A [0 ••• n -1] is defined to be an index such that A[i] = i. The elements in the array can be negative. The elements in the array can be repeated multiple times. There can be more than one magic index. Given a sorted array of distinct integers, write a method to find a magic index, if one exists, in array A. In sample input First line is indicating how many test cases. Second line is indicating number of elements and corresponding array elements Example Sample Input 1: 1 5 -5 -1 2 1 9 Sample Output 1: 2 Explanation For Sample Input 1: The output is 2 because A[2] = 2 and hence 2 is the magic index. Sample Input 2: 2 5 2 3 4 5 6 6 -1 -1 -1 4 4 4 Sample Output 2 : -1 4

2	Matching of Parenthesis	Implement an algorithm to print all valid (e.g., properly opened and closed) combinations of n pairs of parentheses Input: 3 Output: ((())) , (() ()) , (()) () , () (()) , () () () Extension: Multi symbol matching: {()} Input: 3 {}(), {()}, (){}, ({})
3	Triple Step	A lion is running up a staircase with n steps and can hop either 1 step, 2 steps, or 3 steps at a time, implement a method to count how many possible ways the lion can run up the stairs. Write the Program code for both Top-down approach and Bottom-up approaches.

Note:

- If Code similarity is found, assignment will not be considered and Zero (0) Marks will be awarded.
- You have to upload a single document consisting of all the above programs and corresponding Output.

1) Magic Index

Code:

```
import java.util.*;
public class Assignments{
   public static void MI(int arr []){
          ArrayList<Integer> a=new ArrayList<>();
          for(int i=0;i<arr.length;i++){
                if(arr[i]==i){
                      a.add(arr[i]);
                }
          }
          if(a.size()==0){
                System.out.println("-1");
          }
          else{
                for(int i=0;i<a.size();i++){
                      System.out.print(a.get(i)+" ");
                }
          }
   }
   public static void main(String [] args){
          Scanner sc=new Scanner(System.in);
```

```
            int n=sc.nextInt();
            while(n>0){
                int a=sc.nextInt();
                int arr[]=new int[a];
                for(int i=0;i<a;i++) {
                    arr[i]=sc.nextInt();
                }
                MI(arr);
                n--;
            }
    }
}
```

Output:

```
C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
1
5
-5 -1 2 1 9
2
C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
2
5
2 3 4 5 6
-1
6
-1 -1 -1 4 4 4
4
C:\Users\Windows\OneDrive\Documents\CP>
```

2) Matching of Parenthesis

Code:

```
import java.util.ArrayList;
import java.util.*;
public class Assignments{
    public static List<String> paren(int n){
        List<String> r=new ArrayList<>();
        generate("",n,n,r);
        return r;
    }
    public static void generate(String s,int left,int right,List<String> r) {
        if(left==0 && right==0){
            r.add(s);
            return;
```

```
        }
        if(left>0){
            generate(s + "(",left-1,right,r);
        }
        if(right>left){
            generate(s + ")",left,right-1,r);
        }
    }
    public static void main(String [] args){
        int n=3;
        List<String> r=paren(n);
              for(int i=0;i<r.size();i++){
        System.out.print(r.get(i)+" ");
              }
    }
}
```

Output:

```
C:\Users\Windows\OneDrive\Documents\CP>javac Assignments.java

C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
((())) (()()) (())() ()(()) ()()()
C:\Users\Windows\OneDrive\Documents\CP>
```

3) Triple Step

Code:

```
import java.util.*;
public class Assignments{
	static int[][] arr;
	static int s(int[] count,int sum,int n){
		arr=new int[n+1][sum+1];
		for(int [] row:arr){
			Arrays.fill(row,-1);
		}
		int a=lion(count,sum,n);
		return a;
   }
	static int lion(int [] count,int sum,int n){
		if(sum==0){
			return 1;
		}
		if(n==0 || sum<0){
			return 0;
		}
		if(arr[n][sum]!=-1){
			return arr[n][sum];
		}
		arr[n][sum]=lion(count,sum-count[n-1],n)+lion(count,sum,n-1);
		return arr[n][sum];
   }
	static int bottomup(int[] count,int sum,int n){
		int[][] dp=new int[n+1][sum+1];
		for(int i=0;i<=n;i++){
			dp[i][0]=1;
		}
		for(int i=1;i<=sum;i++){
			dp[0][i]=0;
		}
		for(int i=1;i<=n;i++){
			for(int j=1;j<=sum;j++){
				if(count[i-1]<=j){
```

```
                                        dp[i][j]=dp[i][j-count[i-1]]+dp[i-1][j];
                                }
                                else{
                                        dp[i][j]=dp[i-1][j];
                                }
                        }
                }
                return dp[n][sum];
    }
        public static void main(String [] args){
                Scanner sc=new Scanner(System.in);
                int n=3;
                int step[]={1,2,3};
                System.out.print("N:");
                int N=sc.nextInt();
                System.out.println("Top down:"+ s(step,N,n));
                System.out.println("Bottom up:"+ s(step,N,n));
    }
}
```

Output:

```
C:\Users\Windows\OneDrive\Documents\CP>javac Assignments.java

C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
N:3
Top down:3
Bottom up:3

C:\Users\Windows\OneDrive\Documents\CP>
```