**TASK 1 JOHNRAY ONDIKO**

1.Given an array K with N integers from 1 to N+1 such that the array has exactly one integer missing, write a Java function that returns the missing integer. e.g. given K = [3,5,4,1], the function should return 2

class Main {

    // Function to RETURN missing number

    static int getMissingNo(int k[], int n)

    {

        int i, total;

        total = (n + 1) \* (n + 2) / 2;

        for (i = 0; i < n; i++)

            total -= k[i];

        return total;

    }

    /\* program to test above function \*/

    public static void main(String args[])

    {

        int k[] = { 3, 5, 4, 1, };

        int miss = getMissingNo(k, 4);

        System.out.println(miss);

    }

}

**2.** Given a string S of length N, write a Java function that transforms the string by reversing characters in groups of four, and returns the transformed string. e.g. when S = 'Lorem at' the output should be 'eroLta m' when S = ' Tempor ip' the output should be 'meT roppi'

class Main

{

    // Function to reverse a string in Java by Unicode

    // Right-to-left Override (RLO) character

    public static String reverse(String str)

    {

        return "\u202E" + str;

    }

    public static void main (String[] args)

    {

        String str = "Tempor ip";

String str= "meT roppi";

        // string is immutable

        str = reverse(str);

        System.out.println("Reverse of the given string is : " + str);

    }

}

**3.** ) A mouse is looking over a river that's 21 feet wide. There are 20 stones placed in the river 1 foot apart, such that there is 1 foot from the beginning of the river to the first stone, and 1 foot from the first stone to the next stone, and so on; and 1 foot from the last stone to the end of the river. The mouse can jump over 1 foot or 2 feet such that the first step the mouse can make is either from the beginning of the river to the first stone, or from the beginning of the river to the second stone. If the mouse can't jump backwards, write a Java function that returns the number of different ways the mouse can jump from the beginning of the river to the end of the river.

// Javaprogram to find Minimum

// number of jumps to reach end

import java.util.\*;

import java.io.\*;

class GFG {

    // Returns minimum number of

    // jumps to reach arr[h] from arr[l]

    static int minJumps(int arr[], int l, int h)

    {

        // Base case: when source

        // and destination are same

        if (h == l)

            return 0;

        // When nothing is reachable

        // from the given source

        if (arr[l] == 0)

            return Integer.MAX\_VALUE;

        // Traverse through all the points

        // reachable from arr[l]. Recursively

        // get the minimum number of jumps

        // needed to reach arr[h] from these

        // reachable points.

        int min = Integer.MAX\_VALUE;

        for (int i = l + 1; i <= h

                            && i <= l + arr[l];

             i++) {

            int jumps = minJumps(arr, i, h);

            if (jumps != Integer.MAX\_VALUE && jumps + 1 < min)

                min = jumps + 1;

        }

        return min;

    }

4. a) Use the diagram above to create the tables (the database, table and column names should remain as indicated) b) Write a MySQL query that will display the number of students per course per institution in the format below

CREATE DATABASE education;

CREATE TABLE student (   
    studentid int(11),  
    name varchar(250),  
    course int(11)

);

CREATE TABLE course (  
    courseid int(11),  
    name varchar(250),  
    institution int(11)  
);

CREATE TABLE institution (  
    institutionid int(11),  
    name varchar(250),  
    );

SELECT \* FROM institution,

WHERE institution = ‘University College Dublin’ AND course= ‘Bsc Acturial Science’;

SELECT \* FROM institution,

WHERE institution = ‘University College London’ AND course= ‘MPhil Genomics’;