Language used for coding :- JAVA

Answers to the coding test( as mentioned in the test window : pick one question from easy, one from medium and one from hard)

EASY Q1 Given a string s consisting of words and spaces, return the length of the last word in the string.

A word is a maximal

substring consisting of non-space characters only.

Code:-

class Main {

    public static void main(String args[]) {

        // Your code goes here

    Scanner sc = new Scanner(System. in );

    System.out.println("Enter your name");

    String s = sc.nextLine();

    String[] str = s.split("\\s+");

    String lastWord = str[str.length - 1];

    System.out.print(lastWord.length());

    }

}

EASY Q3 Given an integer numRows, return the first numRows of Pascal's triangle.

In Pascal's triangle, each number is the sum of the two numbers directly above it as shown

CODE:-

public static ArrayList<ArrayList<Integer>> generatePascalsTriangle(int numRows) {

        for (int i = 0; i < numRows; i++) {

            ArrayList<Integer> row = new ArrayList<>();

            for (int j = 0; j <= i; j++) {

                if (j == 0 || j == i) {

                    row.add(1);

                } else {

                    int prevRowLeft = triangle.get(i - 1).get(j - 1);

                    int prevRowRight = triangle.get(i - 1).get(j);

                    row.add(prevRowLeft + prevRowRight);

                }

            }

        triangle.add(row);

        }

         return triangle;

    }

s

MEDIUM

Q2 Given an integer array of size n, find all elements that appear more than ⌊ n/3 ⌋ times.

Code:-

public class Solution {

    public int repeatedNumber(int[] A) {

        int n=A.length;

        int majNum1 = A[0];

        int majNum2 = 0;

        int cnt1 = 1;

        int cnt2 = 0;

        if(n <= 2){

            return A[0];

        }

        for(int i = 1 ; i < n ; i++){

            if(A[i] == majNum1){

                cnt1++;

            } else if(A[i] == majNum2){

                cnt2++;

            } else {

                if(cnt1 == 0){

                    majNum1 = A[i];

                    cnt1++;

                } else if (cnt2 == 0){

                    majNum2 = A[i];

                    cnt2++;

                } else {

                    cnt1-- ; cnt2-- ;

                }

            }

        }

        int freq = 0;

        for(int i = 0 ; i < n ; i++){

            if(A[i] == majNum1){

                freq++;

            }

        }

        if (freq> A.length / 3){

            return majNum1;

        }

        freq = 0;

        for(int i = 0 ; i < n ; i++){

            if(A[i] == majNum2){

                freq++;

            }

        }

        if (freq > A.length / 3){

            return majNum2;

        }

        return -1;

    }

}

HARD Q2:-

You are given a string s. You can convert s to a palindrome by adding characters in front of it. Return *the shortest palindrome you can find by performing this transformation*.

Code:-

public static String shortestPalindrome(String s) {

        int n = s.length();

        int i = n - 1;

        while (i >= 0 && !isPalindrome(s.substring(0, i + 1))) {

            i--;

        }

        StringBuilder sb = new StringBuilder();

        for (int j = n - 1; j > i; j--) {

            sb.append(s.charAt(j));

        }

        sb.append(s);

        return sb.toString();

    }

    public static boolean isPalindrome(String s) {

        int n = s.length();

        for (int i = 0; i < n / 2; i++) {

            if (s.charAt(i) != s.charAt(n - i - 1)) {

                return false;

            }

        }

        return true;

    }