JAVA Hacker Ranker Challenge

1. You are given a date. You just need to write the method, getDay, which returns the day on that date.

public static String findDay(int month, int day, int year) {

Calendar cal = Calendar.getInstance();

cal.set(Calendar.MONTH, month-1);

cal.set(Calendar.DAY\_OF\_MONTH, day);

cal.set(Calendar.YEAR, year);

String[] day\_of\_week = {"SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY","SATURDAY"};

return day\_of\_week[cal.get(Calendar.DAY\_OF\_WEEK)-1];

}

2. The challenge here is to read n lines of input until you reach EOF, then number and print all n lines of content.

public class Solution {

public static void main(String[] args)

{

Scanner in= new Scanner(System.in);

int i=1;

while(in.hasNext())

{

String s=in.nextLine();

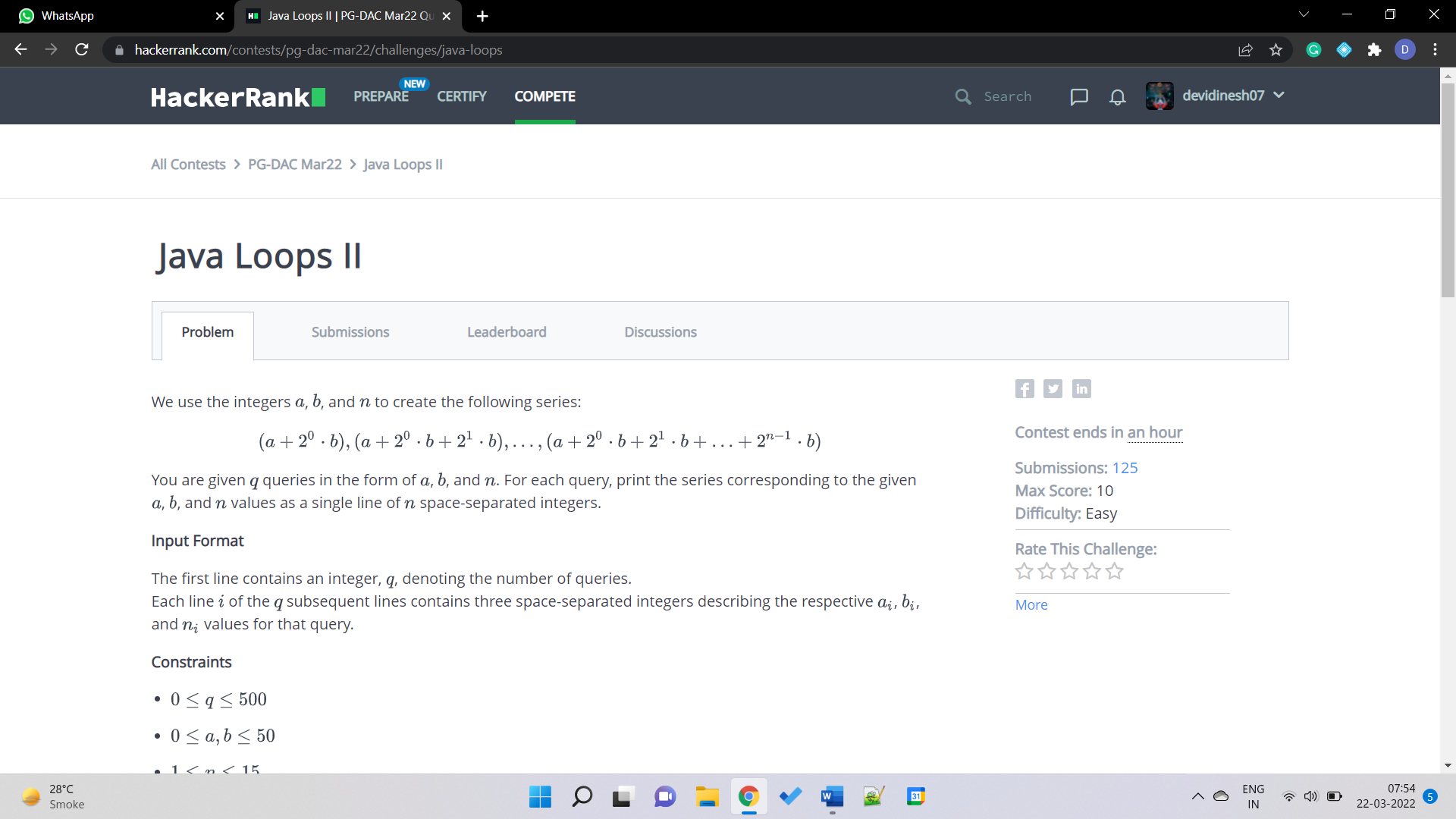
System.out.println(i++ +" "+s);

}

}

}

3.



class Solution{

public static void main(String []argh){

Scanner in = new Scanner(System.in);

int t=in.nextInt();

for(int i=0;i<t;i++)

{

int a = in.nextInt();

int b = in.nextInt();

int n = in.nextInt();

int sum=a;

for(int j=0; j<n; j++)

{

sum+=(int)((Math.pow(2,j)\*b));

System.out.print(sum+" ");

}

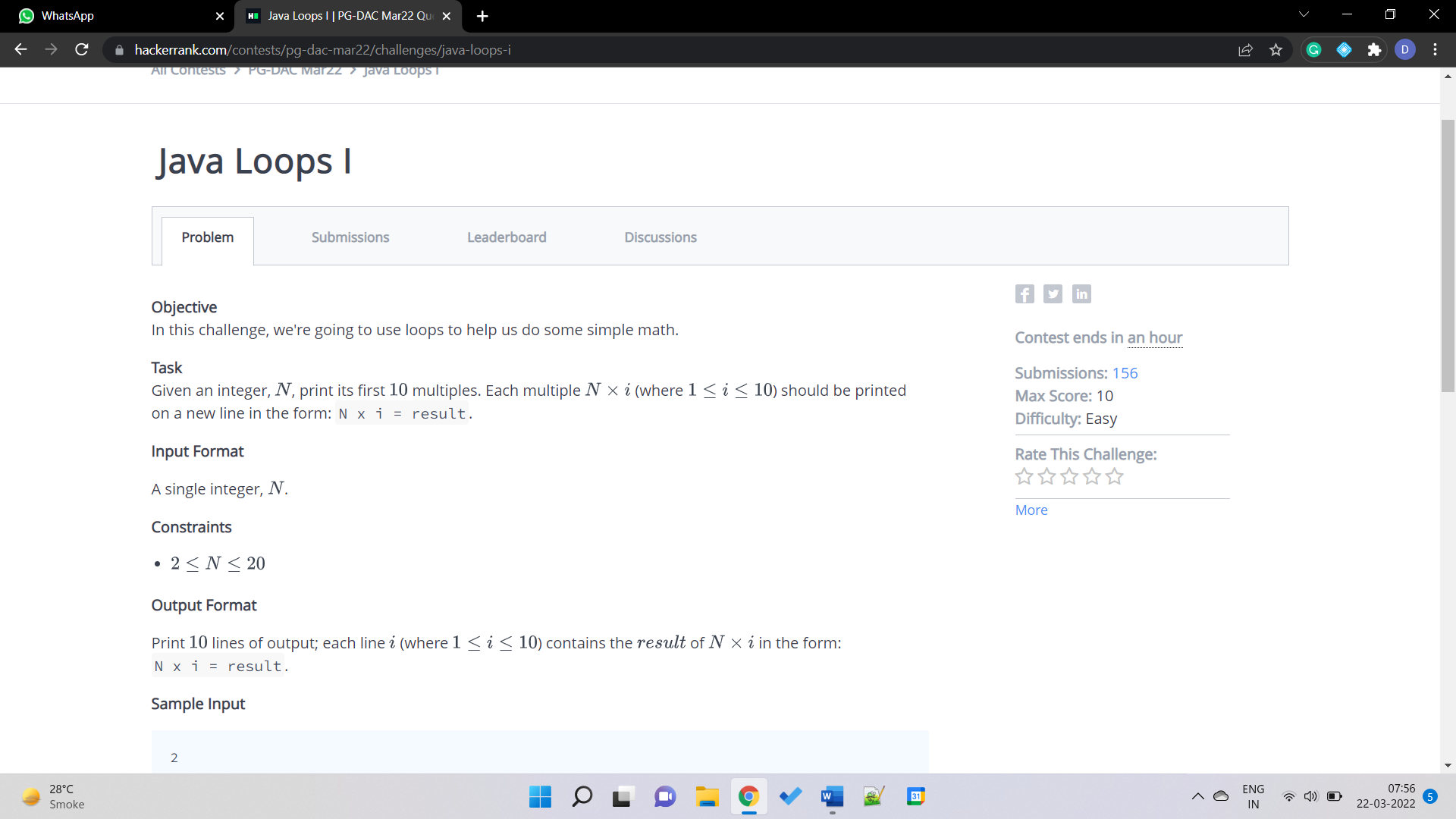
System.out.println();

}

in.close();

}

}

4. 

public class Solution {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

int N = Integer.parseInt(bufferedReader.readLine().trim());

for(int i=1; i<=10; i++)

{

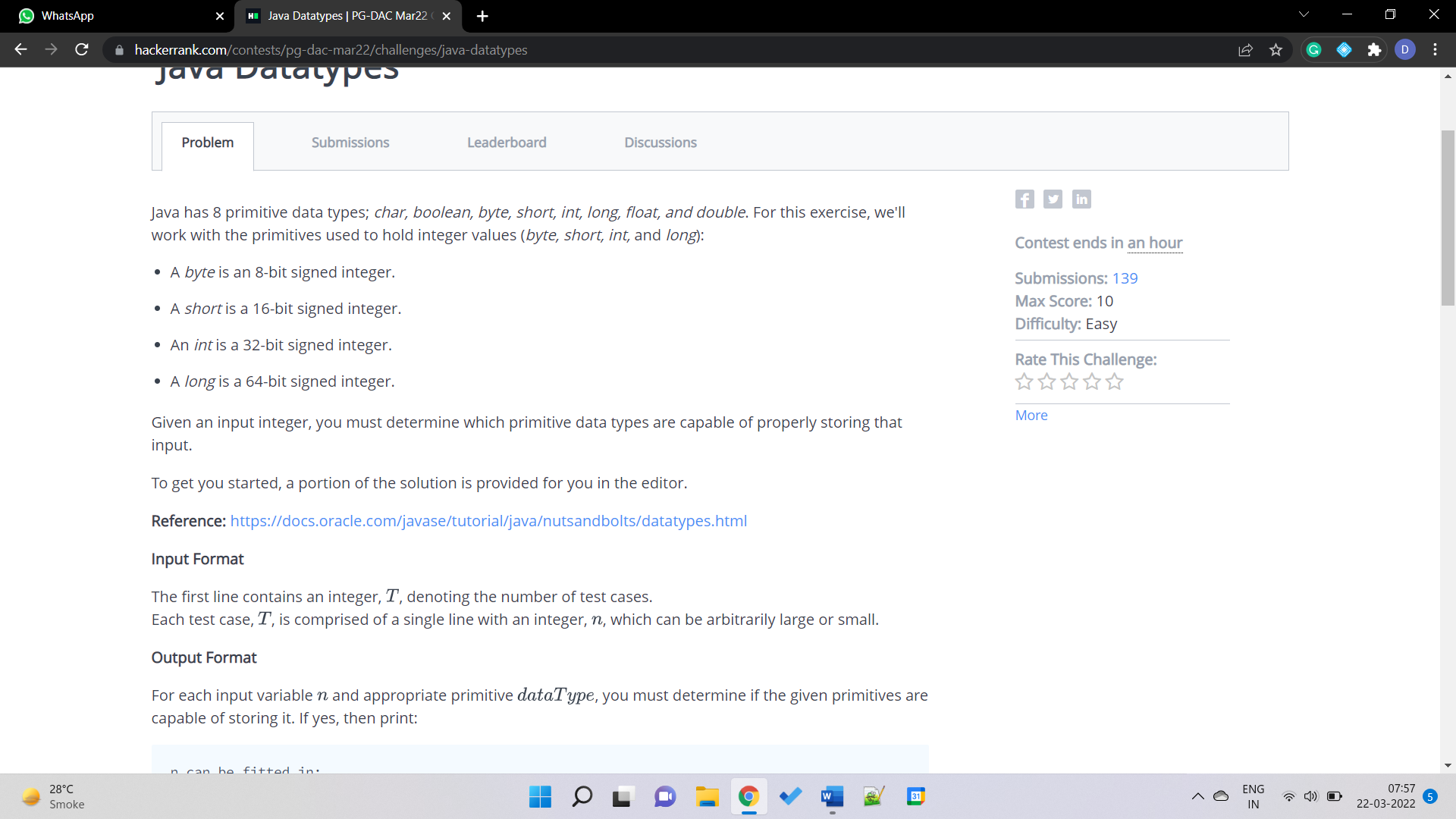
System.out.println(N+" x "+i+" = "+(N\*i));

}

bufferedReader.close();

}

}

5. 

class Solution{

public static void main(String []argh)

{

Scanner sc = new Scanner(System.in);

int t=sc.nextInt();

for(int i=0;i<t;i++)

{

try

{

long x=sc.nextLong();

System.out.println(x+" can be fitted in:");

if(x>=-128 && x<=127)

System.out.println("\* byte");

if(x>=(-1\*Math.pow(2,15))&& x<=(Math.pow(2,15)-1))

System.out.println("\* short");

if(x>=(-1\*Math.pow(2,31))&& x<=(Math.pow(2,31)-1))

System.out.println("\* int");

if(x>=(-1\*Math.pow(2,63))&& x<=(Math.pow(2,63)-1))

System.out.println("\* long");

}

catch(Exception e)

{

System.out.println(sc.next()+" can't be fitted anywhere.");

}

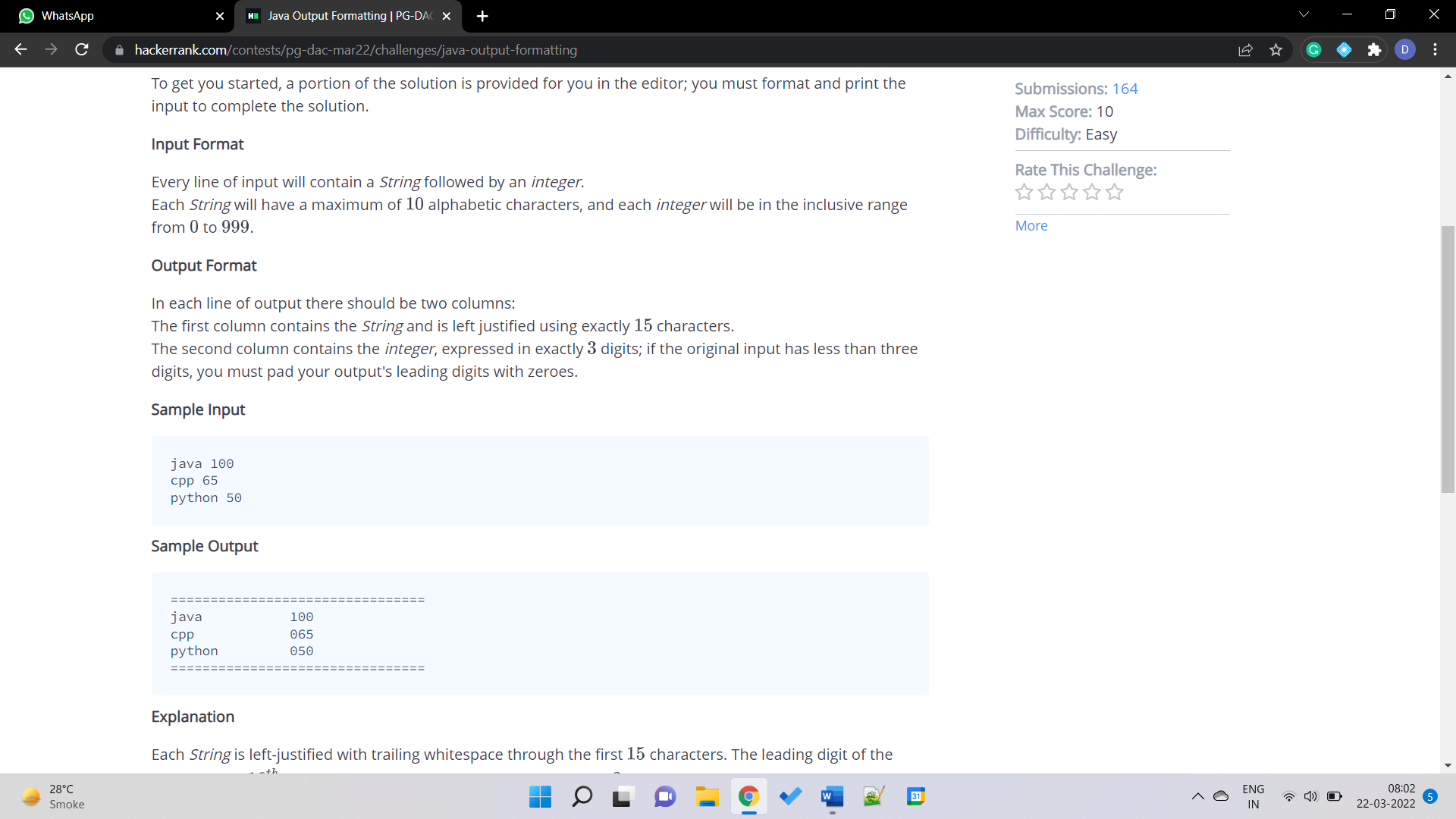
}

}

}

6. You are given an integer , you have to convert it into a string.

String s=Integer.toString(n);

7. 

public class Solution

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

int x[]=new int[3];

String s1[]=new String[3];

for(int i=0;i<3;i++)

{

s1[i]=sc.next();

x[i]=sc.nextInt();

}

System.out.println("================================");

for(int i=0;i<3;i++)

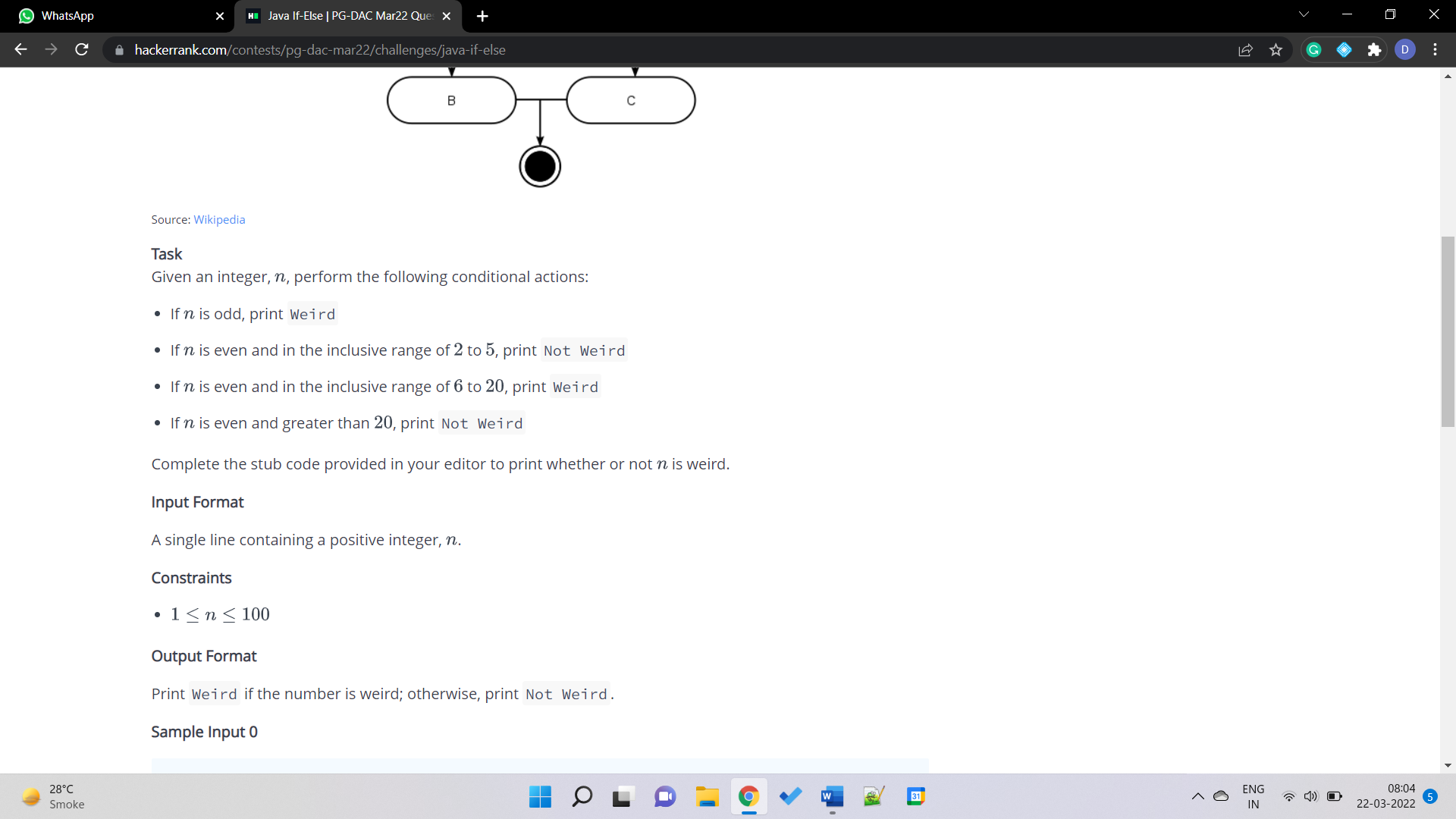
System.out.printf("%-15s%03d\n",s1[i],x[i]);

System.out.println("================================");

}

}

8.



public class Solution {

private static final Scanner scanner = new Scanner(System.in);

public static void main(String[] args)

{

int N = scanner.nextInt();

scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

if(N%2==1)

System.out.println("Weird");

else

{

if (N>=2 && N<=5)

System.out.println("Not Weird");

else if (N>=6 && N<=20)

System.out.println("Weird");

else

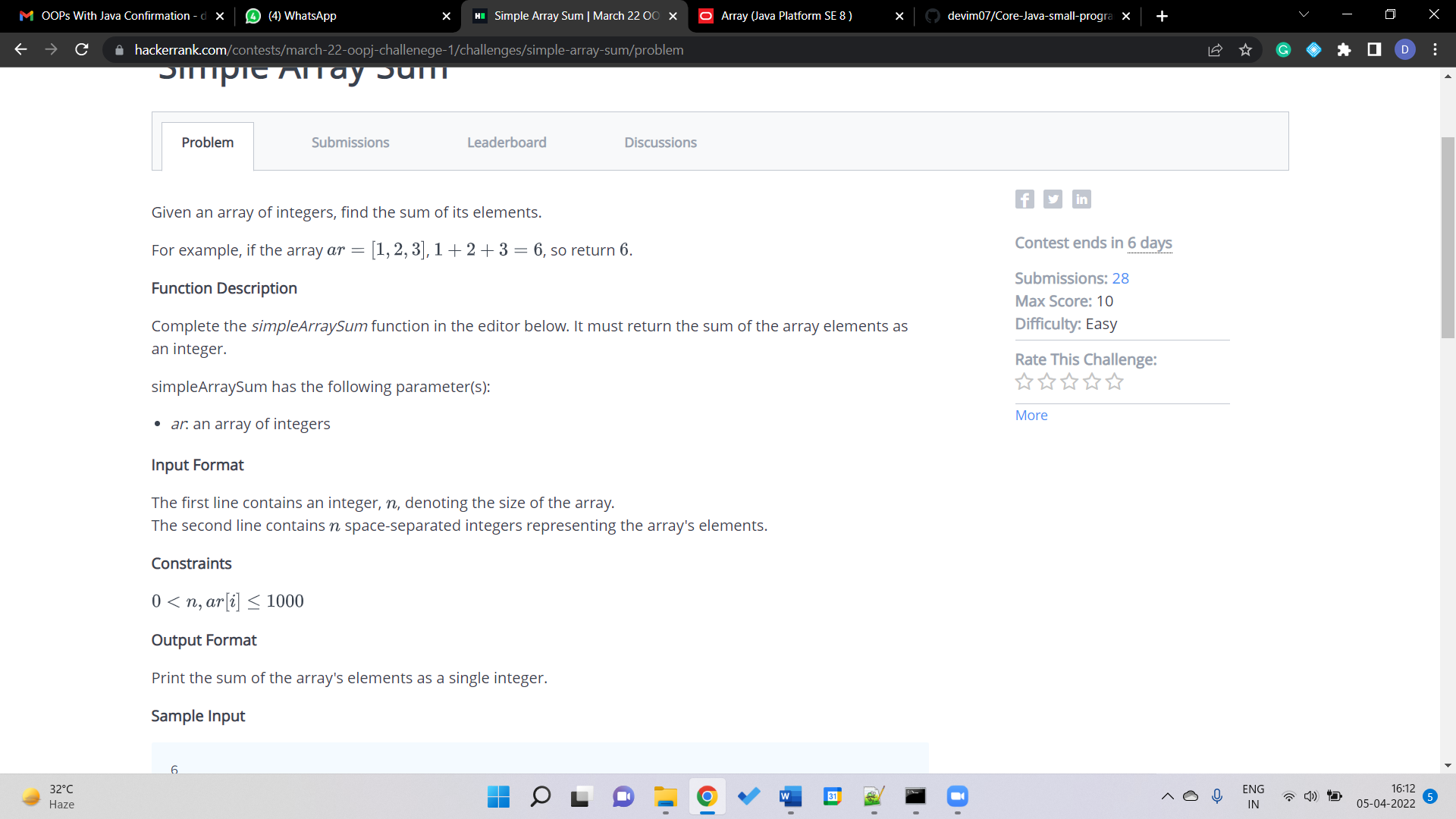
System.out.println("Not Weird");

}

scanner.close();

}

}

9. 

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.regex.\*;

class Result {

public static int simpleArraySum(int n, int... array) {

int sum=0;

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

sum+=array[i];

}

return sum;

}

}

public class Solution {

public static void main(String[] args) throws IOException {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int arr[]=new int[n];

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

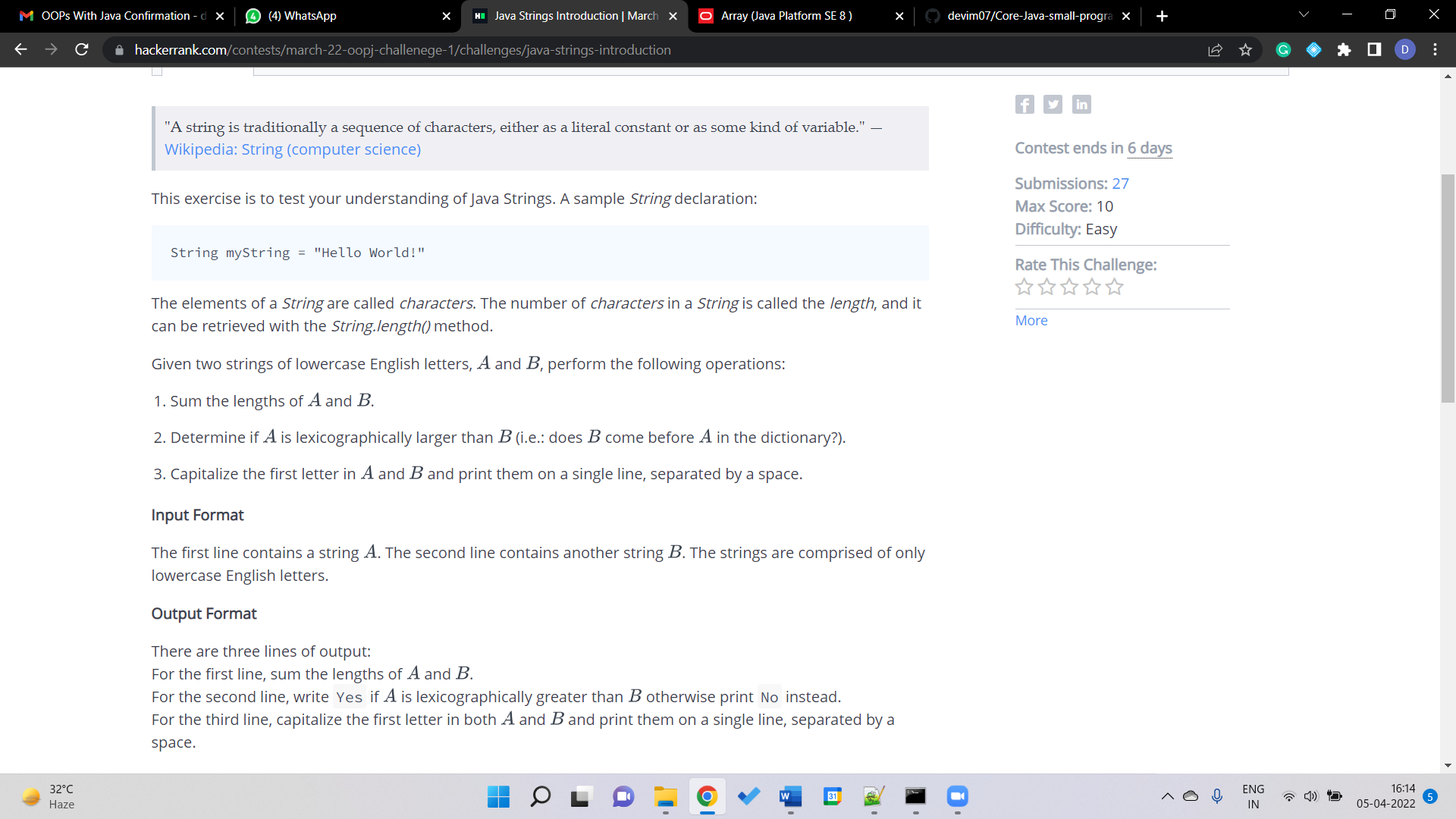
arr[i]=sc.nextInt();

}

System.out.println(Result.simpleArraySum(n, arr));

}

}

10. 

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String A=sc.next();

String B=sc.next();

System.out.println(A.length()+B.length());

int n= A.compareTo(B);

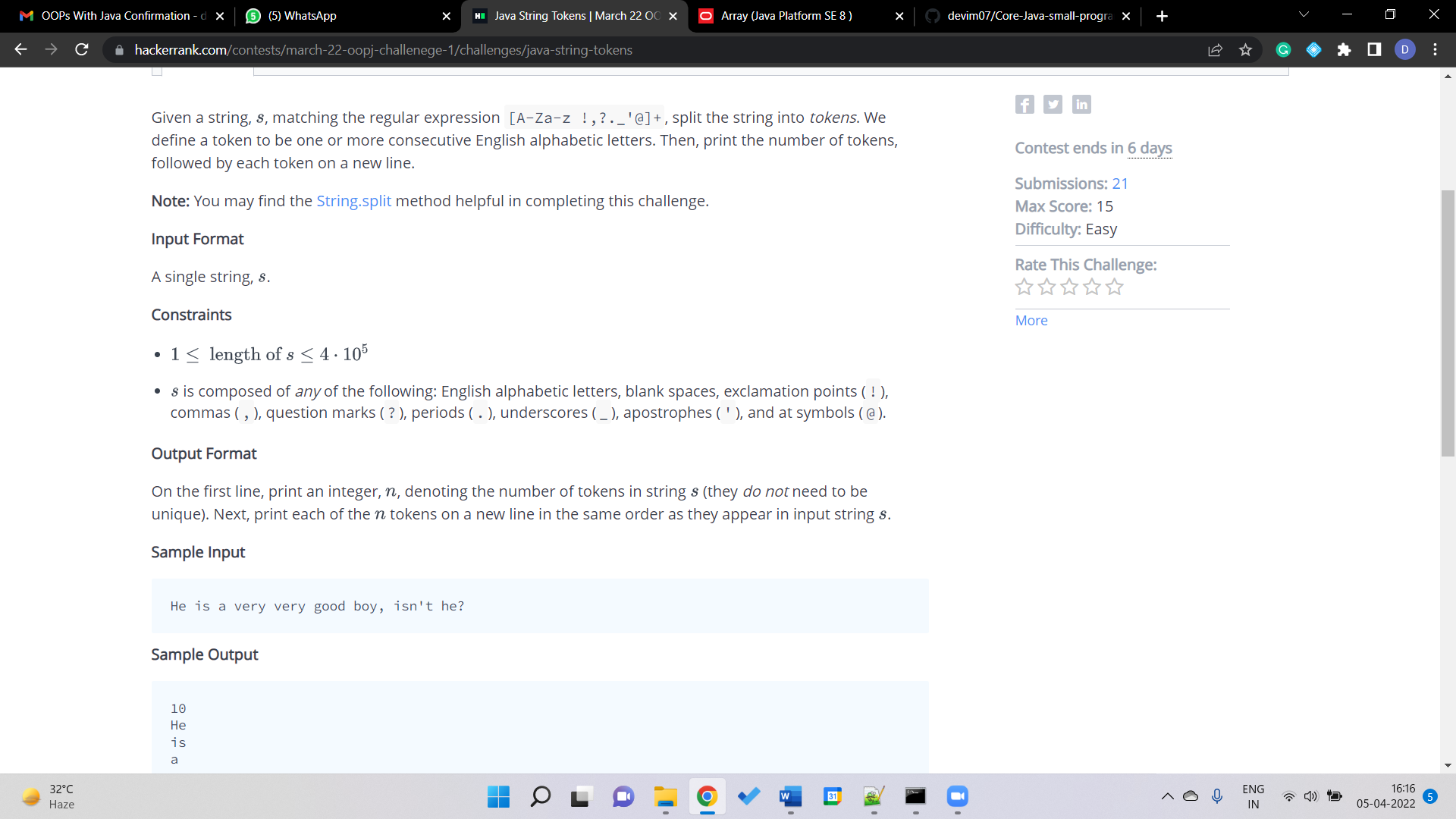
System.out.println((n>0)?"Yes":"No");

String S=A.substring(0, 1).toUpperCase() + A.substring(1) + " " + B.substring(0, 1).toUpperCase() + B.substring(1);

System.out.println(S);

}

}

11.

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

String s = scan.nextLine();

String arr[]=s.split("[^a-zA-Z]"); //split when not a-z or A-Z

String newarr[]=new String[arr.length];

int n=0;

for(int i=0; i<arr.length; i++){

if(arr[i].length()!=0){

newarr[n]=arr[i];

n++;

}

}

System.out.println(n);

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

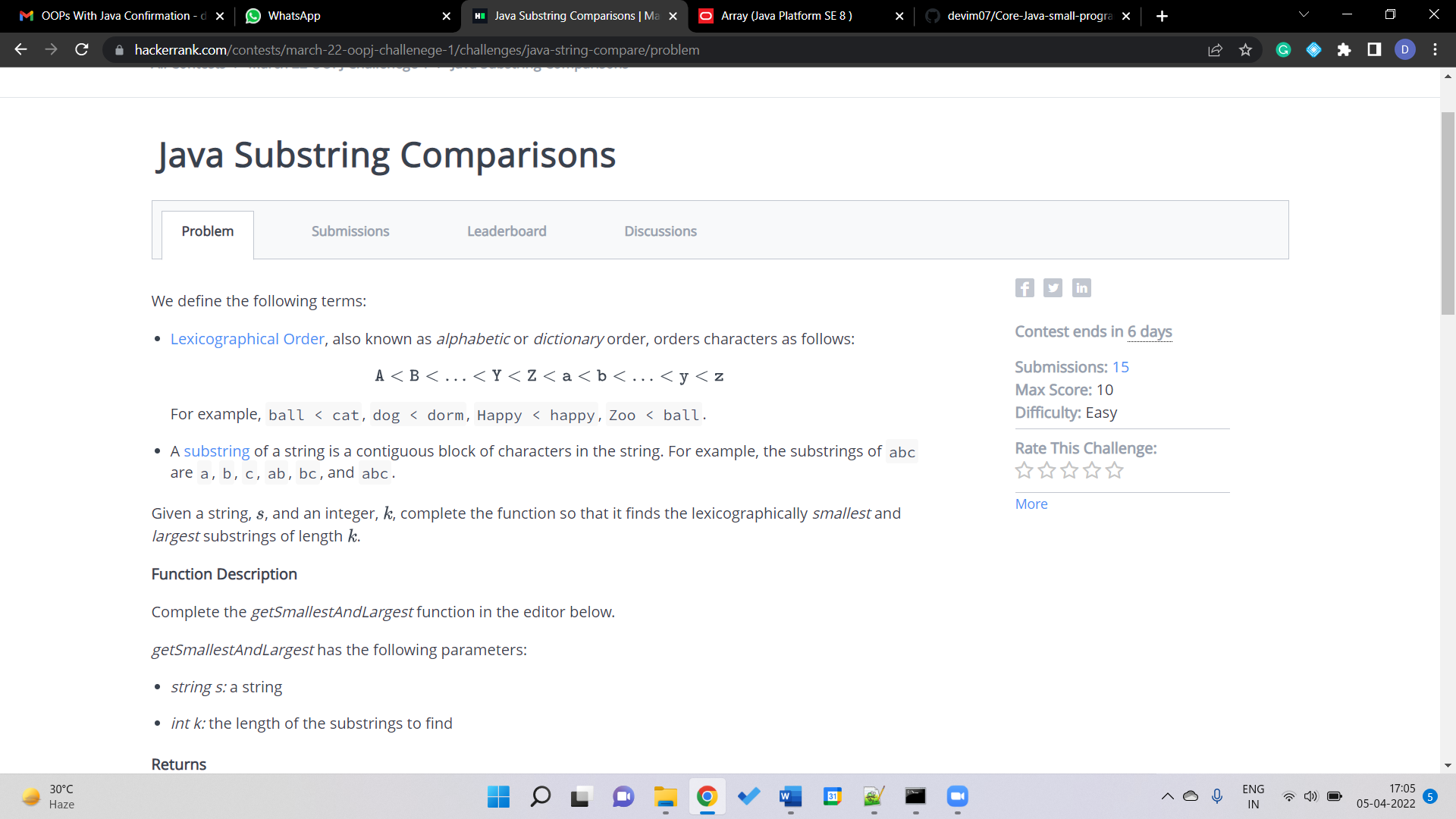
System.out.println(newarr[i]);

}

scan.close();

}

}

12. 

public static String getSmallestAndLargest(String s, int k) {

int n=s.length();

String a[]=new String[n];

for(int i=0; i<=n-k; i++){

a[i]=s.substring(i,(i+k));

}

String smallest = a[0];

String largest = a[0];

for(int i=1; i<=n-k; i++){

//System.out.println(a[i]);

if(smallest.compareTo(a[i])>0)

smallest=a[i];

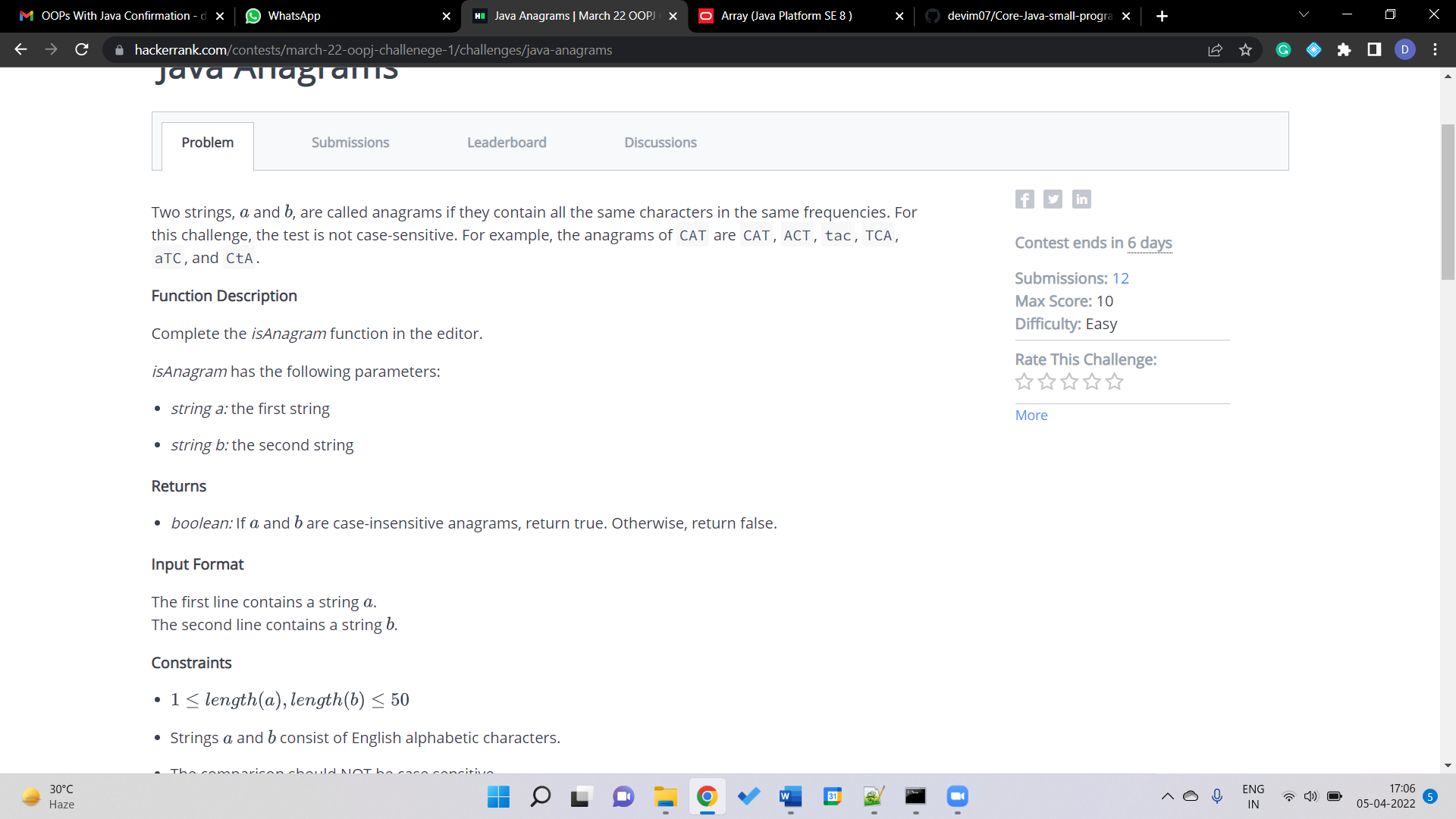
else if (largest.compareTo(a[i])<0)

largest=a[i];

}

return smallest + "\n" + largest;

}

13

static boolean isAnagram(String a, String b) {

if(a.length()!=b.length())

return false;

else{

a=a.toLowerCase();

b=b.toLowerCase();

char arr[]=a.toCharArray();

char brr[]=b.toCharArray();

char temp;

for(int i=0; i<a.length()-1; i++){

for(int j=i+1; j<a.length(); j++){

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

if(brr[i]>brr[j])

{

temp=brr[i];

brr[i]=brr[j];

brr[j]=temp;

}

}

}

a=String.valueOf(arr);

b=String.valueOf(brr);

if(a.equals(b))

return true;

else

return false;

}

}