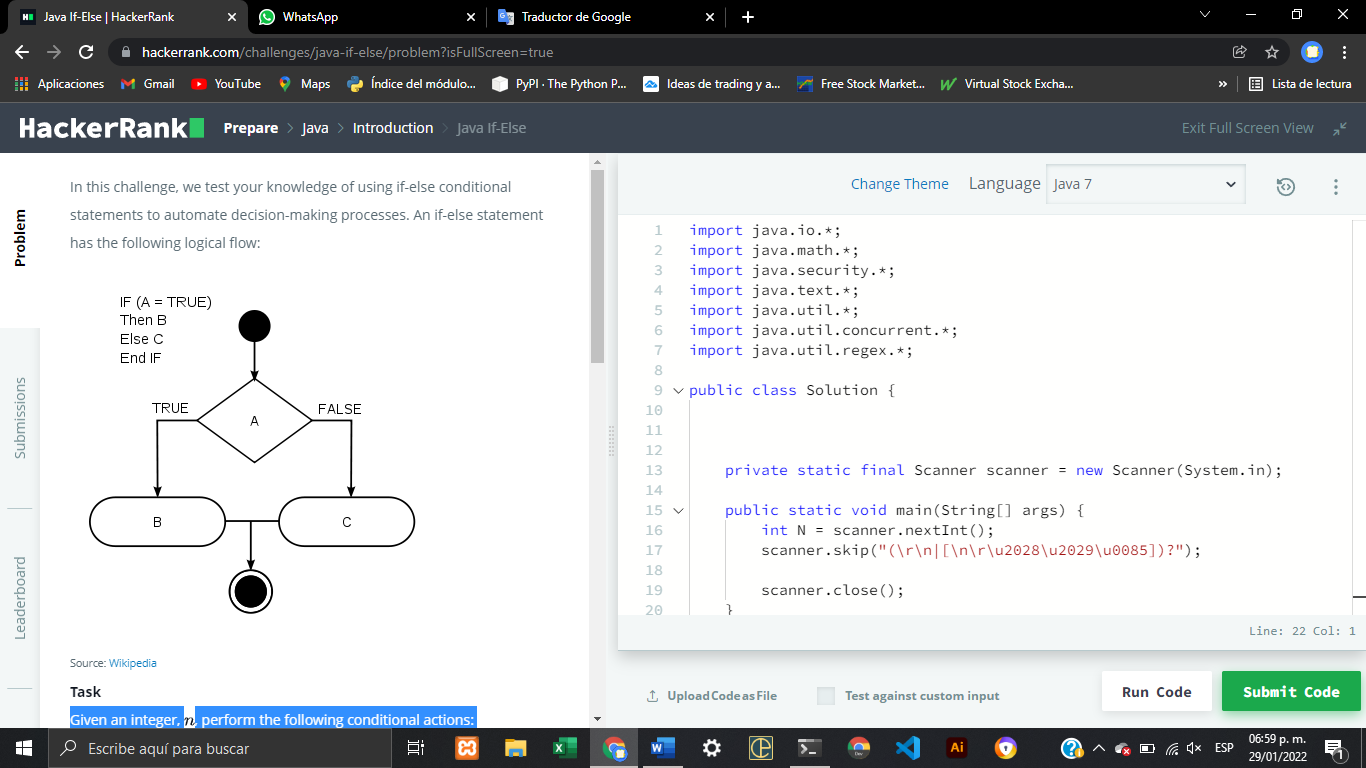
EJERCICIOS HACKER RANK



Given an integer, n, perform the following conditional actions:

* If  n is odd, print Weird

**n%2==1**

* If  n is even and in the inclusive range of  6 to20 , print Weird

**n%2==0 &&n>=6&&n<=20**

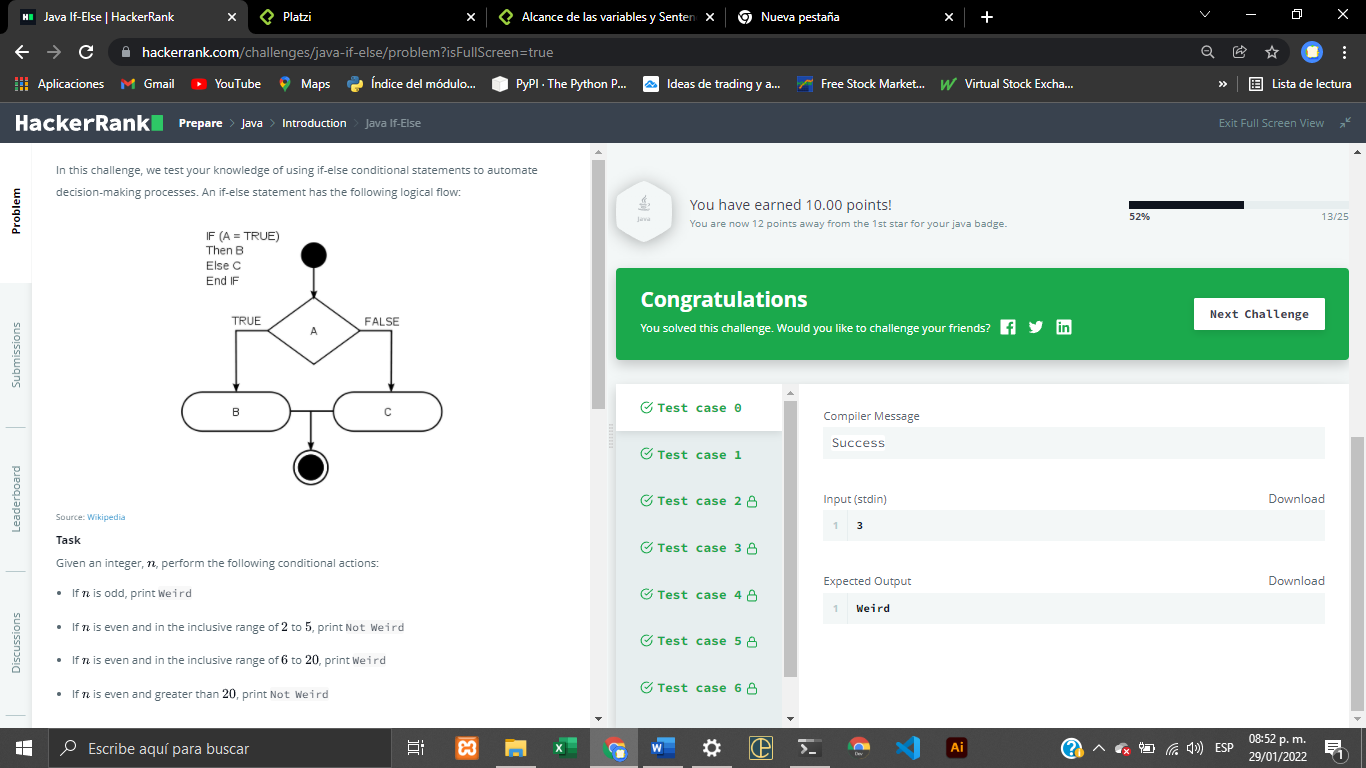
* If  n is even and in the inclusive range of  2 to 5 , print Not Weird

**n%2==0 && n>2 && n<5**

* If  n is even and greater than 20, print Not Weird

**n%2==0 &&n>20**

Complete the stub code provided in your editor to print whether or not n  is weird.



import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.regex.\*;

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])?");

        scanner.close();

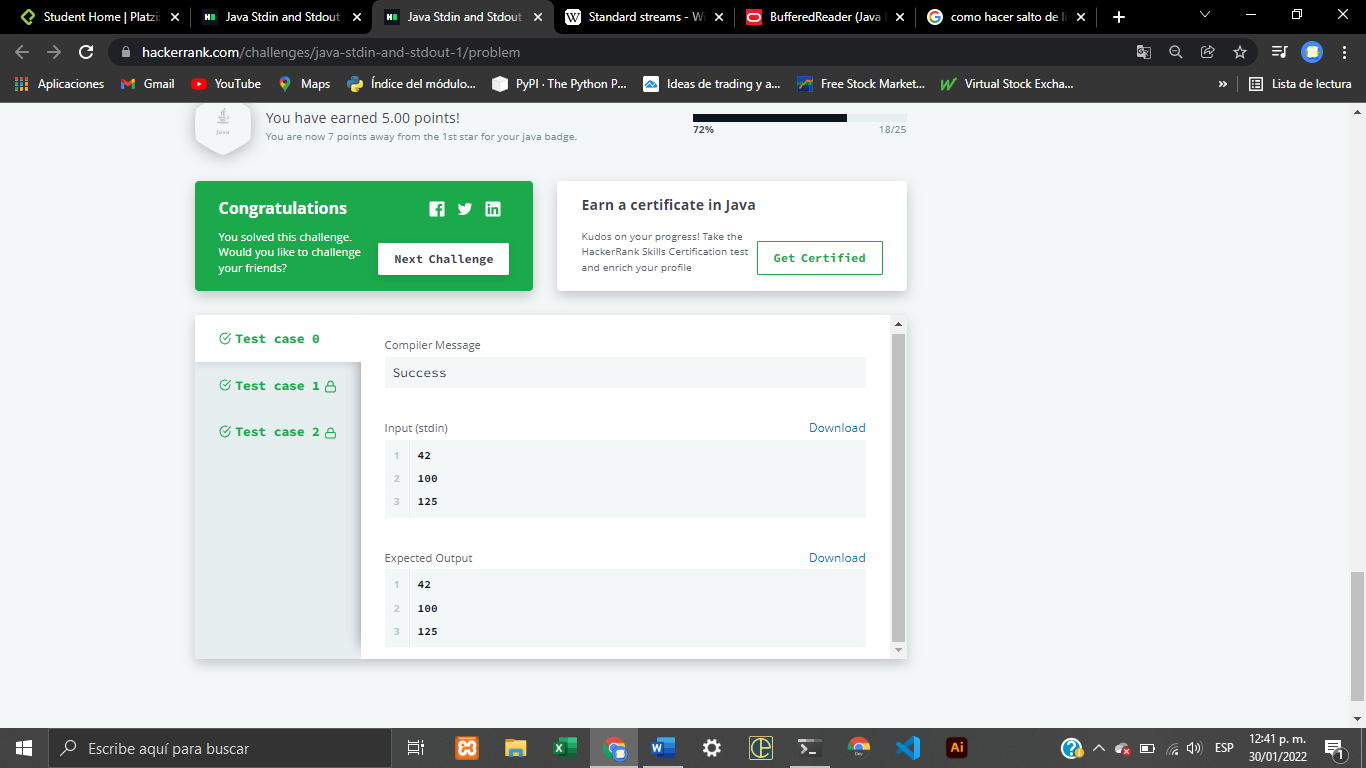
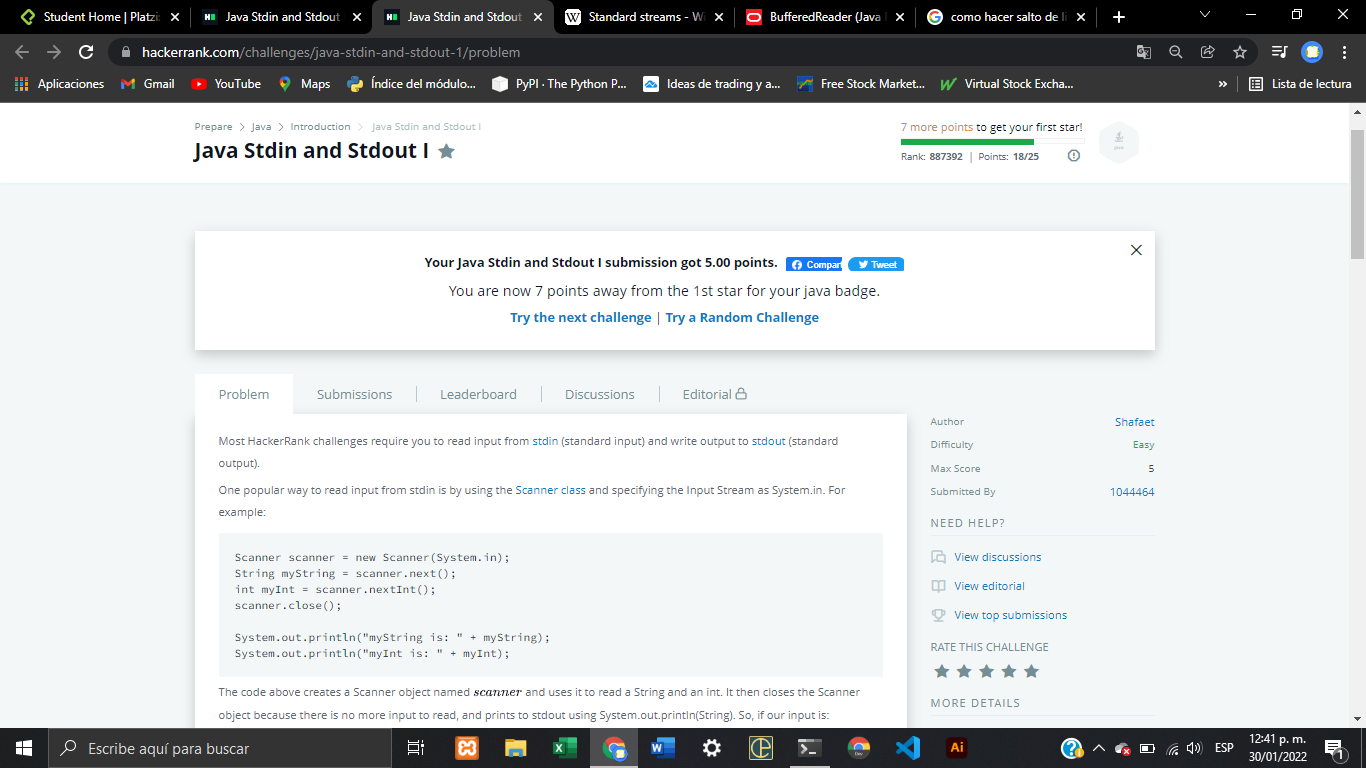
        if ((n%2==0 &&n>=6&&n<=20)||(n%2==1) ){System.out.println("Weird");

        }else{if((n%2==0 && n>2 && n<5)||(n%2==0 &&n>20 ) ){

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

    }}

}



import java.util.\*;

public class Solution {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int a = scan.nextInt();

        int b = scan.nextInt();

        int c = scan.nextInt();

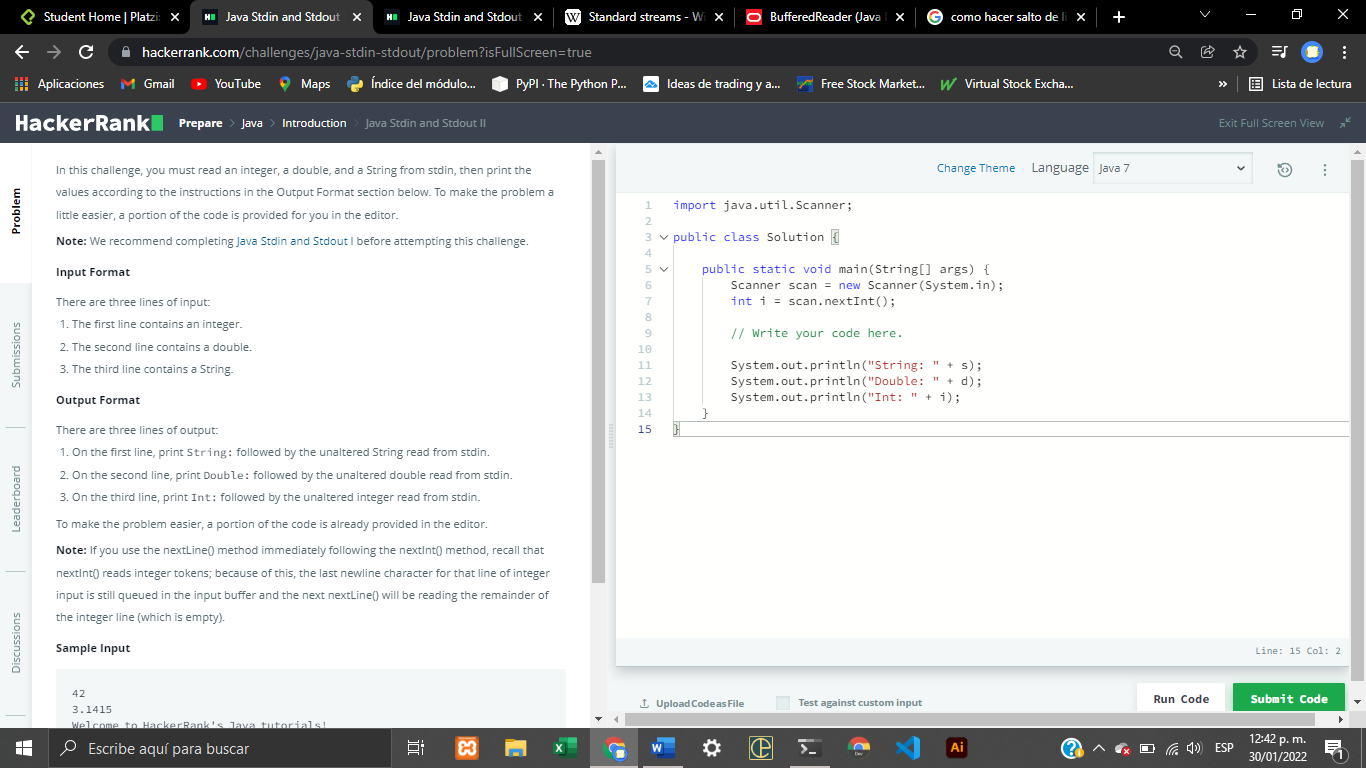
        System.out.println(a);

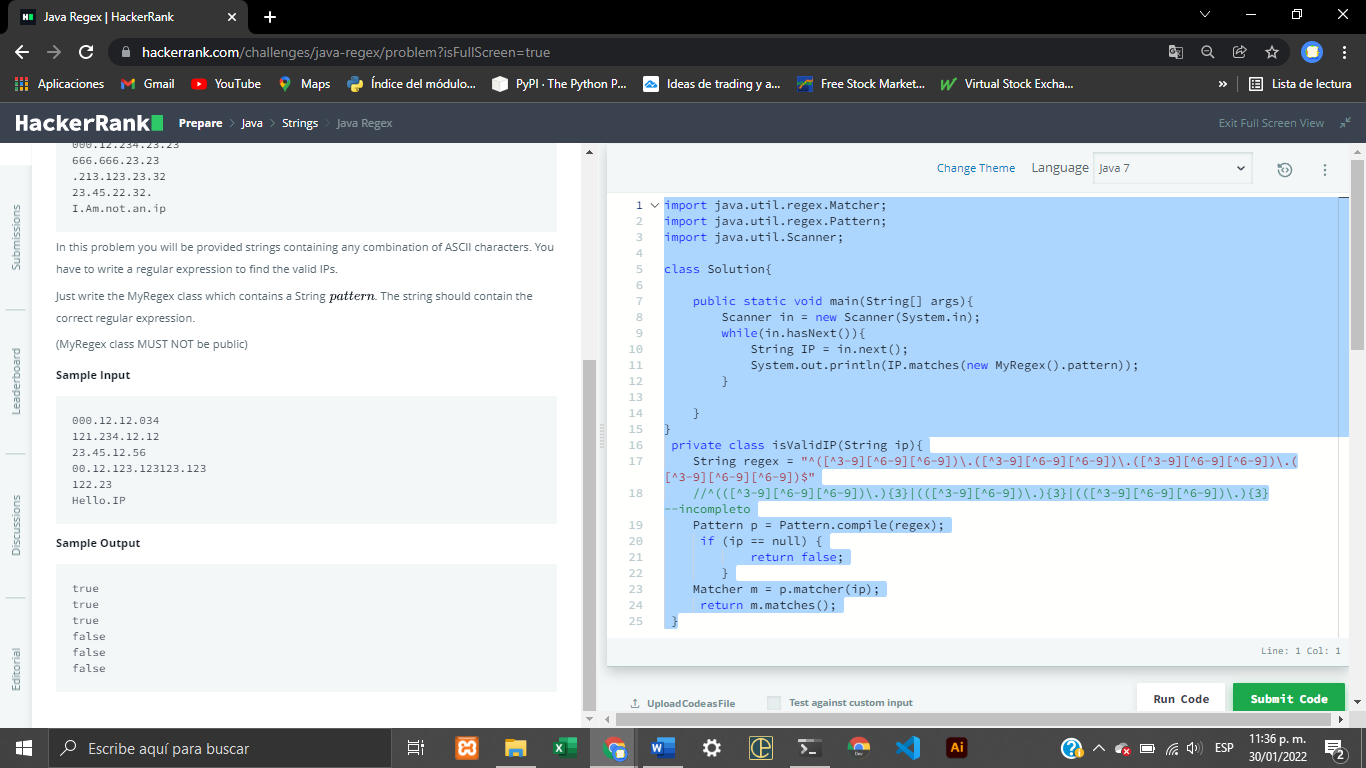
        System.out.println(b);

        System.out.println(c);

    }

}



import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.util.Scanner;

class Solution{

    public static void main(String[] args){

        Scanner in = new Scanner(System.in);

        while(in.hasNext()){

            String IP = in.next();

            System.out.println(IP.matches(new MyRegex().pattern));

        }

    }

}

 private class isValidIP(String ip){

    String regex = "^([^3-9][^6-9][^6-9])\.([^3-9][^6-9][^6-9])\.([^3-9][^6-9][^6-9])\.([^3-9][^6-9][^6-9])$"

    //^(([^3-9][^6-9][^6-9])\.){3}|(([^3-9][^6-9])\.){3}|(([^3-9][^6-9][^6-9])\.){3}--incompleto

    Pattern p = Pattern.compile(regex);

     if (ip == null) {

            return false;

        }

    Matcher m = p.matcher(ip);

     return m.matches();

 }