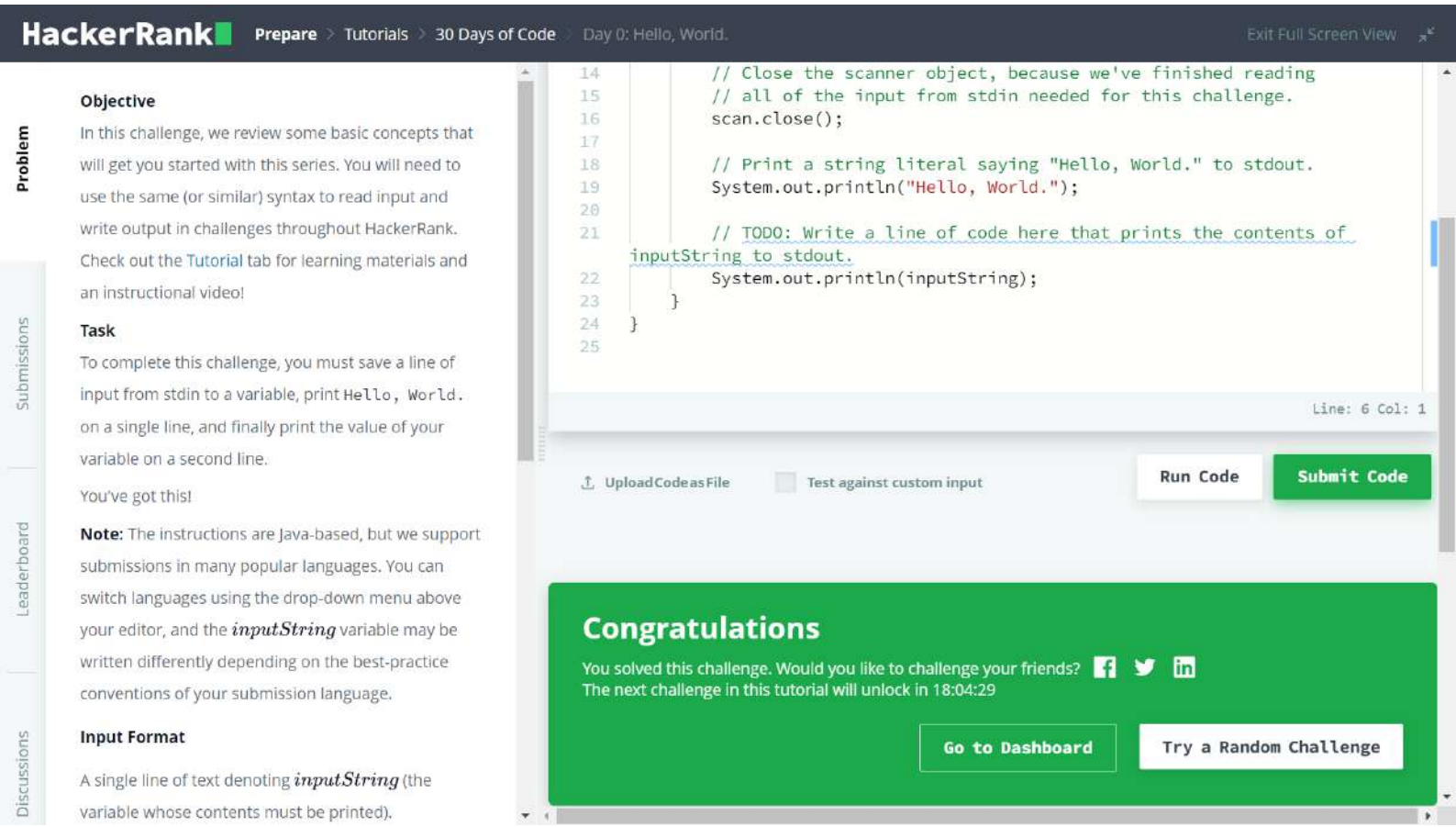
Reto HackerRank 30 días de código (día 0 a día 29)

Nombre: Luciano Manrique Medina

Correo: lumanriquem@unal.edu.co

Grupo: 9

Todos los retos fueron completados, 30 de 30 (del 0 al 29).


HackerRank
Prepare > Tutorials > 30 Days of Code > Day 0: Hello, World.
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
Objective
In this challenge, we review some basic concepts that will get you started with this series. You will need to use the same (or similar) syntax to read input and write output in challenges throughout HackerRank. Check out the Tutorial tab for learning materials and an instructional video!
Task
To complete this challenge, you must save a line of input from stdin to a variable, print Hello, World. on a single line, and finally print the value of your variable on a second line.
You've got this!
Note: The instructions are Java-based, but we support submissions in many popular languages. You can switch languages using the drop-down menu above your editor, and the inputString variable may be written differently depending on the best-practice conventions of your submission language.
Input Format
A single line of text denoting inputString (the variable whose contents must be printed).
// Close the scanner object, because we've finished reading
// all of the input from stdin needed for this challenge.
scan.close();
// Print a string literal saying "Hello, World." to stdout.
System.out.println("Hello, World.");
// TODO: Write a line of code here that prints the contents of inputString to stdout.
System.out.println(inputString);
}
}
Line: 6 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 18:04:29
Go to Dashboard
Try a Random Challenge

Código día 0:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
public class Solution {
 public static void main(String[] args) {
        // Create a Scanner object to read input from stdin.
  Scanner scan = new Scanner(System.in);

  // Read a full line of input from stdin and save it to our variable, inputString.
  String inputString = scan.nextLine();

  // Close the scanner object, because we've finished reading
        // all of the input from stdin needed for this challenge.
  scan.close();

  // Print a string literal saying "Hello, World." to stdout.
  System.out.println("Hello, World.");

    // TODO: Write a line of code here that prints the contents of inputString to stdout.
       System.out.println(inputString);
   }
}
```

Objective

Today, we're discussing data types. Check out the Tutorial tab for learning materials and an instructional video!

Task

Complete the code in the editor below. The variables i, d, and s are already declared and initialized for you. You must:

1. Declare 3 variables: one of type int, one of type double, and one of type String.
2. Read 3 lines of input from stdin (according to the sequence given in the Input Format section below) and initialize your 3 variables.
3. Use the $+$ operator to perform the following operations:
    1. Print the sum of i plus your int variable on a new line.
    2. Print the sum of d plus your double variable to a scale of one decimal place on a new line.
    3. Concatenate s with the string you read as input and print the result on a new line.

Note: If you are using a language that doesn't support using $+$ for string concatenation (e.g.: C), you can just print one variable immediately following the other on the same line. The string provided in your editor must be printed first, immediately followed by the string you read as input.

Input Format

```
21
22      System.out.println(sumaI+i);
23      System.out.println(sumaD+d);
24      System.out.println(s+conS);
25
26
27 >    scan.close();···
```

Line: 15 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 18:04:02

Go to Dashboard Try a Random Challenge

Código Día 1:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        int i = 4;
        double d = 4.0;
        String s = "HackerRank ";

        Scanner scan = new Scanner(System.in);


        int sumaI = scan.nextInt();
        double sumaD = scan.nextDouble();
        scan.nextLine();
        String conS = scan.nextLine();

        System.out.println(sumaI+i);
        System.out.println(sumaD+d);
        System.out.println(s+conS);


        scan.close();
    }
}
```

Objective

In this challenge, you will work with arithmetic operators. Check out the Tutorial tab for learning materials and an instructional video.

Task

Given the meal price (base cost of a meal), tip percent (the percentage of the meal price being added as tip), and tax percent (the percentage of the meal price being added as tax) for a meal, find and print the meal's total cost. Round the result to the nearest integer.

Example

$meal_cost = 100$

$tip_percent = 15$

$tax_percent = 8$

A tip of 15% * 100 = 15, and the taxes are 8% * 100 = 8. Print the value 123 and return from the function.

Function Description

Complete the solve function in the editor below.

solve has the following parameters:

- int meal_cost: the cost of food before tip and tax
- int tip_percent: the tip percentage
- int tax_percent: the tax percentage

Returns The function returns nothing. Print the calculated value,

```
23
24
25      public static void solve(double meal_cost, double tip_percent,
    double tax_percent) {
26      tax_percent = tax_percent/100*meal_cost;
27      tip_percent = (tip_percent/100)*meal_cost;
28      double total = meal_cost+tip_percent+tax_percent;
29
30      System.out.println(Math.round(total));
31
32      }
33
34  }
35
```

Line: 51 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 18:02:43

Go to Dashboard Try a Random Challenge

Código Día 2:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'solve' function below.
     *
     * The function accepts following parameters:
     *  1. DOUBLE meal_cost
     *  2. INTEGER tip_percent
     *  3. INTEGER tax_percent
     */


    public static void solve(double meal_cost, double tip_percent, double tax_percent) {
    tax_percent = tax_percent/100*meal_cost;
    tip_percent = (tip_percent/100)*meal_cost;
    double total = meal_cost+tip_percent+tax_percent;

    System.out.println(Math.round(total));

    }

}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        double meal_cost = Double.parseDouble(bufferedReader.readLine().trim());

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

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

        Result.solve(meal_cost, tip_percent, tax_percent);

        bufferedReader.close();
    }
}
```

Objective

In this challenge, we learn about conditional statements. Check out the Tutorial tab for learning materials and an instructional video.

Task

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

- If n is odd, print `Weird`
- If n is even and in the inclusive range of 2 to 5, print `Not Weird`
- If n is even and in the inclusive range of 6 to 20, print `Weird`
- If n is even and greater than 20, print `Not Weird`

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

Input Format

A single line containing a positive integer, n.

Constraints

- $1 \leq n \leq 100$

Output Format

Print `Weird` if the number is weird; otherwise, print `Not Weird`.

Sample Input 0

```
3
```

```
18
19          int N = Integer.parseInt(bufferedReader.readLine().trim())
    ;
20
21          boolean par = N%2 == 0;
22
23          if(par==true && ((N>=2 && N<=5) || N>20)){
24              System.out.println("Not Weird");
25          } else{
26              System.out.println("Weird");
27          }
28
29          bufferedReader.close();
30      }
```

Line: 32 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 18:02:11

Go to Dashboard Try a Random Challenge

Código Día 3:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



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());

        boolean par = N%2 == 0;

        if(par==true && ((N>=2 && N<=5) || N>20)){
            System.out.println("Not Weird");
        } else{
            System.out.println("Weird");
        }

        bufferedReader.close();
    }
}
```

Objective

In this challenge, we're going to learn about the difference between a class and an instance; because this is an Object Oriented concept, it's only enabled in certain languages. Check out the Tutorial tab for learning materials and an instructional video!

Task

Write a Person class with an instance variable, age, and a constructor that takes an integer, $initialAge$, as a parameter. The constructor must assign $initialAge$ to age after confirming the argument passed as $initialAge$ is not negative; if a negative argument is passed as $initialAge$, the constructor should set age to 0 and print `Age is not valid, setting age to 0.`. In addition, you must write the following instance methods:

1. yearPasses() should increase the age instance variable by 1.
2. amIOld() should perform the following conditional actions:
   - If $age < 13$, print `You are young.`.
   - If $age \geq 13$ and $age < 18$, print `You are a teenager.`.
   - Otherwise, print `You are old.`.

```
            System.out.println("Age is not valid, setting age to 0.");
            age = 0;

        }
    }

    public void amIOld() {
        // Write code determining if this person's age is old and print
the correct statement:
        if(age<13){
            System.out.println("You are young.");
        } else if(age>=13 && age <18){
            System.out.println("You are a teenager.");
        } else{
            System.out.println("You are old.");
```

Line: 3 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 18:01:27

Go to Dashboard Try a Random Challenge

Código Día 4:

```
import java.io.*;
import java.util.*;

public class Person {
   private int age;

 public Person(int initialAge) {
   // Add some more code to run some checks on initialAge
        age = initialAge;
        if(age<0){
           System.out.println("Age is not valid, setting age to 0.");
           age = 0;

        }
   }

 public void amIOld() {
   // Write code determining if this person's age is old and print the correct statement:
       if(age<13){
          System.out.println("You are young.");
       } else if(age>=13 && age <18){
          System.out.println("You are a teenager.");
       } else{
          System.out.println("You are old.");
       }
 }

 public void yearPasses() {
   // Increment this person's age.
       age++;
 }

 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  int T = sc.nextInt();
  for (int i = 0; i < T; i++) {
   int age = sc.nextInt();
   Person p = new Person(age);
   p.amIOld();
   for (int j = 0; j < 3; j++) {
    p.yearPasses();
   }
   p.amIOld();
   System.out.println();
       }
  sc.close();
   }
}
```

Objective

In this challenge, we will use loops to do some math. Check out the Tutorial tab to learn more.

Task

Given an integer, n, print its first 10 multiples. Each multiple $n \times i$ (where $1 \leq i \leq 10$) should be printed on a new line in the form: `n x i = result`.

Example

$n = 3$

The printout should look like this:

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

Input Format

A single integer, n.

Constraints

```
15 public class Solution {
16     public static void main(String[] args) throws IOException {
17         BufferedReader bufferedReader = new BufferedReader(new
   InputStreamReader(System.in));
18
19         int n = Integer.parseInt(bufferedReader.readLine().trim())
   ;
20
21         for(int i=1;i<=10;i++){
22             System.out.printf("%s x %s = %s%n",n,i,n*i);
23         }
24
25         bufferedReader.close();
26     }
27 }
28
```

Line: 28 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 18:00:56

Go to Dashboard Try a Random Challenge

Código día 5:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



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.printf("%s x %s = %s%n",n,i,n*i);
        }

        bufferedReader.close();
    }
}
```

Objective

Today we will expand our knowledge of strings, combining it with what we have already learned about loops.

Check out the Tutorial tab for learning materials and an instructional video.

Task

Given a string, S, of length N that is indexed from 0 to $N-1$, print its even-indexed and odd-indexed characters as 2 space-separated strings on a single line (see the Sample below for more detail).

Note: 0 is considered to be an even index.

Example

$s = \text{adbecf}$

Print abc def

Input Format

The first line contains an integer, T (the number of test cases).

Each line i of the T subsequent lines contain a string, S.

Constraints

- $1 \leq T \leq 10$
- $2 \leq \text{length of } S \leq 10000$

Output Format

For each String S_j (where $0 \leq j \leq T-1$), print S_j's even-indexed characters, followed by a space, followed by S_j's odd-indexed characters.

Sample Input

```
2
Hacker
Rank
```

Sample Output

```
Hce akr
Rn ak
```

Explanation

Test Case 0: $S = \text{"Hacker"}$

$S[0] = \text{"H"}$

```
        scan.nextLine();
        for(int i=0;i<t;i++){
            String s = scan.nextLine();

            String frasePar = "";
            String fraseImpar = "";
            for(int j=0; j<s.length();j++){
                if(j%2==0){
                    frasePar+=s.charAt(j);
                }else{
                    fraseImpar+=s.charAt(j);
                }

            }

            System.out.println(frasePar+" "+fraseImpar);
        }
    }
}
```

Line: 25 Col: 57

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

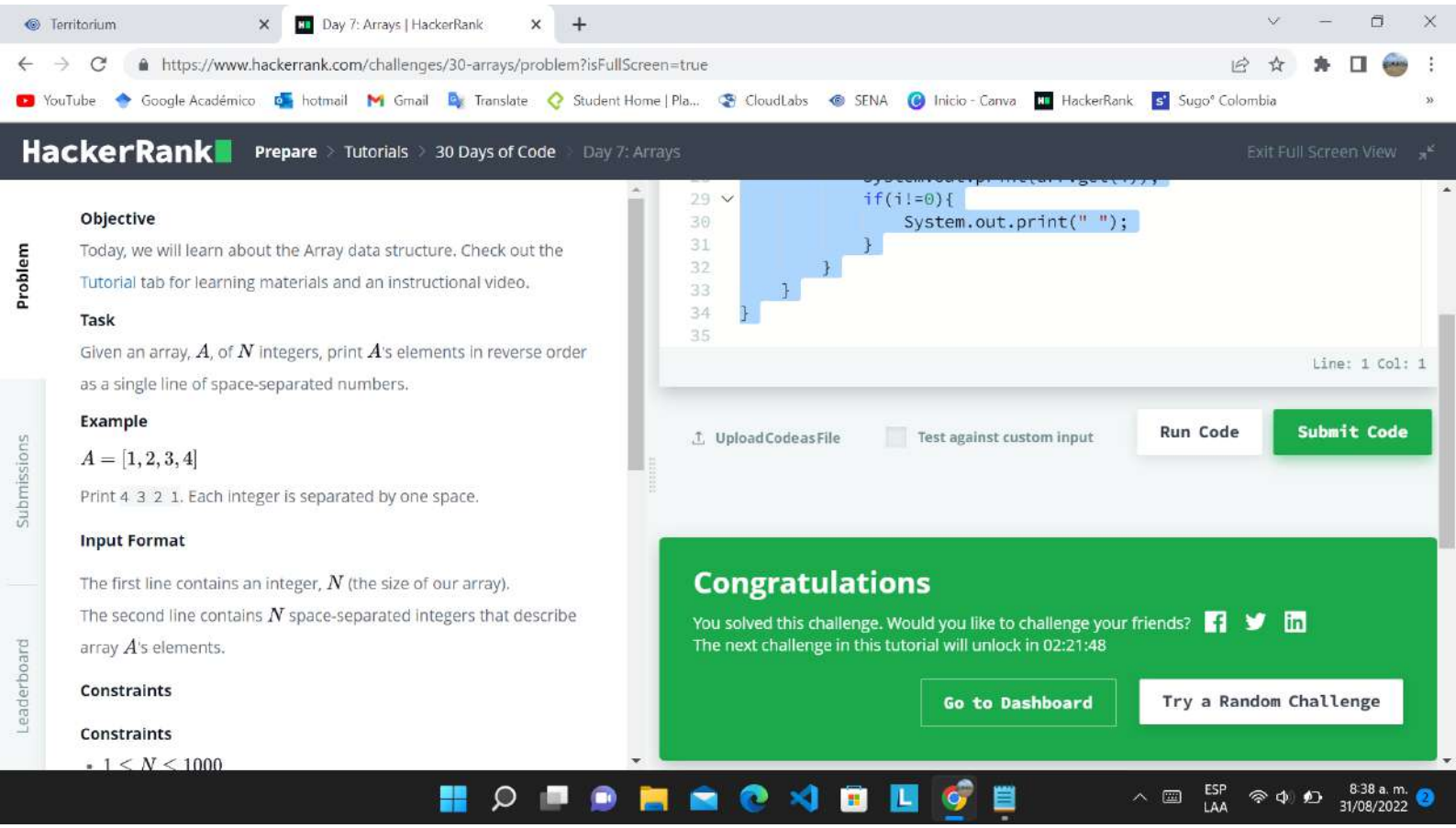
You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 23:24:43

Go to Dashboard Try a Random Challenge

Código Día 6:

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Soluti
on. */
        Scanner scan = new Scanner(System.in);
        int t = scan.nextInt();
        scan.nextLine();
        for(int i=0;i<t;i++){
            String s = scan.nextLine();

            String frasePar = "";
            String fraseImpar = "";
            for(int j=0; j<s.length();j++){
                if(j%2==0){
                    frasePar+=s.charAt(j);
                }else{
                    fraseImpar+=s.charAt(j);
                }

            }

            System.out.println(frasePar+" "+fraseImpar);
        }
    }
}
```


Territorium
Day 7: Arrays | HackerRank
https://www.hackerrank.com/challenges/30-arrays/problem?isFullScreen=true
YouTube
Google Académico
hotmail
Gmail
Translate
Student Home | Pla...
CloudLabs
SENA
Inicio - Canva
HackerRank
Sugo° Colombia
HackerRank
Prepare > Tutorials > 30 Days of Code > Day 7: Arrays
Exit Full Screen View
Problem
Submissions
Leaderboard
Objective
Today, we will learn about the Array data structure. Check out the Tutorial tab for learning materials and an instructional video.
Task
Given an array, A, of N integers, print A's elements in reverse order as a single line of space-separated numbers.
Example
A = [1,2,3,4]
Print 4 3 2 1. Each integer is separated by one space.
Input Format
The first line contains an integer, N (the size of our array).
The second line contains N space-separated integers that describe array A's elements.
Constraints
Constraints
1 ≤ N ≤ 1000
if(i!=0){
System.out.print(" ");
}
}
}
}
Line: 1 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 02:21:48
Go to Dashboard
Try a Random Challenge
ESP
LAA
8:38 a. m.
31/08/2022

Código día 7

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



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());

        List<Integer> arr = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))
            .map(Integer::parseInt)
            .collect(toList());

        bufferedReader.close();

        for(int i = n-1;i>=0;i--){
            System.out.print(arr.get(i));
            if(i!=0){
                System.out.print(" ");
            }
        }
    }
}
```

Tutorial

Sample Input

```
3
sam 99912222
tom 11122222
harry 12299933
sam
edward
harry
```

Sample Output

```
sam=99912222
Not found
harry=12299933
```

Explanation

We add the following $n = 3$ (Key,Value) pairs to our map so it looks like this:

$phoneBook = \{(sam, 99912222), (tom, 11122222), (harry, 12299933)\}$

We then process each query and print `key=value` if the queried key is found in the map; otherwise, we print `Not found`.

Query 0: `sam`

Sam is one of the keys in our dictionary, so we print `sam=99912222`.

Query 1: `edward`

Edward is not one of the keys in our dictionary, so we print `Not found`.

Query 2: `harry`

Harry is one of the keys in our dictionary, so we print `harry=12299933`.

```
13          // Write code here
14          phoneBook.put(name,phone);
15      }
16      while(in.hasNext()){
17          String s = in.next();
18          // Write code here
19          if(phoneBook.containsKey(s)==true){
20              System.out.println(s+"="+phoneBook.get(s));
21          }else{
22            System.out.println("Not found");
23          }
24      }
25      in.close();
26  }
27  }
```

Line: 27 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 6 challenges away from the 3rd star for your 30 days of code badge. 25% 9/15

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 15:47:20

Código Día 8:

```
//Complete this code or write your own from scratch
import java.util.*;
import java.io.*;

class Solution{
    public static void main(String []argh){
        Scanner in = new Scanner(System.in);
        HashMap phoneBook = new HashMap<>();
        int n = in.nextInt();
        for(int i = 0; i < n; i++){
            String name = in.next();
            int phone = in.nextInt();
            // Write code here
            phoneBook.put(name,phone);
        }
        while(in.hasNext()){
            String s = in.next();
            // Write code here
            if(phoneBook.containsKey(s)==true){
                System.out.println(s+"="+phoneBook.get(s));
            }else{
              System.out.println("Not found");
            }
        }
        in.close();
    }
}
```

Objective

Today, we are learning about an algorithmic concept called recursion. Check out the Tutorial tab for learning materials and an instructional video.

Recursive Method for Calculating Factorial

$$factorial(N) = \begin{cases} 1 & N \leq 1 \\ N \times factorial(N-1) & otherwise \end{cases}$$

Function Description

Complete the factorial function in the editor below. Be sure to use recursion.

factorial has the following paramter:

- int n: an integer

Returns

- int: the factorial of n

Note: If you fail to use recursion or fail to name your recursive function factorial or Factorial, you will get a score of 0.

Input Format

A single integer, n (the argument to pass to factorial).

Constraints

- $2 \leq n \leq 12$
- Your submission must contain a recursive function named factorial.

Sample Input

```
3
```

```
public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter
(System.getenv("OUTPUT_PATH")));

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

        int result = Result.factorial(n);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedReader.close();
```

Line: 23 Col: 28

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 5 challenges away from the 3rd star for your 30 days of code badge. 38% 10/15

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 13:41:59

Go to Dashboard Try a Random Challenge

Código día 9:

```
public static int factorial(int n) {
    int factoResultado = 1;
    for(int i = n;i>0;i--){
        factoResultado*=i;
    }
    return factoResultado;
    }
```

Print a single base-10 integer that denotes the maximum number of consecutive 1's in the binary representation of n.

Sample Input 1

```
5
```

Sample Output 1

```
1
```

Sample Input 2

```
13
```

Sample Output 2

```
2
```

Explanation

Sample Case 1:

The binary representation of 5_{10} is 101_2, so the maximum number of consecutive 1's is 1.

Sample Case 2:

The binary representation of 13_{10} is 1101_2, so the maximum number of consecutive 1's is 2.

```
16    public static void main(String[] args) throws IOException {
17        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
18
19        int n = Integer.parseInt(bufferedReader.readLine().trim());
20
21        bufferedReader.close();
22
23        String binary = Integer.toBinaryString(n);
24        int contador = 0;
25        int secuencia = 0;
26        for(int i=0;i<binary.length();i++){
27            if(binary.charAt(i) == '1'){
```

Line: 8 Col: 26

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 4 challenges away from the 3rd star for your 30 days of code badge. 50% 11/15

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 01:05:03

Go to Dashboard Try a Random Challenge

Código día 10:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



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());

        bufferedReader.close();

        String binary = Integer.toBinaryString(n);
        int contador = 0;
        int secuencia = 0;
        for(int i=0;i<binary.length();i++){
            if(binary.charAt(i) == '1'){
                contador++;
                if(secuencia<contador){
                    secuencia = contador;
                }
            }else{
                contador=0;
            }
        }
        System.out.println(secuencia);
    }
}
```

Calculate the hourglass sum for every hourglass in A, then print the maximum hourglass sum.

Example

In the array shown above, the maximum hourglass sum is 7 for the hourglass in the top left corner.

Input Format

There are 6 lines of input, where each line contains 6 space-separated integers that describe the 2D Array A.

Constraints

- $-9 \leq A[i][j] \leq 9$
- $0 \leq i, j \leq 5$

Output Format

Print the maximum hourglass sum in A.

Sample Input

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
0 0 0 2 0 0
0 0 1 2 4 0
```

Sample Output

```
19
```

Explanation

A contains the following hourglasses:

```
1 1 1   1 1 0   1 0 0   0 0 0
  1       0       0       0
1 1 1   1 1 0   1 0 0   0 0 0
```

```
                    contador++;

                }
                fila++;
                contador=0;

            }
            move++;
            if(move%4==0){
                move=0;
                filaCentro++;
            }

            if(sumatoria>sumatoriaMaxima){
                sumatoriaMaxima=sumatoria;
            }
            sumatoria=0;

        }
        System.out.println(sumatoriaMaxima);
    }
}
```

Line: 79 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 3 challenges away from the 3rd star for your 30 days of code badge.

63% 12/15

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 19:10:17

Go to Dashboard Try a Random Challenge

Código día 11:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        List<List<Integer>> arr = new ArrayList<>();

        IntStream.range(0, 6).forEach(i -> {
            try {
                arr.add(
                    Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))
                        .map(Integer::parseInt)
                        .collect(toList())
                );
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        });

        bufferedReader.close();
        int move = 0;
        int filaCentro=0;
        int sumatoriaMaxima=-100;
        int sumatoria=0;

        for(int z=0;z<16;z++){
            int contador=0;
            int fila = 0;

            for(int i=0;i<arr.size();i++){
                for(int j=0;j<arr.get(i).size();j++){

                    if(fila==0 || fila==2){
                        sumatoria+=arr.get(i+filaCentro).get(j+move);
                        if(contador==2){
                            break;
                        }
                    } else if(fila==1){
                        if(contador==1){
                            sumatoria+=arr.get(i+filaCentro).get(j+move);
```

```
                    break;
                }
            }
            contador++;

        }
        fila++;
        contador=0;

    }
    move++;
    if(move%4==0){
        move=0;
        filaCentro++;
    }

    if(sumatoria>sumatoriaMaxima){
        sumatoriaMaxima=sumatoria;
    }
    sumatoria=0;

  }
  System.out.println(sumatoriaMaxima);
 }
}
```

3. An integer, $idNumber$.
4. An integer array (or vector) of test scores, $scores$.

- A char calculate() method that calculates a Student object's average and returns the grade character representative of their calculated average:

Grading Scale

Letter	Average (a)
O	$90 \le a \le 100$
E	$80 \le a < 90$
A	$70 \le a < 80$
P	$55 \le a < 70$
D	$40 \le a < 55$
T	$a < 40$

Input Format

The locked stub code in the editor reads the input and calls the Student class constructor with the necessary arguments. It also calls the calculate method which takes no arguments.

The first line contains $firstName$, $lastName$, and $idNumber$, separated by a space. The second line contains the number of test scores. The third line of space-separated integers describes $scores$.

Constraints

- $1 \le$ length of firstName, length of lastName ≤ 10

```
50          for(int i:testScores){
51              average+=i;
52          }
53          average/=testScores.length;
54
55          if(average>=90 && average<=100){
56              letter='O';
57          }else if(average>80 && average<90){
58              letter='E';
59          }else if(average>=70 && average<80){
60              letter='A';
61          }else if(average>=55 && average<70){
62              letter='P';
63          }else if(average>=40 && average<55){
64              letter='D';
65          }else{
66              letter='T';
67          }
68          return letter;
69      }
70  }
```

Line: 15 Col: 22

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 22:50:11

Go to Dashboard Try a Random Challenge

Código día 12:

```
import java.util.*;

class Person {
 protected String firstName;
 protected String lastName;
 protected int idNumber;

 // Constructor
 Person(String firstName, String lastName, int identification){
  this.firstName = firstName;
  this.lastName = lastName;
  this.idNumber = identification;
 }

 // Print person data
 public void printPerson(){
   System.out.println(
    "Name: " + lastName + ", " + firstName
  +  "\nID: " + idNumber);
 }

}

class Student extends Person{
 private int[] testScores;

    /*
    *   Class Constructor
    *
    *   @param firstName - A string denoting the Person's first name.
    *   @param lastName - A string denoting the Person's last name.
    *   @param id - An integer denoting the Person's ID number.
    *   @param scores - An array of integers denoting the Person's test scores.
    */
    // Write your constructor here
public Student(String firstName, String lastName, int idNumber, int[] score){
    super(firstName,lastName,idNumber);
    this.testScores = score;
    }
    /*
    *   Method Name: calculate
    *   @return A character denoting the grade.
    */
    // Write your method here

    public char calculate(){
        int average=0;
        char letter;

        for(int i:testScores){
            average+=i;
        }
```

```
        average/=testScores.length;

        if(average>=90 && average<=100){
           letter='O';
        }else if(average>80 && average<90){
           letter='E';
        }else if(average>=70 && average<80){
           letter='A';
        }else if(average>=55 && average<70){
           letter='P';
        }else if(average>=40 && average<55){
           letter='D';
        }else{
           letter='T';
        }
        return letter;
    }
}
class Solution {
 public static void main(String[] args) {
  Scanner scan = new Scanner(System.in);
  String firstName = scan.next();
  String lastName = scan.next();
  int id = scan.nextInt();
  int numScores = scan.nextInt();
  int[] testScores = new int[numScores];
  for(int i = 0; i < numScores; i++){
   testScores[i] = scan.nextInt();
  }
  scan.close();

  Student s = new Student(firstName, lastName, id, testScores);
  s.printPerson();
  System.out.println("Grade: " + s.calculate() );
 }
}
```

Output Format

The *void display()* method should print and label the respective *title*, *author*, and *price* of the MyBook object's instance (with each value on its own line) like so:

```
Title: $title
Author: $author
Price: $price
```

Note: The $ is prepended to variable names to indicate they are placeholders for variables.

Sample Input

The following input from stdin is handled by the locked stub code in your editor:

```
The Alchemist
Paulo Coelho
248
```

Sample Output

The following output is printed by your display() method:

```
Title: The Alchemist
Author: Paulo Coelho
Price: 248
```

```
17    int price;
18
19    public MyBook(String title, String author, int price){
20        super(title, author);
21        this.price=price;
22    }
23
24    public void display(){
25        System.out.println("Title: "+title);
26        System.out.println("Author: "+author);
27        System.out.println("Price: "+price);
28
29    }
```

Line: 28 Col: 9

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 1 challenge away from the 3rd star for your 30 days of code badge. 88% 14/15

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 22:57:41

Go to Dashboard Try a Random Challenge

Código día 13:

```
import java.util.*;

abstract class Book {
    String title;
    String author;

    Book(String title, String author) {
        this.title = title;
        this.author = author;
    }

    abstract void display();
}

// Declare your class here. Do not use the 'public' access modifier.
class MyBook extends Book{
    int price;

    public MyBook(String title, String author, int price){
        super(title, author);
        this.price=price;
    }

    public void display(){
        System.out.println("Title: "+title);
        System.out.println("Author: "+author);
        System.out.println("Price: "+price);

    }

}

public class Solution {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String title = scanner.nextLine();
        String author = scanner.nextLine();
        int price = scanner.nextInt();
        scanner.close();

        Book book = new MyBook(title, author, price);
        book.display();
    }
}
```

- A computeDifference method that finds the maximum absolute difference between any 2 numbers in $\_\_elements$ and stores it in the $maximumDifference$ instance variable.

Input Format

You are not responsible for reading any input from stdin. The locked Solution class in the editor reads in 2 lines of input. The first line contains N, the size of the elements array. The second line has N space-separated integers that describe the $\_\_elements$ array.

Constraints

- $1 \leq N \leq 10$
- $1 \leq \_\_elements[i] \leq 100$, where $0 \leq i \leq N-1$

Output Format

You are not responsible for printing any output; the Solution class will print the value of the $maximumDifference$ instance variable.

Sample Input

```
STDIN   Function
-----   --------
3       __elements[] size N = 3
1 2 5   __elements = [1, 2, 5]
```

Sample Output

```
4
```

Explanation

The scope of the $\_\_elements$ array and $maximumDifference$ integer is the entire class instance. The class constructor saves the argument passed to the constructor as the $\_\_elements$ instance variable (where the computeDifference method can access it).

To find the maximum difference, computeDifference checks each element in the array and finds the maximum difference between any 2 elements: $|1-2| = 1$

$|1-5| = 4$

$|2-5| = 3$

```
16
17  public int computeDifference(){
18      int max = 0;
19      int min = 0;
20
21      for(int i:elements){
22          if(i>max){
23              if(min==0){
24                  min=i;
25              }
26              max = i;
27          }else{
28              if(i<min){
29                  min=i;
30              }
31          }
32
33      }
34
35      maximumDifference = max-min;
36      return maximumDifference;
```

Line: 13 Col: 39

Upload Code as File | Test against custom input | Run Code | Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 7 challenges away from the 4th star for your 30 days of code badge.

0% 15/22

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 13:02:39

Go to Dashboard | Try a Random Challenge

Código día 14:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;


class Difference {
  private int[] elements;
  public int maximumDifference;

 // Add your code here
   public Difference(int[] elements){
      this.elements = elements;
   }

   public int computeDifference(){
      int max = 0;
      int min = 0;

      for(int i:elements){
        if(i>max){
           if(min==0){
              min=i;
           }
           max = i;
        }else{
           if(i<min){
              min=i;
           }
        }

      }

      maximumDifference = max-min;
      return maximumDifference;
   }

} // End of Difference class

public class Solution {

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] a = new int[n];
      for (int i = 0; i < n; i++) {
         a[i] = sc.nextInt();
      }
      sc.close();
```

```
        Difference difference = new Difference(a);

        difference.computeDifference();

        System.out.print(difference.maximumDifference);
    }
}
```

```
STDIN   Function
-----   --------
4       T = 4
2       first data = 2
3
4
1       fourth data = 1
```

Sample Output

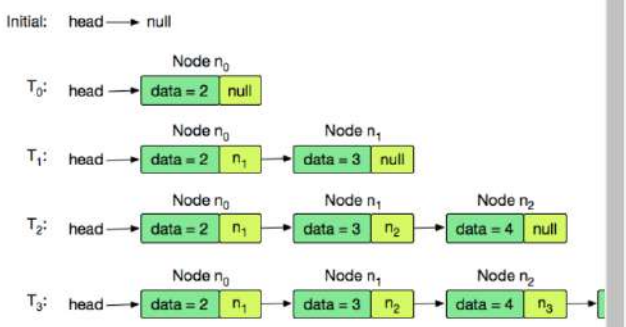
```
2 3 4 1
```

Explanation

$T = 4$, so your method will insert 4 nodes into an initially empty list.

First the code returns a new node that contains the data value 2 as the $head$ of the list.

Then create and insert nodes 3, 4, and 1 at the tail of the list.




```java
33
34      public static void display(Node head) {
35          Node start = head;
36          while(start != null) {
37              System.out.print(start.data + " ");
38              start = start.next;
39          }
40      }
41
42      public static void main(String args[]) {
43          Scanner sc = new Scanner(System.in);
44          Node head = null;
45          int N = sc.nextInt();
46
47          while(N-- > 0) {
48              int ele = sc.nextInt();
49              head = insert(head,ele);
50          }
51          display(head);
52          sc.close();
53      }
54  }
```

Line: 45 Col: 30

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 5 challenges away from the 4th star for your 30 days of code badge.

29% 17/22

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Código día 15:

```
import java.io.*;
import java.util.*;

class Node {
 int data;
 Node next;
 Node(int d) {
       data = d;
       next = null;
   }
}

class Solution {

   public static  Node insert(Node head,int data) {

       if(head == null){
          return new Node(data);
       }
       Node prueba = head;


       while(prueba.next != null){
          prueba = prueba.next;
       }

       prueba.next = new Node(data);

       return head;


   }

 public static void display(Node head) {
       Node start = head;
       while(start != null) {
          System.out.print(start.data + " ");
          start = start.next;
       }
   }

   public static void main(String args[]) {
       Scanner sc = new Scanner(System.in);
       Node head = null;
       int N = sc.nextInt();

       while(N-- > 0) {
          int ele = sc.nextInt();
          head = insert(head,ele);
       }
       display(head);
```

```
        sc.close();
    }
}
```

A single string, S.

Constraints

- $1 \leq |S| \leq 6$, where $|S|$ is the length of string S.
- S is composed of either lowercase letters $(a - z)$ or decimal digits $(0 - 9)$.

Output Format

Print the parsed integer value of S, or `Bad String` if S cannot be converted to an integer.

Sample Input 0

```
3
```

Sample Output 0

```
3
```

Sample Input 1

```
za
```

Sample Output 1

```
Bad String
```

Explanation

Sample Case 0 contains an integer, so it should not raise an exception when

```
13
14
15  public class Solution {
16      public static void main(String[] args) throws IOException {
17          BufferedReader bufferedReader = new BufferedReader(new
    InputStreamReader(System.in));
18
19          String S = bufferedReader.readLine();
20
21          bufferedReader.close();
22          try{
23              System.out.println(Integer.parseInt(S));
24          }catch(Exception e){
25              System.out.println("Bad String");
26          }
27      }
```

Line: 22 Col: 13

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 6 challenges away from the 4th star for your 30 days of code badge. 14% 16/22

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Código día 16:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        String S = bufferedReader.readLine();

        bufferedReader.close();
        try{
            System.out.println(Integer.parseInt(S));
        }catch(Exception e){
            System.out.println("Bad String");
        }
    }
}
```

Objective

Yesterday's challenge taught you to manage exceptional situations by using try and catch blocks. In today's challenge, you will practice throwing and propagating an exception. Check out the Tutorial tab for learning materials and an instructional video.

Task

Write a Calculator class with a single method: int power(int,int). The power method takes two integers, n and p, as parameters and returns the integer result of n^p. If either n or p is negative, then the method must throw an exception with the message: `n and p should be non-negative`.

Note: Do not use an access modifier (e.g.: public) in the declaration for your Calculator class.

Input Format

Input from stdin is handled for you by the locked stub code in your editor. The first line contains an integer, T, the number of test cases. Each of the T subsequent lines describes a test case in 2 space-separated integers that denote n and p, respectively.

Constraints

- No Test Case will result in overflow for correctly written code.

Output Format

Output to stdout is handled for you by the locked stub code in your editor. There are T lines of output, where each line contains the result of n^p as calculated by your Calculator class' power method.

Sample Input

```
4
3 5
2 4
-1 -2
-1 3
```

```
7     public int power(int n, int p) throws Exception{
8         int potencia=1;
9
10        try{
11            if(n<0 || p<0){
12                throw new Exception();
13
14            }else{
15                for(int i=p;i>0;i--){
16                    potencia*=n;
17                }
18                return potencia;
19
20
21            }
22
23        }catch(Exception e){
24            throw new Exception("n and p should be non-negative");
25
26        }
27    }
28 }
29
30 class Solution{
31
32     public static void main(String[] args) {
33
```

Line: 48 Col: 10

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Código día 17:

```
import java.util.*;
import java.io.*;

//Write your code here
class Calculator{

   public int power(int n, int p) throws Exception{
      int potencia=1;

      try{
         if(n<0 || p<0){
            throw new Exception();

         }else{
            for(int i=p;i>0;i--){
               potencia*=n;
            }
            return potencia;


         }

      }catch(Exception e){
        throw new Exception("n and p should be non-negative");

      }
   }
}

class Solution{

   public static void main(String[] args) {

      Scanner in = new Scanner(System.in);
      int t = in.nextInt();
      while (t-- > 0) {

         int n = in.nextInt();
         int p = in.nextInt();
         Calculator myCalculator = new Calculator();
         try {
            int ans = myCalculator.power(n, p);
            System.out.println(ans);
         }
         catch (Exception e) {
            System.out.println(e.getMessage());
         }
      }
      in.close();
   }
}
```

Welcome to Day 18! Today we're learning about Stacks and Queues. Check out the Tutorial tab for learning materials and an instructional video!

A palindrome is a word, phrase, number, or other sequence of characters which reads the same backwards and forwards. Can you determine if a given string, s, is a palindrome?

To solve this challenge, we must first take each character in s, enqueue it in a queue, and also push that same character onto a stack. Once that's done, we must dequeue the first character from the queue and pop the top character off the stack, then compare the two characters to see if they are the same; as long as the characters match, we continue dequeueing, popping, and comparing each character until our containers are empty (a non-match means s isn't a palindrome).

Write the following declarations and implementations:

1. Two instance variables: one for your $stack$, and one for your $queue$.
2. A void pushCharacter(char ch) method that pushes a character onto a stack.
3. A void enqueueCharacter(char ch) method that enqueues a character in the $queue$ instance variable.
4. A char popCharacter() method that pops and returns the character at the top of the $stack$ instance variable.
5. A char dequeueCharacter() method that dequeues and returns the first character in the $queue$ instance variable.

Input Format

You do not need to read anything from stdin. The locked stub code in your editor reads a single line containing string s. It then calls the methods specified above to pass each character to your instance variables.

Constraints

- s is composed of lowercase English letters.

Output Format

You are not responsible for printing any output to stdout.
If your code is correctly written and s is a palindrome, the locked stub code will print

```
14      }
15
16      public char popCharacter(){
17          return stack.pop();
18      }
19
20      public char dequeueCharacter(){
21          return queue.poll();
22      }
23
24
25
26  >   public static void main(String[] args) {···
```

Line: 7 Col: 5

Upload Code as File | Test against custom input | Run Code | Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 3 challenges away from the 4th star for your 30 days of code badge.

57% 19/22

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 16:43:01

Go to Dashboard | Try a Random Challenge

Código día 18:

```
import java.io.*;
import java.util.*;

public class Solution {
    Stack<Character> stack = new Stack<>();
    LinkedList<Character> queue = new LinkedList<>();

    public void pushCharacter(char ch){
        stack.add(ch);
    }

    public void enqueueCharacter(char ch){
        queue.add(ch);
    }

    public char popCharacter(){
        return stack.pop();
    }

    public char dequeueCharacter(){
        return queue.poll();
    }



    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String input = scan.nextLine();
        scan.close();

        // Convert input String to an array of characters:
        char[] s = input.toCharArray();

        // Create a Solution object:
        Solution p = new Solution();

        // Enqueue/Push all chars to their respective data structures:
        for (char c : s) {
            p.pushCharacter(c);
            p.enqueueCharacter(c);
        }

        // Pop/Dequeue the chars at the head of both data structures and compare them:
        boolean isPalindrome = true;
        for (int i = 0; i < s.length/2; i++) {
            if (p.popCharacter() != p.dequeueCharacter()) {
                isPalindrome = false;
                break;
            }
        }

        //Finally, print whether string s is palindrome or not.
```

```
        System.out.println( "The word, " + input + ", is "
                        + ( (!isPalindrome) ? "not a palindrome." : "a palindrome." ) );
    }
}
```

Objective

Today, we're learning about Interfaces. Check out the Tutorial tab for learning materials and an instructional video!

Task

The `AdvancedArithmetic` interface and the method declaration for the abstract `divisorSum(n)` method are provided for you in the editor below.

Complete the implementation of `Calculator` class, which implements the `AdvancedArithmetic` interface. The implementation for the `divisorSum(n)` method must return the sum of all divisors of n.

Example

$n = 25$

The divisors of 25 are $1, 5, 25$. Their sum is 31.

$n = 20$

The divisors of 20 are $1, 2, 4, 5, 10, 20$ and their sum is 42.

Input Format

A single line with an integer, n.

Constraints

- $1 \leq n \leq 1000$

Output Format

You are not responsible for printing anything to stdout. The locked template code in the editor below will call your code and print the necessary output.

Sample Input

```
6
```

Sample Output

```
class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        scan.close();

        AdvancedArithmetic myCalculator = new Calculator();
        int sum = myCalculator.divisorSum(n);
        System.out.println("I implemented: " + myCalculator.getClass().getInterfaces()[0].getName() );
        System.out.println(sum);
    }
}
```

Line: 1 Col: 18

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 2 challenges away from the 4th star for your 30 days of code badge.

71% 20/22

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 21:13:28

Go to Dashboard Try a Random Challenge

Código día 19:

```
import java.io.*;
import java.util.*;

interface AdvancedArithmetic{
  int divisorSum(int n);
}
class Calculator implements AdvancedArithmetic {
    public int divisorSum(int n) {
        int suma=0;
        for(int i = n;i>0;i--){
            if(n%i==0){
                suma+=i;
            }
        }

        return suma;
    }
}

class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        scan.close();

       AdvancedArithmetic myCalculator = new Calculator();
        int sum = myCalculator.divisorSum(n);
        System.out.println("I implemented: " + myCalculator.getClass().getInterfaces()[0].getName() );
        System.out.println(sum);
    }
}
```

3. `Last Element: lastElement`

where $lastElement$ is the last element in the sorted array.

Sample Input 0

```
3
1 2 3
```

Sample Output 0

```
Array is sorted in 0 swaps.
First Element: 1
Last Element: 3
```

Explanation 0

The array is already sorted, so 0 swaps take place and we print the necessary 3 lines of output shown above.

Sample Input 1

```
3
3 2 1
```

Sample Output 1

```
Array is sorted in 3 swaps.
First Element: 1
Last Element: 3
```

Explanation 1

The array $a = [3, 2, 1]$ is not sorted, so we perform the following 3 swaps. Each line

```
27      for(int i=0;i<a.size()-1;i++){
28          for(int j=0;j<a.size()-1;j++){
29              int aux=a.get(j);
30
31
32              if(a.get(j)>a.get(j+1)){
33                  a.set(j, a.get(j+1));
34                  a.set(j+1, aux);
35                  numSwaps++;
36              }
37
38
39          }
40      }
41
42      System.out.println("Array is sorted in "+numSwaps+" swaps.");
43      System.out.println("First Element: "+a.get(0));
44      System.out.println("Last Element: "+a.get(a.size()-1));
45
```

Line: 35 Col: 32

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 1 challenge away from the 4th star for your 30 days of code badge.

86% 21/22

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 17:28:56

Go to Dashboard Try a Random Challenge

Código día 20:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



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());

        List<Integer> a = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))
            .map(Integer::parseInt)
            .collect(toList());

        // Write your code here
        int numSwaps=0;
        for(int i=0;i<a.size()-1;i++){
            for(int j=0;j<a.size()-1;j++){
                int aux=a.get(j);


                if(a.get(j)>a.get(j+1)){
                    a.set(j, a.get(j+1));
                    a.set(j+1, aux);
                    numSwaps++;
                }


            }
        }

        System.out.println("Array is sorted in "+numSwaps+" swaps.");
        System.out.println("First Element: "+a.get(0));
        System.out.println("Last Element: "+a.get(a.size()-1));

        bufferedReader.close();
    }
}
```

Objective

Today we're discussing Generics; be aware that not all languages support this construct, so fewer languages are enabled for this challenge. Check out the Tutorial tab for learning materials and an instructional video!

Task

Write a single generic function named printArray; this function must take an array of generic elements as a parameter (the exception to this is C++, which takes a vector). The locked Solution class in your editor tests your function.

Note: You must use generics to solve this challenge. Do not write overloaded functions.

Input Format

The locked Solution class in your editor will pass different types of arrays to your printArray function.

Constraints

- You must have exactly 1 function named printArray.

Output Format

Your printArray function should print each element of its generic array parameter on a new line.

```
    }


}

public class Generics {

    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        Integer[] intArray = new Integer[n];
        for (int i = 0; i < n; i++) {
            intArray[i] = scanner.nextInt();
        }

        n = scanner.nextInt();
        String[] stringArray = new String[n];
        for (int i = 0; i < n; i++) {
            stringArray[i] = scanner.next();
        }

        Printer<Integer> intPrinter = new Printer<Integer>();
```

Line: 11 Col: 6

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 8 challenges away from the gold level for your 30 days of code badge.

0% 22/30

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Código día 21:

```
import java.util.*;

class Printer <T> {

   private T[] data;

   public void printArray(T[] data){
      this.data=data;
      for(T i:data){
      System.out.println(i);
   }
   }


}

public class Generics {

   public static void main(String args[]){
      Scanner scanner = new Scanner(System.in);
      int n = scanner.nextInt();
      Integer[] intArray = new Integer[n];
      for (int i = 0; i < n; i++) {
         intArray[i] = scanner.nextInt();
      }

      n = scanner.nextInt();
      String[] stringArray = new String[n];
      for (int i = 0; i < n; i++) {
         stringArray[i] = scanner.next();
      }

      Printer<Integer> intPrinter = new Printer<Integer>();
      Printer<String> stringPrinter = new Printer<String>();
      intPrinter.printArray( intArray  );
      stringPrinter.printArray( stringArray );
      if(Printer.class.getDeclaredMethods().length > 1){
         System.out.println("The Printer class should only have 1 method named printArray.");
      }
   }
}
```

Input Format

The locked stub code in your editor reads the following inputs and assembles them into a binary search tree:
The first line contains an integer, n, denoting the number of nodes in the tree.
Each of the n subsequent lines contains an integer, $data$, denoting the value of an element that must be added to the BST.

Output Format

The locked stub code in your editor will print the integer returned by your getHeight function denoting the height of the BST.

Sample Input

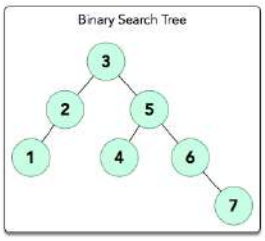
```
7
3
5
2
1
4
6
7
```

Sample Output

```
3
```

Explanation

The input forms the following BST:




```
                if(contadorMetodo>contador){
                    contador= getLeft(ladoDerecho, contadorMetodo);
                }
            }
        }

        if(contadorMetodo>contador){
            return contadorMetodo;
        }else{
            return contador;
        }
    }

    public static int getLeft(Node ladoIzquierdo, int contador){
        int contadorMetodo=contador;

        while(ladoIzquierdo.left!=null){
            contadorMetodo++;
            ladoIzquierdo=ladoIzquierdo.left;
            if(ladoIzquierdo.right!=null){
                if(contadorMetodo>contador){
```

Line: 64 Col: 10

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!
You are now 7 challenges away from the gold level for your 30 days of code badge.

13% 23/30

Congratulations
You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0
Test case 1
Test case 2

Compiler Message
Success

Input (stdin) Download
7

Código día 22:

```
import java.util.*;
import java.io.*;
class Node{
    Node left,right;
    int data;
    Node(int data){
        this.data=data;
        left=right=null;
    }
}
class Solution{


 public static int getHeight(Node root){
        int contadorLeft=0;
        int contadorRight=0;
        contadorRight=getRight(root, contadorRight);
        contadorLeft=getLeft(root, contadorLeft);

        if(contadorLeft>contadorRight){
            return contadorLeft;
        }else{
            return contadorRight;
        }

    }

    public static int getRight(Node ladoDerecho, int contador){
        int contadorMetodo=contador;

        while(ladoDerecho.right!=null){
            contadorMetodo++;
            ladoDerecho=ladoDerecho.right;
            if(ladoDerecho.left!=null){
                if(contadorMetodo>contador){
                    contador= getLeft(ladoDerecho, contadorMetodo);
                }
            }
        }

        if(contadorMetodo>contador){
            return contadorMetodo;
        }else{
            return contador;
        }
    }

    public static int getLeft(Node ladoIzquierdo, int contador){
        int contadorMetodo=contador;

        while(ladoIzquierdo.left!=null){
            contadorMetodo++;
```

```
            ladoIzquierdo=ladoIzquierdo.left;
            if(ladoIzquierdo.right!=null){
                if(contadorMetodo>contador){
                    contador= getRight(ladoIzquierdo, contadorMetodo);
                }
            }
        }

        if(contadorMetodo>contador){
            return contadorMetodo;
        }else{
            return contador;
        }


    }

    public static Node insert(Node root,int data){
        if(root==null){
            return new Node(data);
        }
        else{
            Node cur;
            if(data<=root.data){
                cur=insert(root.left,data);
                root.left=cur;
            }
            else{
                cur=insert(root.right,data);
                root.right=cur;
            }
            return root;
        }
    }
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int T=sc.nextInt();
        Node root=null;
        while(T-->0){
            int data=sc.nextInt();
            root=insert(root,data);
        }
        int height=getHeight(root);
        System.out.println(height);
    }
}
```

nodes from left to right, top to bottom. You are given a pointer, ***root***, pointing to the root of a binary search tree. Complete the levelOrder function provided in your editor so that it prints the level-order traversal of the binary search tree.

Hint: You'll find a queue helpful in completing this challenge.

Function Description

Complete the levelOrder function in the editor below.

levelOrder has the following parameter:

- Node pointer root: a reference to the root of the tree

Prints

- Print node.data items as space-separated line of integers. No return value is expected.

Input Format

The locked stub code in your editor reads the following inputs and assembles them into a BST:

The first line contains an integer, T (the number of test cases).

The T subsequent lines each contain an integer, $data$, denoting the value of an element that must be added to the BST.

Constraints

$1 \leq N \leq 20$

$1 \leq node.data[i] \leq 100$

Output Format

Print the $data$ value of each node in the tree's level-order traversal as a single line of N space-separated integers.

Sample Input

```
6
3
5
4
```

```
14  static void levelOrder(Node root){
15      cola.add(root);
16
17      for(int i=0;i<cola.size();i++){
18          if(cola.get(i).left!=null){
19              cola.add(cola.get(i).left);
20          }
21
22          if(cola.get(i).right!=null){
23              cola.add(cola.get(i).right);
24          }
25      }
26
27      for(Node i:cola){
28          System.out.print(i.data+" ");
29      }
30  }
31
32
33  public static Node insert(Node root,int data){
```

Line: 6 Col: 20

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 1 challenge away from the gold level for your 30 days of code badge.

88% 29/30

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 00:41:26

Go to Dashboard Try a Random Challenge

Código día 23:

```
import java.util.*;
import java.io.*;
class Node{
    Node left,right;
    int data;
    Node(int data){
        this.data=data;
        left=right=null;
    }
}
class Solution{
static LinkedList<Node> cola= new LinkedList<Node>();

static void levelOrder(Node root){
    cola.add(root);

    for(int i=0;i<cola.size();i++){
        if(cola.get(i).left!=null){
            cola.add(cola.get(i).left);
        }

        if(cola.get(i).right!=null){
            cola.add(cola.get(i).right);
        }
    }

    for(Node i:cola){
        System.out.print(i.data+" ");
    }
}


public static Node insert(Node root,int data){
        if(root==null){
            return new Node(data);
        }
        else{
            Node cur;
            if(data<=root.data){
                cur=insert(root.left,data);
                root.left=cur;
            }
            else{
                cur=insert(root.right,data);
                root.right=cur;
            }
            return root;
        }
    }
    public static void main(String args[]){
            Scanner sc=new Scanner(System.in);
            int T=sc.nextInt();
```

```
        Node root=null;
        while(T-->0){
            int data=sc.nextInt();
            root=insert(root,data);
        }
        levelOrder(root);
    }
}
```

instance pointer, $next$, pointing to another node (i.e.: the next node in a list).

A removeDuplicates function is declared in your editor, which takes a pointer to the $head$ node of a linked list as a parameter. Complete removeDuplicates so that it deletes any duplicate nodes from the list and returns the head of the updated list.

Note: The $head$ pointer may be null, indicating that the list is empty. Be sure to reset your $next$ pointer when performing deletions to avoid breaking the list.

Input Format

You do not need to read any input from stdin. The following input is handled by the locked stub code and passed to the removeDuplicates function:

The first line contains an integer, N, the number of nodes to be inserted.

The N subsequent lines each contain an integer describing the $data$ value of a node being inserted at the list's tail.

Constraints

- The data elements of the linked list argument will always be in non-decreasing order.

Output Format

Your removeDuplicates function should return the head of the updated linked list. The locked stub code in your editor will print the returned list to stdout.

Sample Input

```
6
1
2
2
3
3
4
```

Sample Output

```
1 2 3 4
```

Explanation

$N = 6$, and our non-decreasing list is $\{1, 2, 2, 3, 3, 4\}$. The values 2 and 3 both occur twice in the list, so we remove the two duplicate nodes. We then return our updated (ascending) list, which is $\{1, 2, 3, 4\}$.

```
15  public static Node removeDuplicates(Node head) {
16    LinkedList<Integer> remover = new LinkedList<Integer>();
17
18    while(head!=null){
19        if(!remover.contains(head.data)){
20            remover.add(head.data);
21        }
22        head=head.next;
23    }
24
25    for(int dato:remover){
26        head=insert(head, dato);
27    }
28
29
30    return head;
31  }
32
33  public static  Node insert(Node head,int data)
```

Line: 33 Col: 49

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 6 challenges away from the gold level for your 30 days of code badge.

25% 24/30

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Código día 24:

```
import java.io.*;
import java.util.*;
class Node{
 int data;
 Node next;
 Node(int d){
       data=d;
       next=null;
   }

}
class Solution
{

   public static Node removeDuplicates(Node head) {
     LinkedList<Integer> remover = new LinkedList<Integer>();

     while(head!=null){
        if(!remover.contains(head.data)){
           remover.add(head.data);
        }
        head=head.next;
     }

     for(int dato:remover){
        head=insert(head, dato);
     }


     return head;
   }

  public static  Node insert(Node head,int data)
    {
      Node p=new Node(data);
      if(head==null)
         head=p;
      else if(head.next==null)
         head.next=p;
      else
      {
         Node start=head;
         while(start.next!=null)
            start=start.next;
         start.next=p;

      }
      return head;
   }
   public static void display(Node head)
      {
```

```
        Node start=head;
        while(start!=null)
        {
            System.out.print(start.data+" ");
            start=start.next;
        }
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        Node head=null;
        int T=sc.nextInt();
        while(T-->0){
            int ele=sc.nextInt();
            head=insert(head,ele);
        }
        head=removeDuplicates(head);
        display(head);

    }
}
```

Task

A prime is a natural number greater than 1 that has no positive divisors other than 1 and itself. Given a number, n, determine and print whether it is `Prime` or `Not prime`.

Note: If possible, try to come up with a $O(\sqrt{n})$ primality algorithm, or see what sort of optimizations you come up with for an $O(n)$ algorithm. Be sure to check out the Editorial after submitting your code.

Input Format

The first line contains an integer, T, the number of test cases.

Each of the T subsequent lines contains an integer, n, to be tested for primality.

Constraints

- $1 \leq T \leq 30$
- $1 \leq n \leq 2 \times 10^9$

Output Format

For each test case, print whether n is `Prime` or `Not prime` on a new line.

Sample Input

```
3
12
5
7
```

Sample Output

```
Not prime
Prime
Prime
```

Explanation

Test Case 0: $n = 12$.

12 is divisible by numbers other than 1 and itself (i.e.: 2, 3, 4, 6), so we print `Not prime` on a new line.

Test Case 1: $n = 5$.

5 is only divisible 1 and itself, so we print `Prime` on a new line.

Test Case 2: $n = 7$.

7 is only divisible 1 and itself, so we print `Prime` on a new line.

```
            paridad="Not prime";
        }

        System.out.println(paridad);

    }

    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        int T=scan.nextInt();
        while(T-->0){
            int num=scan.nextInt();
            prime(num);
        }
    }
}
```

Line: 24 Col: 19

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 5 challenges away from the gold level for your 30 days of code badge.

38% 25/30

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Código día 25:

```
import java.io.*;
import java.util.*;

public class Solution {
   public static void prime(int num){
      String paridad="Prime";
      for(int i=1000;i>1;i--){
         if(num%i==0 && i!=num){
            paridad="Not prime";
            break;
         }
      }
      if(num==1){
         paridad="Not prime";
      }

      System.out.println(paridad);

   }

   public static void main(String[] args) {
      Scanner scan=new Scanner(System.in);
      int T=scan.nextInt();
      while(T-->0){
         int num=scan.nextInt();
         prime(num);
      }
   }
}
```

tab for a video on testing.

Task

Your local library needs your help! Given the expected and actual return dates for a library book, create a program that calculates the fine (if any). The fee structure is as follows:

1. If the book is returned on or before the expected return date, no fine will be charged (i.e.: $fine = 0$).
2. If the book is returned after the expected return day but still within the same calendar month and year as the expected return date,
   $fine = 15 \text{ Hackos} \times (\text{the number of days late})$.
3. If the book is returned after the expected return month but still within the same calendar year as the expected return date, the
   $fine = 500 \text{ Hackos} \times (\text{the number of months late})$.
4. If the book is returned after the calendar year in which it was expected, there is a fixed fine of 10000 Hackos.

Example

$d1, m1, y1 = 12312014$ returned date

$d2, m2, y2 = 112015$ due date

The book is returned on time, so no fine is applied.

$d1, m1, y1 = 112015$ returned date

$d2, m2, y2 = 12312014$ due date

The book is returned in the following year, so the fine is a fixed 10000.

Input Format

The first line contains 3 space-separated integers denoting the respective day, $month$, and $year$ on which the book was actually returned.

The second line contains 3 space-separated integers denoting the respective day, $month$, and $year$ on which the book was expected to be returned (due date).

Constraints

```
        }else{
            if(monthReturn<monthDue){
                fine=0;
            }else{
                fine=500*(monthReturn-monthDue);
            }
        }
    }else{
        if(yearReturn<yearDue){
            fine=0;
```

Line: 4 Col: 24

Upload Code as File | Test against custom input | Run Code | Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 4 challenges away from the gold level for your 30 days of code badge.

50% 26/30

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Compiler Message

Success

Input (stdin) Download

```
31 8 2004
```

Código día 26:

```
import java.io.*;
import java.util.*;

public class Solution {

   public static void main(String[] args) {
      Scanner scan = new Scanner(System.in);
      int dayReturn=scan.nextInt();
      int monthReturn=scan.nextInt();
      int yearReturn=scan.nextInt();

      int dayDue=scan.nextInt();
      int monthDue=scan.nextInt();
      int yearDue=scan.nextInt();
      int fine;

      if(yearReturn==yearDue){
         if(monthReturn==monthDue){
            if(dayReturn>dayDue){
               fine=15*(dayReturn-dayDue);
            }else{
               fine=0;
            }
         }else{
            if(monthReturn<monthDue){
               fine=0;
            }else{
               fine=500*(monthReturn-monthDue);
            }
         }
      }else{
         if(yearReturn<yearDue){
            fine=0;
         }else{
            fine=10000;
         }
      }

      System.out.println(fine);
   }
}
```

Your company needs a function that meets the following requirements:

- For a given array of n integers, the function returns the index of the element with the minimum value in the array. If there is more than one element with the minimum value, it returns the smallest one.
- If an empty array is passed to the function, it raises an exception. A colleague has written this method. The implementation in Python is listed below. Implementations in other languages can be found in the code template.

```
def minimum_index(seq):
    if len(seq) == 0:
        raise ValueError("Cannot get the minimum value index from an empt
    min_idx = 0
    for i in range(1, len(seq)):
        if a[i] < a[min_idx]:
            min_idx = i
    return min_idx
```

A coworker has prepared functions that will perform the tests and validate return values. Finish the implementation of **3** classes to provide data and expected results for the tests.

Complete the following methods.

In the class `TestDataEmptyArray`:

- `get_array()` returns an empty array

In the class `TestDataUniqueValues`:

- `get_array()` returns an array of size at least 2 with all unique elements
- `get_expected_result()` returns the expected minimum value index for this array

In the class `TestDataExactlyTwoDifferentMinimums`:

- `get_array()` returns an array where the minimum value occurs at exactly 2 indices
- `get_expected_result()` returns the expected index

Take a look at the code template to see the exact implementation of functions that your colleague already implemented.

Note: The arrays are indexed from **0**.

```
}

public static class TestDataExactlyTwoDifferentMinimums{
    public static int[] get_array(){
        int[] lista={5,9,2,7,2};
        return lista;
    }

    public static int get_expected_result(){
        int[] lista={5,9,2,7,2};
        int minimo=0;
        for(int i=1;i<lista.length;i++){
            if(lista[i]<lista[minimo]){
                minimo=i;
            }
        }
        return minimo;
    }

}
```

Line: 8 Col: 6

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!
You are now 3 challenges away from the gold level for your 30 days of code badge.

63% 27/30

Congratulations
You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Compiler Message

Success

Código día 27:

```
import java.util.*;

public class Solution {

   public static int minimum_index(int[] seq) {
      if (seq.length == 0) {
         throw new IllegalArgumentException("Cannot get the minimum value index from an empty sequence");
      }
      int min_idx = 0;
      for (int i = 1; i < seq.length; ++i) {
         if (seq[i] < seq[min_idx]) {
            min_idx = i;
         }
      }
      return min_idx;
   }

public static class TestDataEmptyArray{
   public static int[] get_array(){
      int[] empty={};
      return empty;
   }
}

public static class TestDataUniqueValues{
   public static int[] get_array(){
      int[] lista={3,5};
      return lista;
   }

   public static int get_expected_result(){
      int[] lista={3,5};
      int minimo=0;
      for(int i=1;i<lista.length;i++){
         if(lista[i]<lista[minimo]){
            minimo=i;
         }
      }
      return minimo;
   }

}

public static class TestDataExactlyTwoDifferentMinimums{
   public static int[] get_array(){
      int[] lista={5,9,2,7,2};
      return lista;
   }

   public static int get_expected_result(){
      int[] lista={5,9,2,7,2};
```

```
        int minimo=0;
        for(int i=1;i<lista.length;i++){
            if(lista[i]<lista[minimo]){
                minimo=i;
            }
        }
        return minimo;
    }

}
```

Constraints

- $2 \leq N \leq 30$
- Each of the first names consists of lower case letters $[a - z]$ only.
- Each of the email IDs consists of lower case letters $[a - z]$, @ and . only.
- The length of the first name is no longer than 20.
- The length of the email ID is no longer than 50.

Output Format

Print an alphabetically-ordered list of first names for every user with a gmail account. Each name must be printed on a new line.

Sample Input

```
6
riya riya@gmail.com
julia julia@julia.me
julia sjulia@gmail.com
julia julia@gmail.com
samantha samantha@gmail.com
tanya tanya@gmail.com
```

Sample Output

```
julia
julia
riya
samantha
tanya
```

```
34              aux=nombres.get(j);
35              nombres.set(j, nombres.get(j+1));
36              nombres.set(j+1, aux);
37          }else if(nombres.get(j).charAt(0)==nombres.get(j+1).charAt(0)){
38
39
40              while(nombres.get(j).charAt(igual)==nombres.get(j+1).charAt(igual) && igual!=nombres.get(j).length()-1){
41                  igual++;
42                  if(nombres.get(j).charAt(igual)!=nombres.get(j+1).charAt(igual)){
43                      aux=nombres.get(j);
44                      nombres.set(j, nombres.get(j+1));
45                      nombres.set(j+1, aux);
46                  }
47              }
48          }
```

Line: 9 Col: 27

Upload Code as File | Test against custom input | Run Code | Submit Code

30 Days of Code

You have earned 30.00 points!

You are now 2 challenges away from the gold level for your 30 days of code badge.

75% 28/30

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin) Download

1 6

Código día 28:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



public class Solution {
   static LinkedList<String> nombres=new LinkedList<String>();
   public static void guardar(String nombre, String email){
      Pattern patron=Pattern.compile("@gmail.com");
      Matcher matcher=patron.matcher(email);

      if(matcher.find()){
         nombres.add(nombre);
      }
   }

   public static void ordenar(LinkedList<String> nombres){
      String aux;
      int igual=1;

      for(int i=0;i<nombres.size()-1;i++){
         for(int j=0;j<nombres.size()-1;j++){
            //System.out.println(nombres.get(1).charAt(0)!=nombres.get(2).charAt(0));
            if(nombres.get(j).charAt(0)>nombres.get(j+1).charAt(0)){
               aux=nombres.get(j);
               nombres.set(j, nombres.get(j+1));
               nombres.set(j+1, aux);
            }else if(nombres.get(j).charAt(0)==nombres.get(j+1).charAt(0)){


               while(nombres.get(j).charAt(igual)==nombres.get(j+1).charAt(igual) && igual!=nombres.get(j).length(
)-1){
                  igual++;
                  if(nombres.get(j).charAt(igual)!=nombres.get(j+1).charAt(igual)){
                     aux=nombres.get(j);
                     nombres.set(j, nombres.get(j+1));
                     nombres.set(j+1, aux);
                  }
               }
            }
            igual=0;

         }
```

```
        }

        for(String name:nombres){
            System.out.println(name);
        }
    }

    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

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

        IntStream.range(0, N).forEach(NItr -> {
            try {
                String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");

                String firstName = firstMultipleInput[0];

                String emailID = firstMultipleInput[1];

                guardar(firstName,emailID);
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        });

        bufferedReader.close();

        ordenar(nombres);


    }
}
```

- $1 \leq T \leq 10^3$
- $2 \leq N \leq 10^3$
- $2 \leq K \leq N$

Sample Input

```
STDIN   Function
-----   --------
3       T = 3
5 2     N = 5, K = 2
8 5     N = 8, K = 5
2 2     N = 8, K = 5
```

Sample Output

```
1
4
0
```

Explanation

$N = 5, K = 2$ $S = \{1, 2, 3, 4, 5\}$

All possible values of A and B are:

1. $A = 1, B = 2$; A & B $= 0$
2. $A = 1, B = 3$; A & B $= 1$
3. $A = 1, B = 4$; A & B $= 0$
4. $A = 1, B = 5$; A & B $= 1$
5. $A = 2, B = 3$; A & B $= 2$
6. $A = 2, B = 4$; A & B $= 0$
7. $A = 2, B = 5$; A & B $= 0$
8. $A = 3, B = 4$; A & B $= 0$
9. $A = 3, B = 5$; A & B $= 1$
10. $A = 4, B = 5$; A & B $= 4$

The maximum possible value of $A \& B$ that is also $< (K = 2)$ is 1, so we print 1 on a new line.

```
                for(int j=i+1;j<=N;j++){

                    if((i&j)<K && (i&j)>max){
                        max=i&j;
                    }
                }
            }
            return max;
        }

    }

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in))
;
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv
("OUTPUT_PATH")));

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

        IntStream.range(0, t).forEach(tItr -> {
            try {
                String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$",
"").split(" ");
```

Line: 13 Col: 15

Upload Code as File Test against custom input Run Code Submit Code

30 Days of Code

You have earned 30.00 points!

30/30 challenges solved.

100%

Congratulations

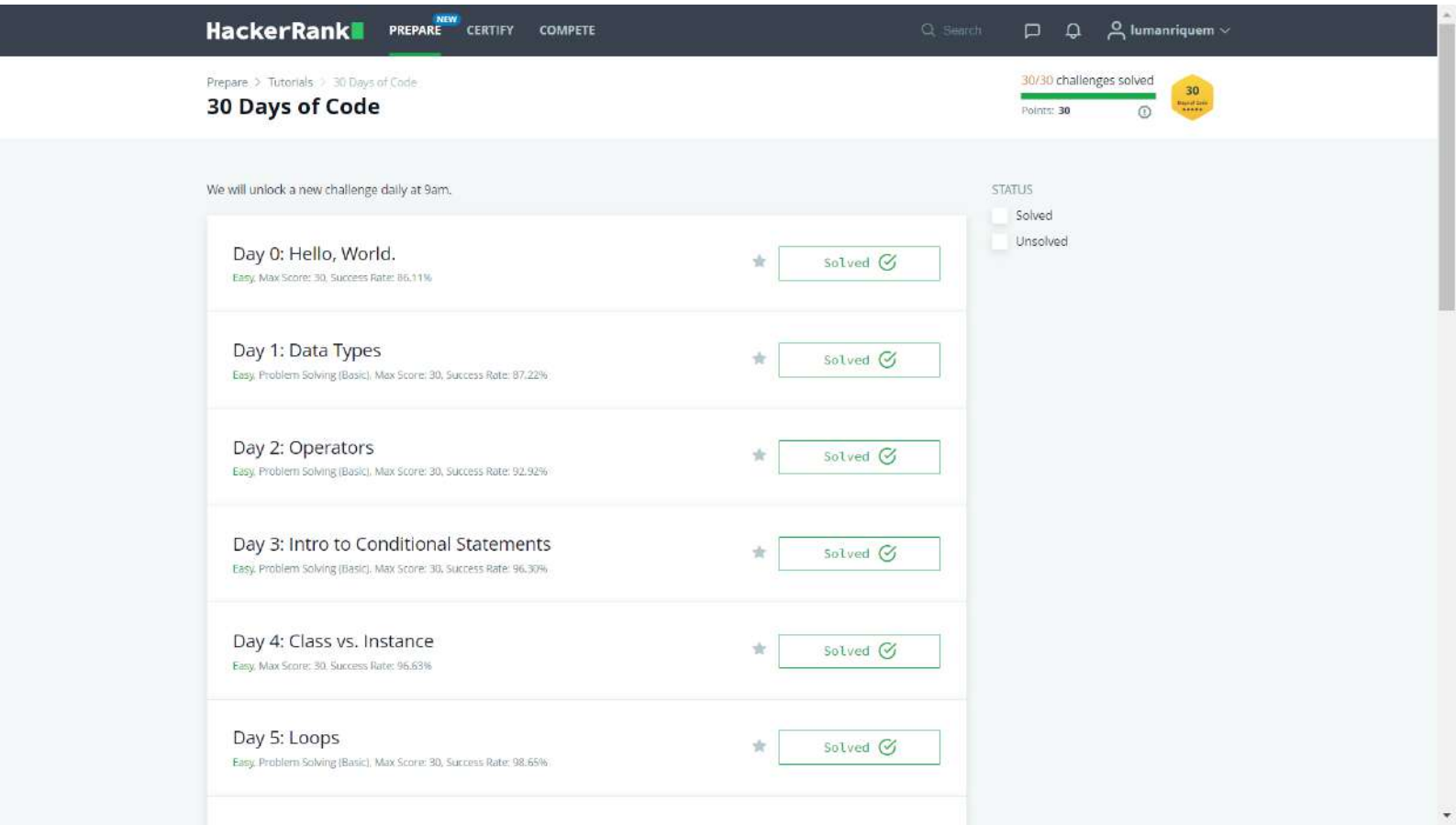
You solved this challenge. Would you like to challenge your friends?

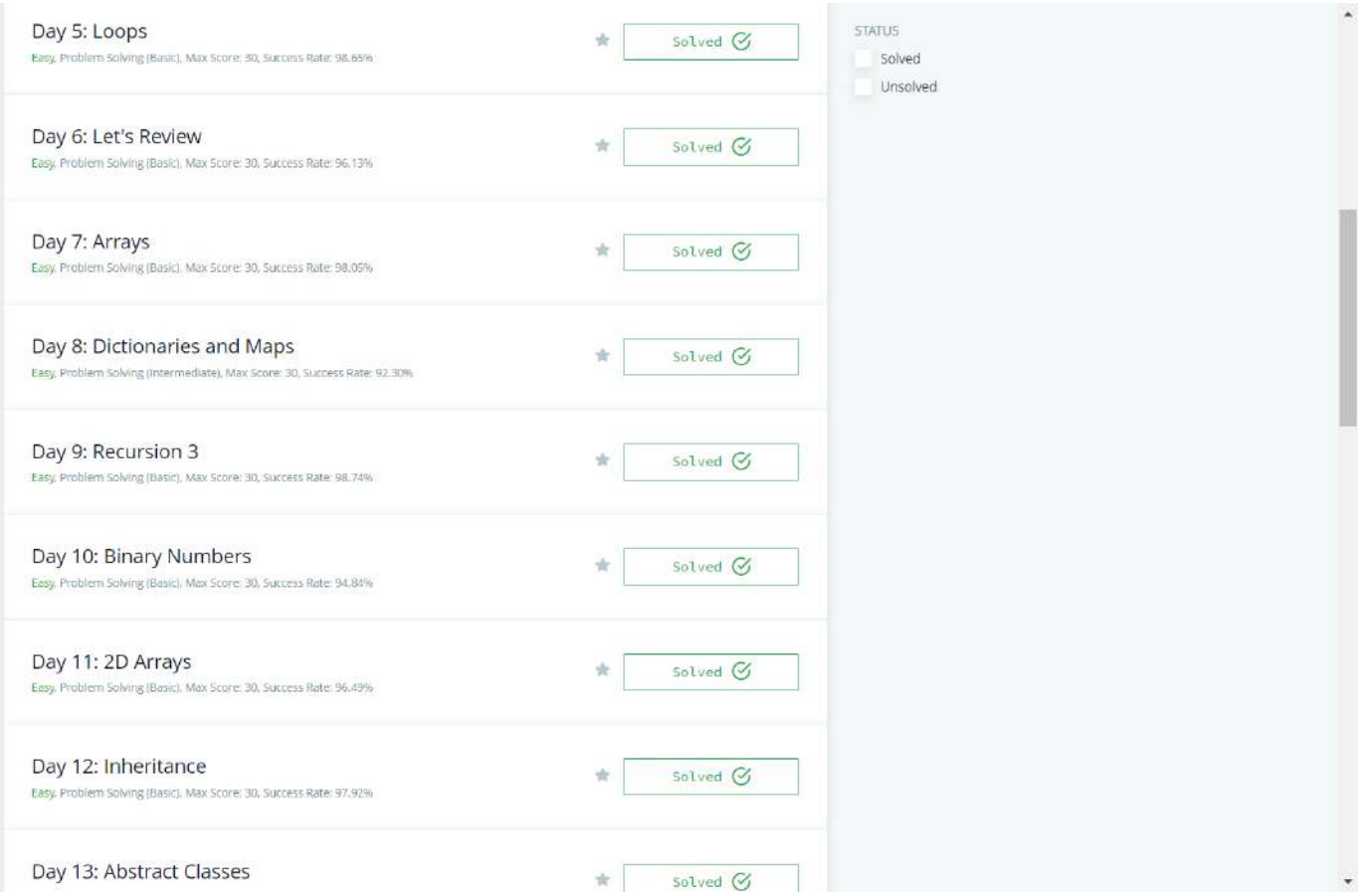
Next Challenge

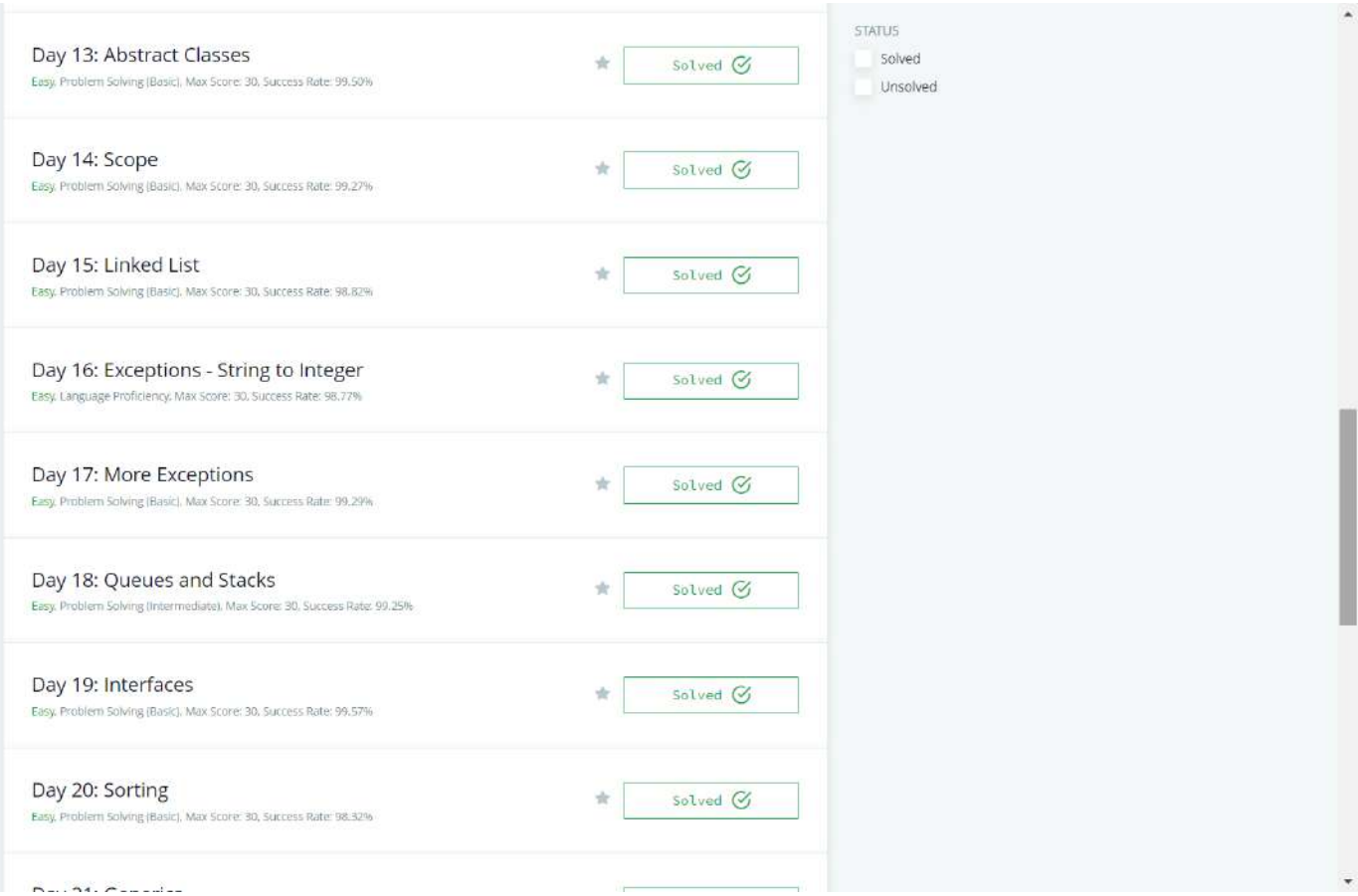
Código día 29:

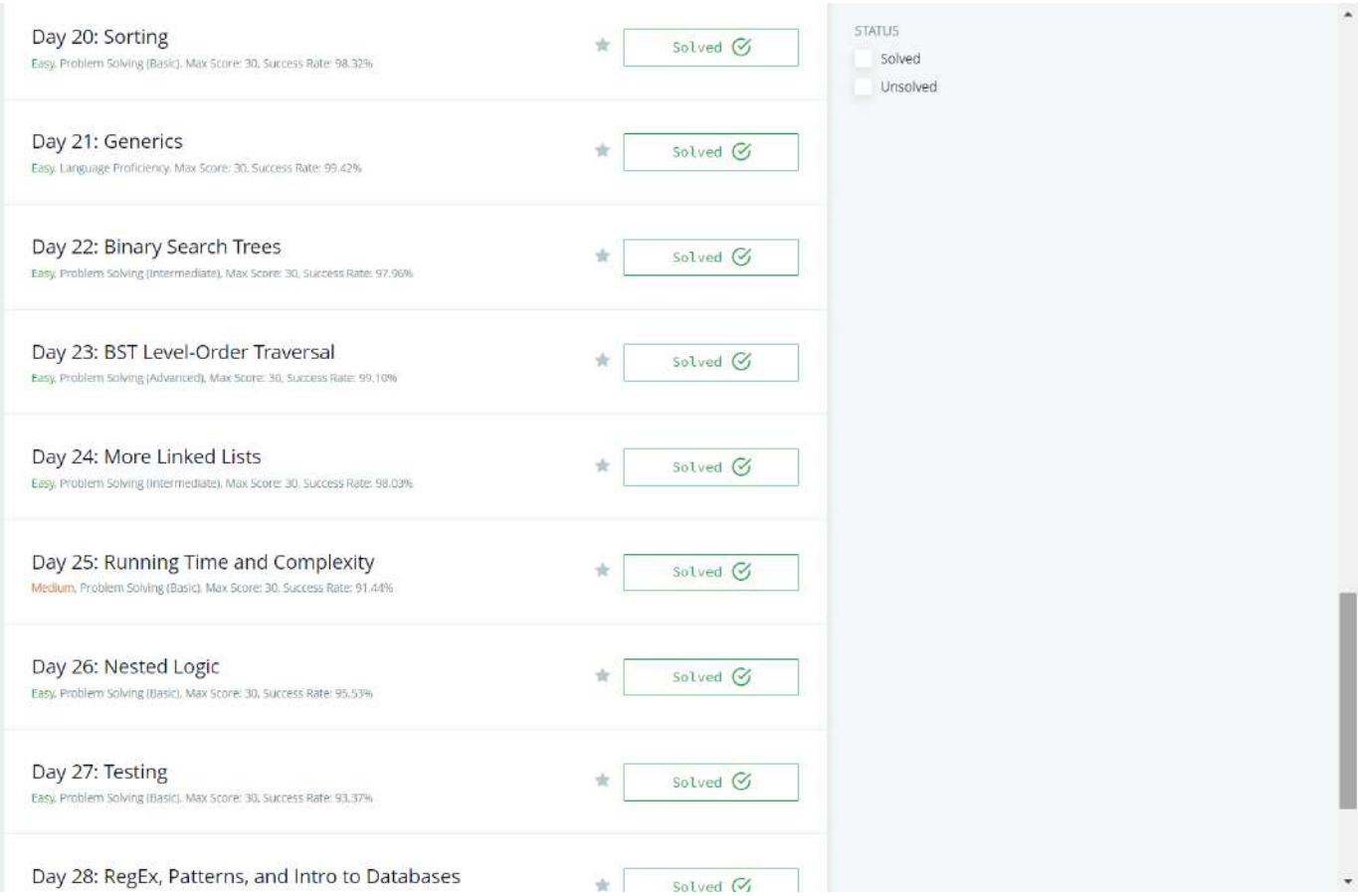
```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'bitwiseAnd' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER N
     *  2. INTEGER K
     */

    public static int bitwiseAnd(int N, int K) {
        int max=0;

        for(int i=1;i<N;i++){
            for(int j=i+1;j<=N;j++){

                if((i&j)<K && (i&j)>max){
                    max=i&j;
                }
            }
        }
        return max;
    }

}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

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

        IntStream.range(0, t).forEach(tItr -> {
            try {
                String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");

                int count = Integer.parseInt(firstMultipleInput[0]);
```

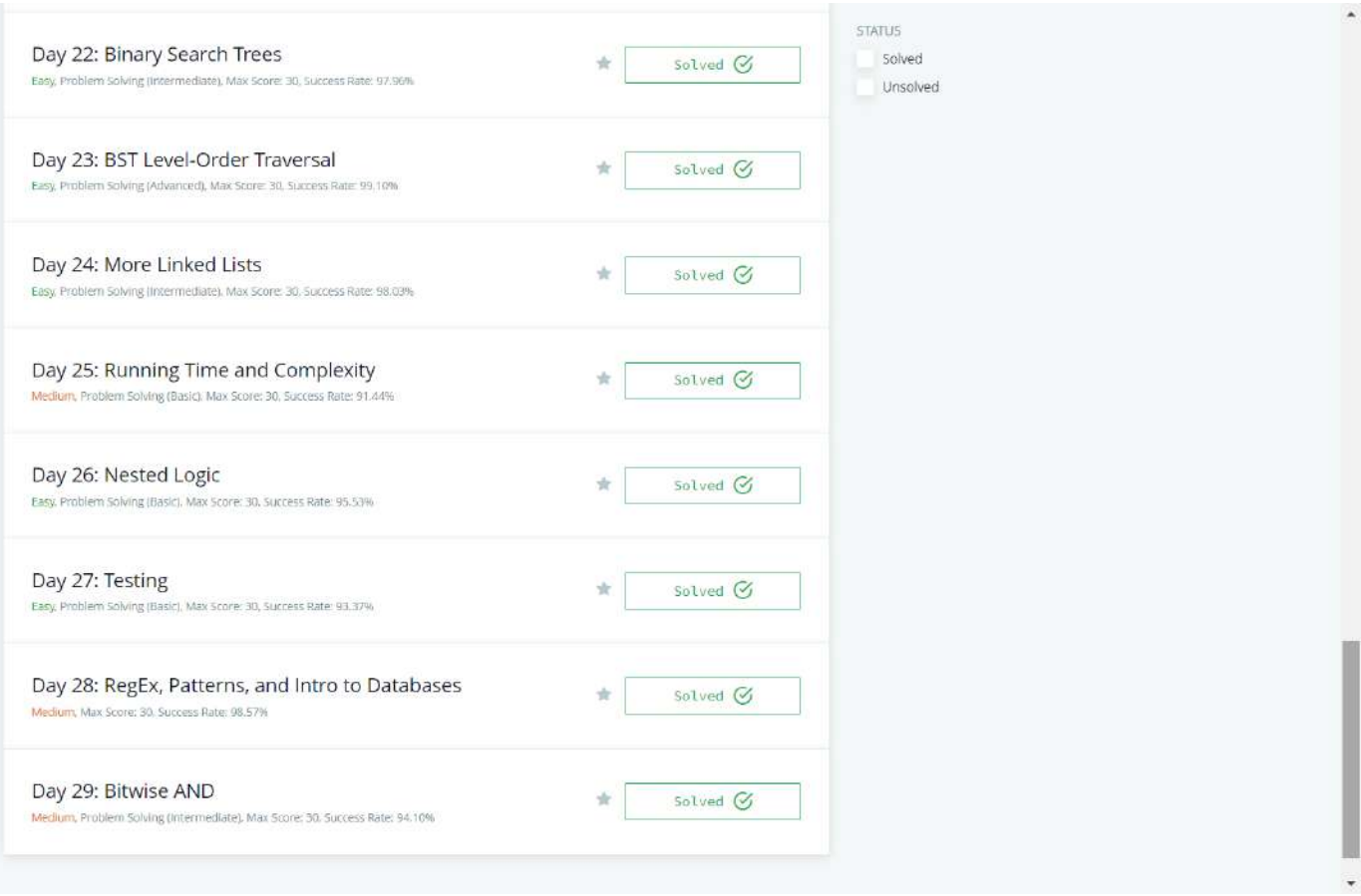
```
            int lim = Integer.parseInt(firstMultipleInput[1]);

            int res = Result.bitwiseAnd(count, lim);

            bufferedWriter.write(String.valueOf(res));
            bufferedWriter.newLine();
        } catch (IOException ex) {
            throw new RuntimeException(ex);
        }
    });

    bufferedReader.close();
    bufferedWriter.close();
  }
}
```


HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
lumanriquem
Prepare > Tutorials > 30 Days of Code
30 Days of Code
30/30 challenges solved
Points: 30
30
We will unlock a new challenge daily at 9am.
STATUS
Solved
Unsolved
Day 0: Hello, World.
Easy, Max Score: 30, Success Rate: 86.11%
Solved
Day 1: Data Types
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 87.22%
Solved
Day 2: Operators
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 92.92%
Solved
Day 3: Intro to Conditional Statements
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 96.30%
Solved
Day 4: Class vs. Instance
Easy, Max Score: 30, Success Rate: 96.63%
Solved
Day 5: Loops
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.65%
Solved


Day 5: Loops
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.65%
Solved
Day 6: Let's Review
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 96.13%
Solved
Day 7: Arrays
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.05%
Solved
Day 8: Dictionaries and Maps
Easy, Problem Solving (Intermediate), Max Score: 30, Success Rate: 92.30%
Solved
Day 9: Recursion 3
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.74%
Solved
Day 10: Binary Numbers
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 94.84%
Solved
Day 11: 2D Arrays
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 96.49%
Solved
Day 12: Inheritance
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 97.92%
Solved
Day 13: Abstract Classes
Solved
STATUS
Solved
Unsolved


Day 13: Abstract Classes
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 99.50%
Solved
Day 14: Scope
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 99.27%
Solved
Day 15: Linked List
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.82%
Solved
Day 16: Exceptions - String to Integer
Easy, Language Proficiency, Max Score: 30, Success Rate: 98.77%
Solved
Day 17: More Exceptions
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 99.29%
Solved
Day 18: Queues and Stacks
Easy, Problem Solving (Intermediate), Max Score: 30, Success Rate: 99.25%
Solved
Day 19: Interfaces
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 99.57%
Solved
Day 20: Sorting
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.32%
Solved
STATUS
Solved
Unsolved


Day 20: Sorting
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 98.32%
Solved
Day 21: Generics
Easy, Language Proficiency, Max Score: 30, Success Rate: 99.42%
Solved
Day 22: Binary Search Trees
Easy, Problem Solving (Intermediate), Max Score: 30, Success Rate: 97.96%
Solved
Day 23: BST Level-Order Traversal
Easy, Problem Solving (Advanced), Max Score: 30, Success Rate: 99.10%
Solved
Day 24: More Linked Lists
Easy, Problem Solving (Intermediate), Max Score: 30, Success Rate: 98.03%
Solved
Day 25: Running Time and Complexity
Medium, Problem Solving (Basic), Max Score: 30, Success Rate: 91.44%
Solved
Day 26: Nested Logic
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 95.53%
Solved
Day 27: Testing
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 93.37%
Solved
Day 28: RegEx, Patterns, and Intro to Databases
Solved
STATUS
Solved
Unsolved


Day 22: Binary Search Trees
Easy, Problem Solving (Intermediate), Max Score: 30, Success Rate: 97.96%
Solved
Day 23: BST Level-Order Traversal
Easy, Problem Solving (Advanced), Max Score: 30, Success Rate: 99.10%
Solved
Day 24: More Linked Lists
Easy, Problem Solving (Intermediate), Max Score: 30, Success Rate: 98.03%
Solved
Day 25: Running Time and Complexity
Medium, Problem Solving (Basic), Max Score: 30, Success Rate: 91.44%
Solved
Day 26: Nested Logic
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 95.53%
Solved
Day 27: Testing
Easy, Problem Solving (Basic), Max Score: 30, Success Rate: 93.37%
Solved
Day 28: RegEx, Patterns, and Intro to Databases
Medium, Max Score: 30, Success Rate: 98.57%
Solved
Day 29: Bitwise AND
Medium, Problem Solving (Intermediate), Max Score: 30, Success Rate: 94.10%
Solved
STATUS
Solved
Unsolved