JAVA PROGRAMS

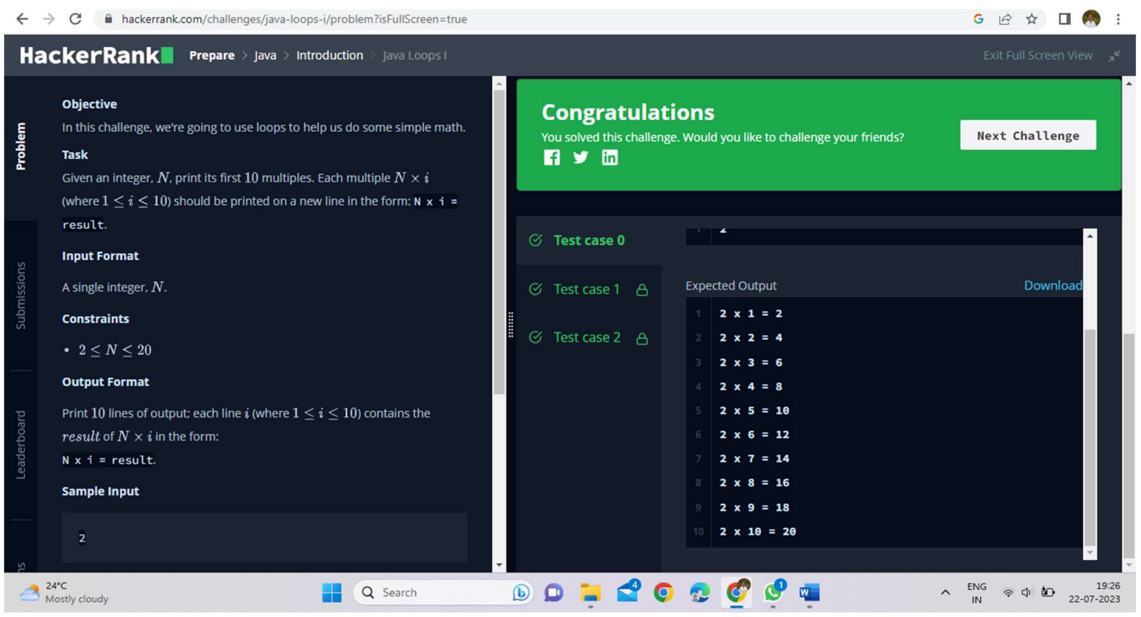
1. Java Loops I

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

public class Solution {

    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();
        for (int i = 1; i <= 10; i++) {
            int result = N * i;
            System.out.println(N + " x " + i + " = " + result);
        }
    }
}
```



Objective

In this challenge, we're going to use loops to help us do some simple math.

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`.

Input Format

A single integer, N.

Constraints

- $2 \leq N \leq 20$

Output Format

Print 10 lines of output; each line i (where $1 \leq i \leq 10$) contains the $result$ of $N \times i$ in the form:
`N x i = result`.

Sample Input

```
2
```

Congratulations

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

Expected Output

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

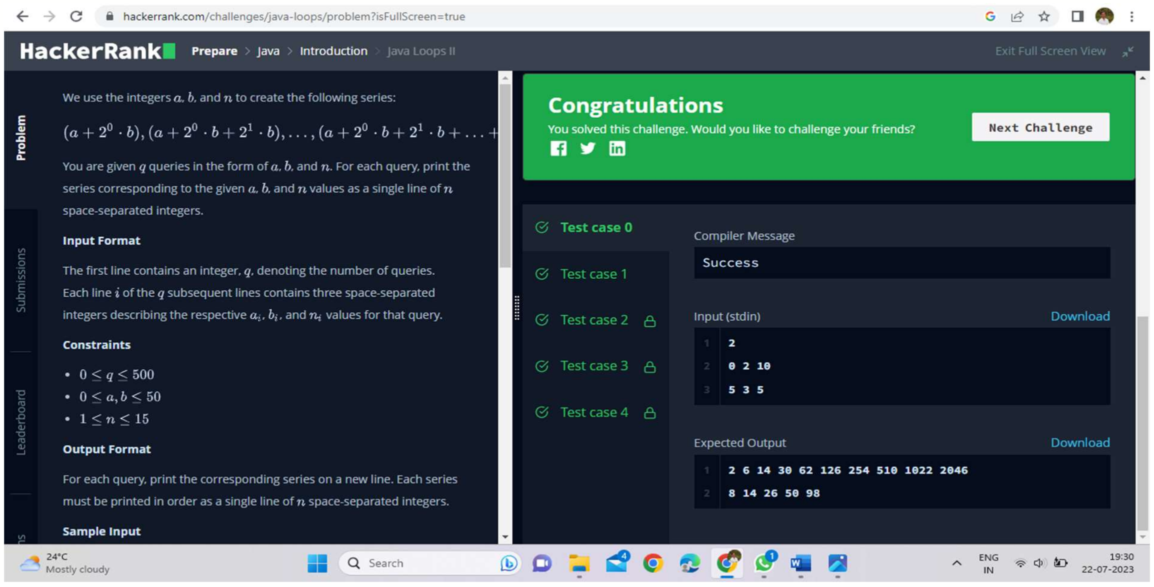
2. Java Loops II

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

class Solution{
    public static void main(String []argh){
        Scanner in = new Scanner(System.in);
        int t=in.nextInt();
        for(int i=0;i<t;i++){
            int a = in.nextInt();
            int b = in.nextInt();
            int n = in.nextInt();
            printSeries(a, b, n);
        }
        in.close();
    }
    private static void printSeries(int a, int b, int n) {
        int result = a;

        // Iterate from 0 to n-1
        for (int i = 0; i < n; i++) {
            // Add (2^i * b) to the result
            result += (int) Math.pow(2, i) * b;
            System.out.print(result + " ");
        }

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


hackerrank.com/challenges/java-loops/problem?isFullScreen=true
HackerRank
Prepare > Java > Introduction > Java Loops II
Exit Full Screen View
Problem
Submissions
Leaderboard

We use the integers a, b, and n to create the following series:

$$(a + 2^0 \cdot b), (a + 2^0 \cdot b + 2^1 \cdot b), \ldots, (a + 2^0 \cdot b + 2^1 \cdot b + \ldots +$$

You are given q queries in the form of a, b, and n. For each query, print the series corresponding to the given a, b, and n values as a single line of n space-separated integers.

Input Format

The first line contains an integer, q, denoting the number of queries. Each line i of the q subsequent lines contains three space-separated integers describing the respective a_i, b_i, and n_i values for that query.

Constraints

- $0 \le q \le 500$
- $0 \le a, b \le 50$
- $1 \le n \le 15$

Output Format

For each query, print the corresponding series on a new line. Each series must be printed in order as a single line of n space-separated integers.

Sample Input

Congratulations

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

Compiler Message

```
Success
```

Input (stdin)

```
2
0 2 10
5 3 5
```

Expected Output

```
2 6 14 30 62 126 254 510 1022 2046
8 14 26 50 98
```

3. Java End of File

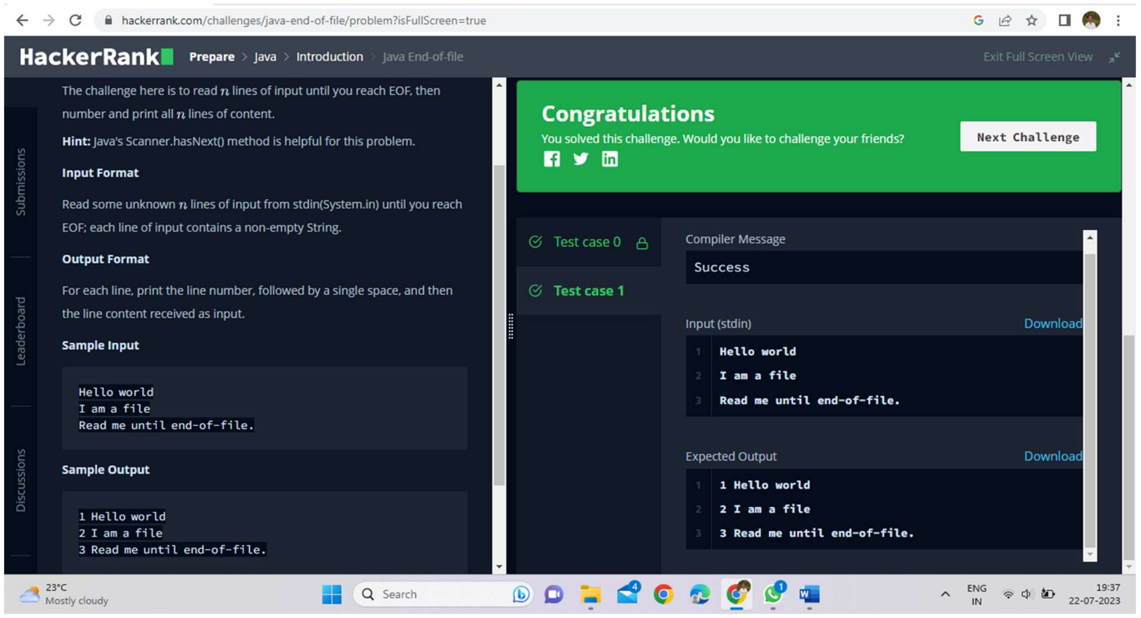
```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class s
hould be named Solution. */
        Scanner scanner = new Scanner(System.in);

        int lineNumber = 1;
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            System.out.println(lineNumber + " " + line);
            lineNumber++;
        }

        scanner.close();
    }
}
```



hackerrank.com/challenges/java-end-of-file/problem?isFullScreen=true

HackerRank Prepare > Java > Introduction > Java End-of-file

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

Hint: Java's Scanner.hasNext() method is helpful for this problem.

Input Format

Read some unknown n lines of input from stdin(System.in) until you reach EOF; each line of input contains a non-empty String.

Output Format

For each line, print the line number, followed by a single space, and then the line content received as input.

Sample Input

```
Hello world
I am a file
Read me until end-of-file.
```

Sample Output

```
1 Hello world
2 I am a file
3 Read me until end-of-file.
```

Congratulations

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

Test case 0
Test case 1

Compiler Message

```
Success
```

Input (stdin)

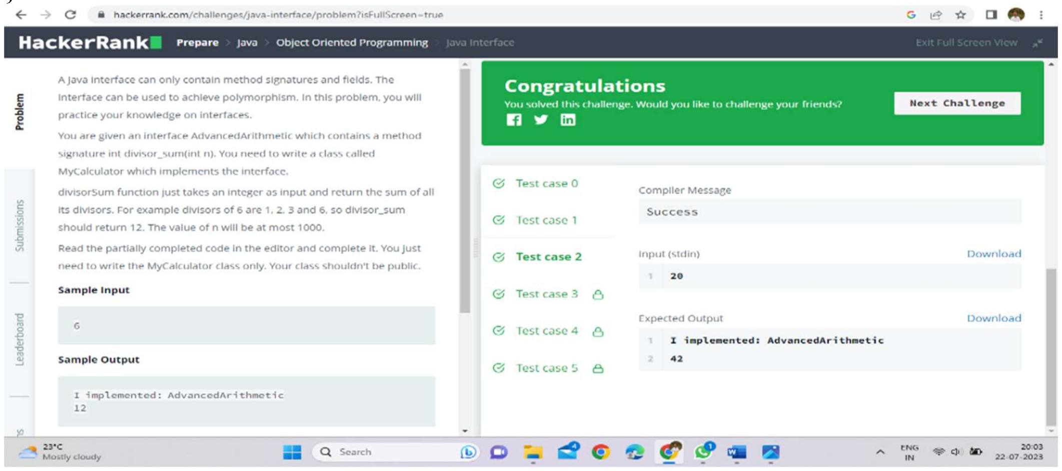
```
Hello world
I am a file
Read me until end-of-file.
```

Expected Output

```
1 Hello world
2 I am a file
3 Read me until end-of-file.
```

4. Java Interface

```
import java.util.*;
interface AdvancedArithmetic{
 int divisor_sum(int n);
}
//Write your code here
class MyCalculator implements AdvancedArithmetic {
    public int divisor_sum(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                sum += i;
            }
            }
        return sum;
    }}
class Solution{
    public static void main(String []args){
        MyCalculator my_calculator = new MyCalculator();
        System.out.print("I implemented: ");
        ImplementedInterfaceNames(my_calculator);
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.print(my_calculator.divisor_sum(n) + "\n");
        sc.close();
    }
    static void ImplementedInterfaceNames(Object o){
        Class[] theInterfaces = o.getClass().getInterfaces();
        for (int i = 0; i < theInterfaces.length; i++){
            String interfaceName = theInterfaces[i].getName();
            System.out.println(interfaceName);
        }
    }
}
```


hackerrank.com/challenges/java-interface/problem?isFullScreen=true
HackerRank
Prepare > Java > Object Oriented Programming > Java Interface
Exit Full Screen View
Problem
Submissions
Leaderboard
Next Challenge
23°C
Mostly cloudy
Search
ENG IN
20:03
22-07-2023

A Java interface can only contain method signatures and fields. The interface can be used to achieve polymorphism. In this problem, you will practice your knowledge on interfaces.

You are given an interface AdvancedArithmetic which contains a method signature int divisor_sum(int n). You need to write a class called MyCalculator which implements the interface.

divisorSum function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so divisor_sum should return 12. The value of n will be at most 1000.

Read the partially completed code in the editor and complete it. You just need to write the MyCalculator class only. Your class shouldn't be public.

Sample Input

```
6
```

Sample Output

```
I implemented: AdvancedArithmetic
12
```

Congratulations

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

✓ Test case 0
✓ Test case 1
✓ Test case 2
✓ Test case 3
✓ Test case 4
✓ Test case 5

Compiler Message

```
Success
```

Input (stdin) Download

```
20
```

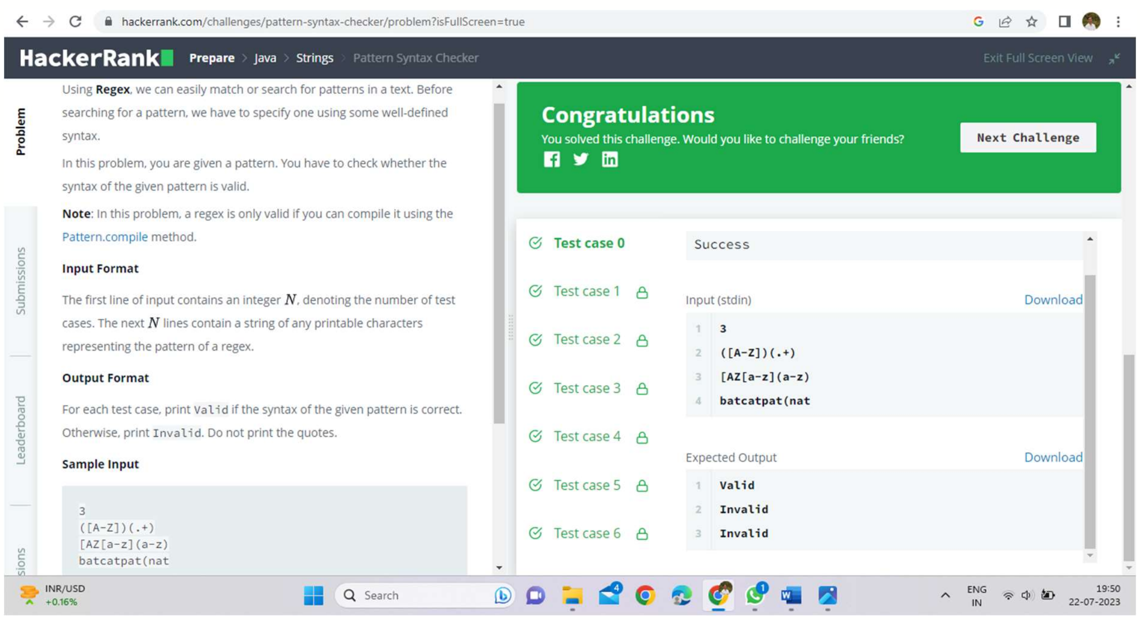
Expected Output Download

```
I implemented: AdvancedArithmetic
42
```

5. Java Pattern Syntax Checker

```
import java.util.Scanner;
import java.util.regex.*;

public class Solution
{
   public static void main(String[] args){
      Scanner in = new Scanner(System.in);
      int testCases = Integer.parseInt(in.nextLine());
      while(testCases>0){
         String pattern = in.nextLine();
         //Write your code
           try {
            // Attempt to compile the pattern
            Pattern.compile(pattern);
            System.out.println("Valid");
         } catch (PatternSyntaxException e) {
            // If an exception is thrown, it's an invalid pattern.
            System.out.println("Invalid");
         }
         testCases--;
      }
   }
}
```


hackerrank.com/challenges/pattern-syntax-checker/problem?isFullScreen=true
HackerRank
Prepare > Java > Strings > Pattern Syntax Checker
Exit Full Screen View
Problem
Submissions
Leaderboard
Next Challenge
INR/USD +0.16%
Search
ENG IN
19:50 22-07-2023

Using **Regex**, we can easily match or search for patterns in a text. Before searching for a pattern, we have to specify one using some well-defined syntax.

In this problem, you are given a pattern. You have to check whether the syntax of the given pattern is valid.

Note: In this problem, a regex is only valid if you can compile it using the Pattern.compile method.

Input Format

The first line of input contains an integer N, denoting the number of test cases. The next N lines contain a string of any printable characters representing the pattern of a regex.

Output Format

For each test case, print `Valid` if the syntax of the given pattern is correct. Otherwise, print `Invalid`. Do not print the quotes.

Sample Input

```
3
([A-Z])(.+)
[AZ[a-z](a-z)
batcatpat(nat
```

Congratulations

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

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

```
Success
```

Input (stdin)

```
3
([A-Z])(.+)
[AZ[a-z](a-z)
batcatpat(nat
```

Expected Output

```
Valid
Invalid
Invalid
```