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

Exit Full Screen View

Problem

Submissions

Leaderboard

"In computing, End Of File (commonly abbreviated EOF) is a condition in a computer operating system where no more data can be read from a data source." — (Wikipedia: End-of-file)

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

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

#### **Input Format**

Read some unknown 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.

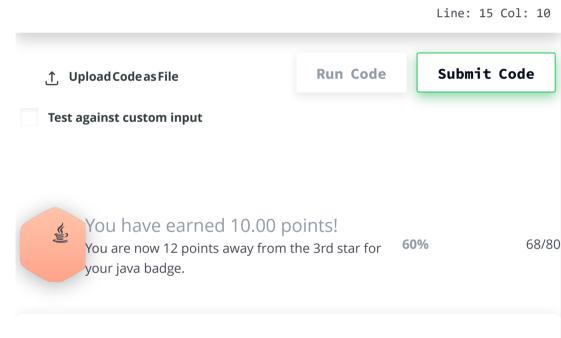
#### Sample Input

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.

```
Change Theme
             Language | Java 7
                                                 100
    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 STDI
    Your class should be named Solution. */
             Scanner input = new Scanner (System.in);
             int n = 1;
            while (input.hasNext()){
                 System.out.printf("%d %s\n" , n++ , input
```



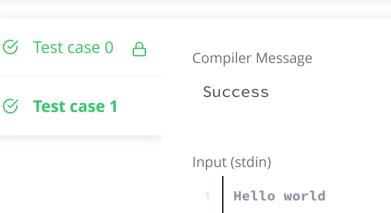


You solved this challenge. Would you like to challenge your friends?

Next Challenge

Download

Download



I am a file
Read me until end-of-file.

1 Hello world
2 I am a file

**Expected Output** 

**Prepare** > Java > Introduction > Java Datatypes

Exit Full Screen View

Java has 8 primitive data types; char, boolean, byte, short, int, long, float, and double. For this exercise, we'll work with the primitives used to hold integer values (byte, short, int, and long):

- A byte is an 8-bit signed integer.
- A short is a 16-bit signed integer.
- An int is a 32-bit signed integer.
- A long is a 64-bit signed integer.

Given an input integer, you must determine which primitive data types are capable of properly storing that input.

To get you started, a portion of the solution is provided for you in the editor.

#### Reference:

Problem

Submissions

Leaderboard

https://docs.oracle.com/javase/tutorial/java/nutsandb olts/datatypes.html

#### **Input Format**

The first line contains an integer, \_ , denoting the number of test cases.

Each test case, \_ , is comprised of a single line with an integer, ,, which can be arbitrarily large or small.

#### **Output Format**

For each input variable and appropriate primitive , you must determine if the given primitives are capable of storing it. If yes, then print:

```
n can be fitted in:
* dataType
```

If there is more than one appropriate data type, print each one on its own line and order them by size (i.e.:

If the number cannot be stored in one of the four aforementioned primitives, print the line:

n can't be fitted anywhere.

## Sample Input

```
5
-150
150000
1500000000
2133333333333333333333333333333333333
-1000000000000000
```

## **Sample Output**

```
-150 can be fitted in:
```

```
* short
```

```
Change Theme
                    Language | Java 7
                                                          (O)
              Scanner sc = new Scanner(System.in);
              int t=sc.nextInt();
              for(int i=0;i<t;i++)</pre>
                  try
                      long x=sc.nextLong();
                      System.out.println(x+" can be fitted in:")
                      if(x \ge -128 && x \le 127)System.out.println("*
                      //Complete the code
                      if(x>=-32768 \&\& x<= 32767)System.out.print
                  if(x)=Math.pow(-2,31) \&\& x<= (Math.pow(2,31)-1)
                  if(x)=Math.pow(-2,63) \&\& x<= (Math.pow(2,63)-1)
                  catch(Exception e)
                      System.out.println(sc.next()+" can't be fi
                                                      Line: 24 Col: 74
                                                      Submit Code
↑ Upload Code as File
                                       Run Code
Test against custom input
```

# You have earned 10.00 points!

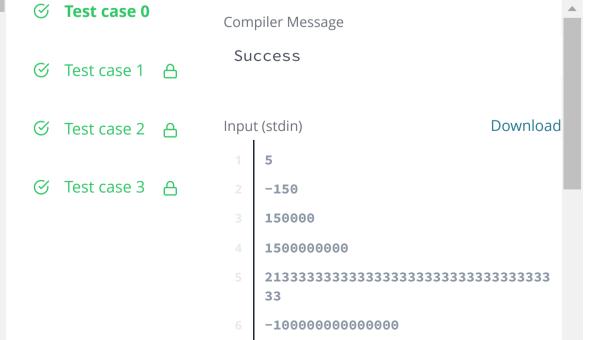
You are now 22 points away from the 3rd star for your java badge.

58/80

# **Congratulations**

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

**Next Challenge** 



2/13/23, 9:53 AM Java Loops II | HackerRank

Prepare > Java > Introduction > Java Loops II We use the integers and to create the following series: Problem You are given a queries in the form of \_\_\_, and \_\_\_. For each query, print the series corresponding to the given \_\_\_\_, and \_\_\_\_ values as a single line of space-separated integers. **Input Format** Submissions The first line contains an integer, a, denoting the number of queries. Each line of the subsequent lines contains three spaceseparated integers describing the respective  $_{\kappa}$  ,  $_{\kappa}$  , and  $_{\kappa}$ values for that query. **Constraints** Leaderboard **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 space-separated integers. Sample Input 0 2 10 5 3 5 Sample Output 2 6 14 30 62 126 254 510 1022 2046 8 14 26 50 98 **Explanation** We have two queries: to produce some 1. We use \_\_\_\_\_\_, and \_\_\_\_ series 🛌 . ... and so on. Once we hit \_\_\_\_\_, we print the first ten terms as a single line of space-separated integers.

```
Change Theme
                                                     Language Java 7
                                                                                                                                                                               (O)
                      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++){</pre>
                                                                int a = in.nextInt();
                                                                int b = in.nextInt();
                                                                int n = in.nextInt();
                                                                System.out.print((a=a+b) +" ");
                                                                for (int j = 0; j < n-1; j++) {
                                                                              System.out.printf("%d ", (a = a + (b = a) 
                                                                System.out.println();
                                                  in.close();
                                                                                                                                                                 Line: 16 Col: 34
                                                                                                                                                                  Submit Code
                                                                                                                  Run Code
      ↑ Upload Code as File
       Test against custom input
                     You have earned 10.00 points!
                                                                                                                                                   92%
                                                                                                                                                                                                    48/50
                     You are now 2 points away from the 2nd star for
                     your java badge.
     Congratulations
                                                                                                                                              Next Challenge
     You solved this challenge. Would you like to
     challenge your friends?
⊘ Test case 0
                                                                          Compiler Message
                                                                              Success
Input (stdin)
                                                                                                                                                                 Download
0 2 10
                                                                                       5 3 5
Expected Output
                                                                                                                                                                 Download
                                                                                          2 6 14 30 62 126 254 510 1022
```

Exit Full Screen View

2. We use \_\_\_\_\_, and \_\_\_\_\_ to produce some series

2/13/23, 9:43 AM Java Loops I | HackerRank

```
Prepare > Java > Introduction > Java Loops I
                                                                                                                    Exit Full Screen View
   Objective
                                                                    Change Theme
                                                                                    Language Java 7
                                                                                                                              (0)
   In this challenge, we're going to use loops to help us do some
Problem
   simple math.
                                                                         import java.io.*;
                                                                         import java.math.*;
   Task
                                                                         import java.security.*;
   Given an integer, __, print its first _ multiples. Each multiple
                                                                         import java.text.*;
                                                                         import java.util.*;
                          ) should be printed on a new line in
         (where _
                                                                         import java.util.concurrent.*;
   the form: N \times i = result.
                                                                         import java.util.regex.*;
Submissions
   Input Format
   A single integer, _ .
                                                                         public class Solution {
                                                                             public static void main(String[] args) throws IOEx
   Constraints
                                                                                   BufferedReader bufferedReader = new BufferedRe
                                                                         (System.in));
                                                                                  int N = Integer.parseInt(bufferedReader.readL
   Output Format
                                                                                   for (int i=1; i<=10; i++){
Leaderboard
                                                                                  System.out.println(N + " x " + i + " = " + N*
   Print lines of output; each line (where ___
                                                    ) contains
                                                                                  bufferedReader.close();
   the
                      in the form:
   N \times i = result.
   Sample Input
      2
Discussions
                                                                                                                         Line: 17 Col: 58
   Sample Output
                                                                                                                          Submit Code

↑ Upload Code as File

                                                                                                         Run Code
      2 \times 1 = 2
                                                                   Test against custom input
      2 \times 2 = 4
      2 \times 3 = 6
      2 \times 4 = 8
      2 \times 5 = 10
      2 \times 6 = 12
                                                                         You have earned 10.00 points!
      2 \times 7 = 14
      2 \times 8 = 16
                                                                                                                                      38/50
                                                                         You are now 12 points away from the 2nd star for
      2 \times 9 = 18
                                                                        your java badge.
      2 \times 10 = 20
                                                                   Congratulations
                                                                                                                   Next Challenge
                                                                   You solved this challenge. Would you like to
                                                                   challenge your friends?
                                                                 ⊘ Test case 0
                                                                                           Compiler Message
                                                                                            Success
                                                                 Download
                                                                                           Input (stdin)
                                                                 Expected Output
                                                                                                                          Download
```