

Problem

Most HackerRank challenges require you to read input from `stdin` (standard input) and write output to `stdout` (standard output).

One popular way to read input from `stdin` is by using the `Scanner` class and specifying the Input Stream as `System.in`. For example:

```
Scanner scanner = new Scanner(System.in);
String myString = scanner.next();
int myInt = scanner.nextInt();
scanner.close();

System.out.println("myString is: " + myString);
System.out.println("myInt is: " + myInt);
```

The code above creates a `Scanner` object named *scanner* and uses it to read a `String` and an `int`. It then closes the `Scanner` object because there is no more input to read, and prints to `stdout` using `System.out.println(String)`. So, if our input is:

Hi 5

Our code will print:

myString is: Hi
myInt is: 5

Alternatively, you can use the [BufferedReader](#) class.

Task

In this challenge, you must read **3** integers from `stdin` and then print them to `stdout`. Each integer must be printed on a new line. To make the problem a little easier, a portion of the code is provided for you in the editor below.

Input Format

There are **3** lines of input, and each line contains a single integer.

Sample Input

Change Theme Language Java 7

```
1 import java.util.*;
2 public class Solution {
3     public static void main(String[] args) {
4         Scanner scan = new Scanner(System.in);
5         int a = scan.nextInt();
6         int b = scan.nextInt();
7         int c = scan.nextInt();
8         System.out.println(a);
9         System.out.println(b);
10        System.out.println(c);
11    }
12 }
```

Line: 12 Col: 2

[Upload Code as File](#) ☐ Test against custom input

[Run Code](#) [Submit Code](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

Most HackerRank challenges require you to read input from `stdin` (standard input) and write output to `stdout` (standard output).

One popular way to read input from `stdin` is by using the `Scanner` class and specifying the Input Stream as `System.in`. For example:

```
Scanner scanner = new Scanner(System.in);
String myString = scanner.next();
int myInt = scanner.nextInt();
scanner.close();

System.out.println("myString is: " + myString);
System.out.println("myInt is: " + myInt);
```

The code above creates a `Scanner` object named *scanner* and uses it to read a `String` and an `int`. It then closes the `Scanner` object because there is no more input to read, and prints to `stdout` using `System.out.println(String)`. So, if our input is:

```
Hi 5
```

Our code will print:

```
myString is: Hi
myInt is: 5
```

Alternatively, you can use the `BufferedReader` class.

Task

In this challenge, you must read **3** integers from `stdin` and then print them to `stdout`. Each integer must be printed on a new line. To make the problem a little easier, a portion of the code is provided for you in the editor below.

Input Format

There are **3** lines of input, and each line contains a single integer.

Sample Input

Line: 12 Col: 2

Upload Code as File

Test against custom input

Run Code Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin)		Download
1	42	
2	100	
3	125	
Your Output (stdout)		
1	42	
2	100	
3	125	
Expected Output		Download
1	42	

Problem

Submissions

Leaderboard

Discussions

Editorial

Most HackerRank challenges require you to read input from `stdin` (standard input) and write output to `stdout` (standard output).

One popular way to read input from `stdin` is by using the `Scanner` class and specifying the Input Stream as `System.in`. For example:

```
Scanner scanner = new Scanner(System.in);
String myString = scanner.next();
int myInt = scanner.nextInt();
scanner.close();

System.out.println("myString is: " + myString);
System.out.println("myInt is: " + myInt);
```

The code above creates a `Scanner` object named *scanner* and uses it to read a `String` and an `int`. It then closes the `Scanner` object because there is no more input to read, and prints to `stdout` using `System.out.println(String)`. So, if our input is:

```
Hi 5
```

Our code will print:

```
myString is: Hi
myInt is: 5
```

Alternatively, you can use the `BufferedReader` class.

Task

In this challenge, you must read **3** integers from `stdin` and then print them to `stdout`. Each integer must be printed on a new line. To make the problem a little easier, a portion of the code is provided for you in the editor below.

Input Format

There are **3** lines of input, and each line contains a single integer.

Sample Input

← → ↺ 🏠 🔒 [hackerrank.com/challenges/java-stdin-and-stdout-1/problem?isFullScreen=true&h_r=next-challenge&h_v=zen](#) ⌵ ⭐ 🏠 🔴 ⋮

HackerRank Prepare > Java > Introduction > Java Stdin and Stdout I Exit Full Screen View

Line: 12 Col: 2

⬆️ Upload Code as File

☐ Test against custom input

Run Code

Submit Code

🔥 java

You have earned 5.00 points!
You are now 17 points away from the 1st star for your java badge.

32%8/25

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

Next Challenge

✔️ Test case 0

✔️ Test case 1 [🔒](#)

✔️ Test case 2 [🔒](#)

Compiler Message

Success

Input (stdin) [Download](#)

142

2100

3125

Expected Output [Download](#)

142

2100

3125

🌤️ 24°C
Mostly sunny

🪟 🔍 Search 📁 📧 📁 🌐

ENG IN 🔊 🔍 8:48 AM
2/13/2023 🔴