

HackerRank password reset instr

Java Stdin and Stdout I | HackerR

Java Training - Google Drive

Java Stdin and Stdout II Discussio

hackerrank.com/challenges/java-stdin-and-stdout-1/problem?isFullScreen=true

HackerRank

Prepare > Java > Introduction > Java Stdin and Stdout I

Exit Full Screen View

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

Problem

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**  
42  
100  
125

**Sample Output**  
42  
100  
125

Change Theme

Language

Java 7

1 import java.util.\*;  
2  
3 public class Solution {  
4  
5 public static void main(String[] args) {  
6 Scanner scan = new Scanner(System.in);  
7 int a = scan.nextInt();  
8 // Complete this line  
9 // Complete this line  
10  
11 System.out.println(a);  
12 // Complete this line  
13 // Complete this line  
14  
15 int i = scan.nextInt();  
16 int j = scan.nextInt();  
17  
18 //String s = scan.nextLine();  
19  
20 //System.out.println("String: " + s);  
21 //System.out.println("Double: " + d);  
22 System.out.println(i);  
23 System.out.println(j);  
24  
25 scan.close();  
26 }  
27 }

Line: 27 Col: 23

Upload Code as File

Test against custom input

Run Code

Submit Code

Type here to search

Desktop

9:35 AM

2/13/2023

Problem

Submissions

Leaderboard

Discussions

Editorial

hackerank.com/challenges/java-stdin-and-stdout-1/problem?isFullScreen=true

HackerRank

Prepare > Java > Introduction > Java Stdin and Stdout I

Exit Full Screen View

```
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**

```
42
100
125
```

**Sample Output**

```
42
100
125
```

Line: 27 Col: 23

Upload Code as File

Test against custom input

Run Code

Submit Code

java

You have earned 5.00 points!

You are now 20 points away from the 1st star for your java badge.

20%

5/25

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 42

2 100

3 125

Expected Output

Download

1 42

2 100

3 125

Type here to search

Desktop

9:36 AM

2/13/2023

Welcome to the world of Java! In this challenge, we practice printing to stdout.

The code stubs in your editor declare a Solution class and a main method. Complete the main method by copying the two lines of code below and pasting them inside the body of your main method.

```
System.out.println("Hello, World.");
System.out.println("Hello, Java.");
```

### Input Format

There is no input for this challenge.

### Output Format

You must print two lines of output:

1. Print Hello, World. on the first line.
2. Print Hello, Java. on the second line.

### Sample Output

```
Hello, World.
Hello, Java.
```

Change Theme Language Java 7

```
1 public class Solution {
2
3     public static void main(String[] args) {
4         /* Enter your code here. Print output to STDOUT. Your class should be named Solution. */
5         System.out.println("Hello, World.");
6         System.out.println("Hello, Java.");
7     }
8 }
9
10
```

Line: 10 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

Problem

Submissions

Leaderboard

Discussions

Editorial

HackerRank

Prepare > Java > Introduction > Welcome to Java!

Exit Full Screen View

Welcome to the world of Java! In this challenge, we practice printing to stdout.

The code stubs in your editor declare a Solution class and a main method. Complete the main method by copying the two lines of code below and pasting them inside the body of your main method.

```
System.out.println("Hello, World.");
System.out.println("Hello, Java.");
```

Input Format

There is no input for this challenge.

Output Format

You must print two lines of output:

1. Print Hello, World. on the first line.

2. Print Hello, Java. on the second line.

Sample Output

```
Hello, World.
Hello, Java.
```

Line: 10 Col: 1

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

java

You have earned 3.00 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

Compiler Message

Success

Expected Output

Download

1

Hello, World.

2

Hello, Java.

Type here to search

Desktop

9:38 AM

2/13/2023

HackerRank password reset instr

Java If-Else | HackerRank

Java Training - Google Drive

HackerRank/Java If-Else.java at m

hackerrank.com/challenges/java-if-else/problem?isFullScreen=true

HackerRank

Prepare > Java > Introduction > Java If-Else

Exit Full Screen View

Problem

Submissions

Leaderboard

Discussions

Editorial

• 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

Sample Output 0

Weird

Sample Input 1

24

Sample Output 1

Not Weird

Explanation

Sample Case 0:  $n = 3$   
 $n$  is odd and odd numbers are weird, so we print Weird.  
Sample Case 1:  $n = 24$   
 $n > 20$  and  $n$  is even, so it isn't weird. Thus, we print Not Weird.

Change Theme

Language

Java 7

```
1 import java.io.*;
2 import java.math.*;
3 import java.security.*;
4 import java.text.*;
5 import java.util.*;
6 import java.util.concurrent.*;
7 import java.util.regex.*;
8
9 public class Solution {
10
11
12
13     private static final Scanner scanner = new Scanner(System.in);
14
15     public static void main(String[] args) {
16         int N = scanner.nextInt();
17         scanner.skip("(\\r\\n|\\n|\\u0020|\\u000B|\\u000A)?");
18
19         scanner.close();
20         if (N%2 != 0 )
21
22             System.out.println("Weird");
23
24
25         if (N%2 == 0 )
26
27             if(N>=2 && N<=5 || N>20)
28                 System.out.println("Not Weird");
29
30             else if(N>=6 && N<=20)
31                 System.out.println("Weird");
32
33
34
35
36
37
```

Line: 35 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

Type here to search

Desktop

9:40 AM

2/13/2023

ENG IN



A single line containing a positive integer,  $n$ .

**Output Format**

**Sample Input 0**

3

board Sample Output 0 You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [NEXT Challenge](#)

Learn More About Us [Weird](#)

Compiler Message

IONS

Dis...

**Explanation:** The correct answer is **1**. The patient is a 40-year-old female with a 10-year history of rheumatoid arthritis (RA) who is on chronic low-dose prednisone (5 mg daily). She presents with a 2-week history of lower extremity weakness and numbness, which is worse at night. She also reports a 1-month history of weight loss (10 lb) and decreased appetite. Her physical examination reveals bilateral lower extremity weakness and numbness, particularly in the feet. Her laboratory studies show a hemoglobin of 10 g/dL, a white blood cell count of 12,000/mm<sup>3</sup>, and a platelet count of 450,000/mm<sup>3</sup>. Her serum electrolytes, renal function, and liver function tests are within normal limits. Her serum vitamin D level is 10 ng/mL, and her serum parathyroid hormone-related protein (PTHrP) level is 150 pg/mL. Her serum ferritin level is 100 ng/mL, and her serum transferrin saturation is 20%. Her serum vitamin B12 level is 150 pg/mL, and her serum folate level is 10 ng/mL. Her serum creatine kinase (CK) level is 100 U/L, and her serum lactate dehydrogenase (LDH) level is 1,000 U/L. Her serum uric acid level is 6 mg/dL, and her serum glucose level is 100 mg/dL. Her serum protein level is 8 g/dL, and her serum albumin level is 3 g/dL. Her serum immunoglobulin G (IgG) level is 1,500 mg/dL, and her serum immunoglobulin A (IgA) level is 100 mg/dL. Her serum immunoglobulin M (IgM) level is 100 mg/dL, and her serum immunoglobulin D (IgD) level is 100 mg/dL. Her serum immunoglobulin E (IgE) level is 100 mg/dL, and her serum immunoglobulin G2 (IgG2) level is 100 mg/dL. Her serum immunoglobulin G1 (IgG1) level is 100 mg/dL, and her serum immunoglobulin G3 (IgG3) level is 100 mg/dL. Her serum immunoglobulin G4 (IgG4) level is 100 mg/dL, and her serum immunoglobulin G5 (IgG5) level is 100 mg/dL. Her serum immunoglobulin G6 (IgG6) level is 100 mg/dL, and her serum immunoglobulin G7 (IgG7) level is 100 mg/dL. Her serum immunoglobulin G8 (IgG8) level is 100 mg/dL, and her serum immunoglobulin G9 (IgG9) level is 100 mg/dL. Her serum immunoglobulin G10 (IgG10) level is 100 mg/dL, and her serum immunoglobulin G11 (IgG11) level is 100 mg/dL. Her serum immunoglobulin G12 (IgG12) level is 100 mg/dL, and her serum immunoglobulin G13 (IgG13) level is 100 mg/dL. Her serum immunoglobulin G14 (IgG14) level is 100 mg/dL, and her serum immunoglobulin G15 (IgG15) level is 100 mg/dL. Her serum immunoglobulin G16 (IgG16) level is 100 mg/dL, and her serum immunoglobulin G17 (IgG17) level is 100 mg/dL. Her serum immunoglobulin G18 (IgG18) level is 100 mg/dL, and her serum immunoglobulin G19 (IgG19) level is 100 mg/dL. Her serum immunoglobulin G20 (IgG20) level is 100 mg/dL, and her serum immunoglobulin G21 (IgG21) level is 100 mg/dL. Her serum immunoglobulin G22 (IgG22) level is 100 mg/dL, and her serum immunoglobulin G23 (IgG23) level is 100 mg/dL. Her serum immunoglobulin G24 (IgG24) level is 100 mg/dL, and her serum immunoglobulin G25 (IgG25) level is 100 mg/dL. Her serum immunoglobulin G26 (IgG26) level is 100 mg/dL, and her serum immunoglobulin G27 (IgG27) level is 100 mg/dL. Her serum immunoglobulin G28 (IgG28) level is 100 mg/dL, and her serum immunoglobulin G29 (IgG29) level is 100 mg/dL. Her serum immunoglobulin G30 (IgG30) level is 100 mg/dL, and her serum immunoglobulin G31 (IgG31) level is 100 mg/dL. Her serum immunoglobulin G32 (IgG32) level is 100 mg/dL, and her serum immunoglobulin G33 (IgG33) level is 100 mg/dL. Her serum immunoglobulin G34 (IgG34) level is 100 mg/dL, and her serum immunoglobulin G35 (IgG35) level is 100 mg/dL. Her serum immunoglobulin G36 (IgG36) level is 100 mg/dL, and her serum immunoglobulin G37 (IgG37) level is 100 mg/dL. Her serum immunoglobulin G38 (IgG38) level is 100 mg/dL, and her serum immunoglobulin G39 (IgG39) level is 100 mg/dL. Her serum immunoglobulin G40 (IgG40) level is 100 mg/dL, and her serum immunoglobulin G41 (IgG41) level is 100 mg/dL. Her serum immunoglobulin G42 (IgG42) level is 100 mg/dL, and her serum immunoglobulin G43 (IgG43) level is 100 mg/dL. Her serum immunoglobulin G44 (IgG44) level is 100 mg/dL, and her serum immunoglobulin G45 (IgG45) level is 100 mg/dL. Her serum immunoglobulin G46 (IgG46) level is 100 mg/dL, and her serum immunoglobulin G47 (IgG47) level is 100 mg/dL. Her serum immunoglobulin G48 (IgG48) level is 100 mg/dL, and her serum immunoglobulin G49 (IgG49) level is 100 mg/dL. Her serum immunoglobulin G50 (IgG50) level is 100 mg/dL, and her serum immunoglobulin G51 (IgG51) level is 100 mg/dL. Her serum immunoglobulin G52 (IgG52) level is 100 mg/dL, and her serum immunoglobulin G53 (IgG53) level is 100 mg/dL. Her serum immunoglobulin G54 (IgG54) level is 100 mg/dL, and her serum immunoglobulin G55 (IgG55) level is 100 mg/dL. Her serum immunoglobulin G56 (IgG56) level is 100 mg/dL, and her serum immunoglobulin G57 (IgG57) level is 100 mg/dL. Her serum immunoglobulin G58 (IgG58) level is 100 mg/dL, and her serum immunoglobulin G59 (IgG59) level is 100 mg/dL. Her serum immunoglobulin G60 (IgG60) level is 100 mg/dL, and her serum immunoglobulin G61 (IgG61) level is 100 mg/dL. Her serum immunoglobulin G62 (IgG62) level is 100 mg/dL, and her serum immunoglobulin G63 (IgG63) level is 100 mg/dL. Her serum immunoglobulin G64 (IgG64) level is 100 mg/dL, and her serum immunoglobulin G65 (IgG65) level is 100 mg/dL. Her serum immunoglobulin G66 (IgG66) level is 100 mg/dL, and her serum immunoglobulin G67 (IgG67) level is 100 mg/dL. Her serum immunoglobulin G68 (IgG68) level is 100 mg/dL, and her serum immunoglobulin G69 (IgG69) level is 100 mg/dL. Her serum immunoglobulin G70 (IgG70) level is 100 mg/dL, and her serum immunoglobulin G71 (IgG71) level is 100 mg/dL. Her serum immunoglobulin G72 (IgG72) level is 100 mg/dL, and her serum immunoglobulin G73 (IgG73) level is 100 mg/dL. Her serum immunoglobulin G74 (IgG74) level is 100 mg/dL, and her serum immunoglobulin G75 (IgG75) level is 100 mg/dL. Her serum immunoglobulin G76 (IgG76) level is 100 mg/dL, and her serum immunoglobulin G77 (IgG77) level is 100 mg/dL. Her serum immunoglobulin G78 (IgG78) level is 100 mg/dL, and her serum immunoglobulin G79 (IgG79) level is 100 mg/dL. Her serum immunoglobulin G80 (IgG80) level is 100 mg/dL, and her serum immunoglobulin G81 (IgG81) level is 100 mg/dL. Her serum immunoglobulin G82 (IgG82) level is 100 mg/dL, and her serum immunoglobulin G83 (IgG83) level is 100 mg/dL. Her serum immunoglobulin G84 (IgG84) level is 100 mg/dL, and her serum immunoglobulin G85 (IgG85) level is 100 mg/dL. Her serum immunoglobulin G86 (IgG86) level is 100 mg/dL, and her serum immunoglobulin G87 (IgG87) level is 100 mg/dL. Her serum immunoglobulin G88 (IgG88) level is 100 mg/dL, and her serum immunoglobulin G89 (IgG89) level is 100 mg/dL. Her serum immunoglobulin G90 (IgG90) level is 100 mg/dL, and her serum immunoglobulin G91 (IgG91) level is 100 mg/dL. Her serum immunoglobulin G92 (IgG92) level is 100 mg/dL, and her serum immunoglobulin G93 (IgG93) level is 100 mg/dL. Her serum immunoglobulin G94 (IgG94) level is 100 mg/dL, and her serum immunoglobulin G95 (IgG95) level is 100 mg/dL. Her serum immunoglobulin G96 (IgG96) level is 100 mg/dL, and her serum immunoglobulin G97 (IgG97) level is 100 mg/dL. Her serum immunoglobulin G98 (IgG98) level is 100 mg/dL, and her serum immunoglobulin G99 (IgG99) level is 100 mg/dL. Her serum immunoglobulin G100 (IgG100) level is 100 mg/dL, and her serum immunoglobulin G101 (IgG101) level is 100 mg/dL. Her serum immunoglobulin G102 (IgG102) level is 100 mg/dL, and her serum immunoglobulin G103 (IgG103) level is 100 mg/dL. Her serum immunoglobulin G104 (IgG104) level is 100 mg/dL, and her serum immunoglobulin G105 (IgG105) level is 100 mg/dL. Her serum immunoglobulin G106 (IgG106) level is 100 mg/dL, and her serum immunoglobulin G107 (IgG107) level is 100 mg/dL. Her serum immunoglobulin G108 (IgG108) level is 100 mg/dL, and her serum immunoglobulin G109 (IgG109) level is 100 mg/dL. Her serum immunoglobulin G110 (IgG110) level is 100 mg/dL, and her serum immunoglobulin G111 (IgG111) level is 100 mg/dL. Her serum immunoglobulin G112 (IgG112) level is 100 mg/dL, and her serum immunoglobulin G113 (IgG113) level is 100 mg/dL. Her serum immunoglobulin G114 (IgG114) level is 100 mg/dL, and her serum immunoglobulin G115 (IgG115) level is 100 mg/dL. Her serum immunoglobulin G116 (IgG116) level is 100 mg/dL, and her serum immunoglobulin G117 (IgG117) level is 100 mg/dL. Her serum immunoglobulin G118 (IgG118) level is 100 mg/dL, and her serum immunoglobulin G119 (IgG119) level is 100 mg/dL. Her serum immunoglobulin G120 (IgG120) level is 100 mg/dL, and her serum immunoglobulin G121 (IgG121) level is 100 mg/dL. Her serum immunoglobulin G122 (IgG122) level is 100 mg/dL, and her serum immunoglobulin G123 (IgG123) level is 100 mg/dL. Her serum immunoglobulin G124 (IgG124) level is 100 mg/dL, and her serum immunoglobulin G125 (IgG125) level is 100 mg/dL. Her serum immunoglobulin G126 (IgG126) level is 100 mg/dL, and her serum immunoglobulin G127 (IgG127) level is 100 mg/dL. Her serum immunoglobulin G128 (IgG128) level is 100 mg/dL, and her serum immunoglobulin G129 (IgG129) level is 100 mg/dL. Her serum immunoglobulin G130 (IgG130) level is 100 mg/dL, and her serum immunoglobulin G131 (IgG131) level is 100 mg/dL. Her serum immunoglobulin G132 (IgG132) level is 100 mg/dL, and her serum immunoglobulin G133 (IgG133) level is 100 mg/dL. Her serum immunoglobulin G134 (IgG134) level is 100 mg/dL, and her serum immunoglobulin G135 (IgG135) level is 100 mg/dL. Her serum immunoglobulin G136 (IgG136) level is 100 mg/dL, and her serum immunoglobulin G137 (IgG137) level is 100 mg/dL. Her serum immunoglobulin G138 (IgG138) level is 100 mg/dL, and her serum immunoglobulin G139 (IgG139) level is 100 mg/dL. Her serum immunoglobulin G140 (IgG140) level is 100 mg/dL, and her serum immunoglobulin G141 (IgG141) level is 100 mg/dL. Her serum immunoglobulin G142 (IgG142) level is 100 mg/dL, and her serum immunoglobulin G143 (IgG143) level is 100 mg/dL. Her serum immunoglobulin G144 (IgG144)

Sample Case 0:  $n = 3$  1 Weird

Sample Case 1:  $n = 24$

type here to search Desktop 2/13/2023

In this challenge, you must read an integer, a double, and a String from stdin, then print the values according to the instructions in the Output Format section below. To make the problem a little easier, a portion of the code is provided for you in the editor.

**Note:** We recommend completing [Java Stdin and Stdout I](#) before attempting this challenge.

### Input Format

There are three lines of input:

1. The first line contains an integer.
2. The second line contains a double.
3. The third line contains a String.

### Output Format

There are three lines of output:

1. On the first line, print `String:` followed by the unaltered String read from stdin.
2. On the second line, print `Double:` followed by the unaltered double read from stdin.
3. On the third line, print `Int:` followed by the unaltered integer read from stdin.

To make the problem easier, a portion of the code is already provided in the editor.

**Note:** If you use the `nextLine()` method immediately following the `nextInt()` method, recall that `nextInt()` reads integer tokens; because of this, the last newline character for that line of integer input is still queued in the input buffer and the next `nextLine()` will be reading the remainder of the integer line (which is empty).

### Sample Input

```
42
3.1415
Welcome to HackerRank's Java tutorials!
```

### Sample Output

```
String: Welcome to HackerRank's Java tutorials!
Double: 3.1415
Int: 42
```

Change Theme Language Java 7

```
1 import java.util.Scanner;
2
3 public class Solution {
4
5     public static void main(String[] args) {
6         Scanner scan = new Scanner(System.in);
7         int i = scan.nextInt();
8
9         // Write your code here.
10        double d = scan.nextDouble();
11        scan.nextLine();
12        String s = scan.nextLine();
13        System.out.println("String: " + s);
14        System.out.println("Double: " + d);
15        System.out.println("Int: " + i);
16    }
17 }
```

Line: 17 Col: 2

Upload Code as File

Test against custom input

Run Code

Submit Code

In this challenge, you must read an integer, a double, and a String from stdin, then print the values according to the instructions in the Output Format section below. To make the problem a little easier, a portion of the code is provided for you in the editor.

**Note:** We recommend completing [Java Stdin and Stdout I](#) before attempting this challenge.

### Input Format

There are three lines of input:

1. The first line contains an integer.
2. The second line contains a double.
3. The third line contains a String.

### Output Format

There are three lines of output:

1. On the first line, print `String`: followed by the unaltered String read from stdin.
2. On the second line, print `Double`: followed by the unaltered double read from stdin.
3. On the third line, print `Int`: followed by the unaltered integer read from stdin.

To make the problem easier, a portion of the code is already provided in the editor.

**Note:** If you use the `nextLine()` method immediately following the `nextInt()` method, recall that `nextInt()` reads integer tokens; because of this, the last newline character for that line of integer input is still queued in the input buffer and the next `nextLine()` will be reading the remainder of the integer line (which is empty).

### Sample Input

```
42
3.1415
Welcome to HackerRank's Java tutorials!
```

### Sample Output

```
String: Welcome to HackerRank's Java tutorials!
Double: 3.1415
Int: 42
```

Line: 17 Col: 2

Upload Code as File

Test against custom input

Run Code

Submit Code

Test case 0

Compiler Message

Test case 1

Success

Test case 2

Input (stdin)

Download

Test case 3

```
1 42
2 3.1415
3 Welcome to HackerRank's Java tutorials!
```

Test case 4

Expected Output

Download

```
1 String: Welcome to HackerRank's Java tutorials!
2 Double: 3.1415
3 Int: 42
```



Java's System.out.printf function can be used to print formatted output. The purpose of this exercise is to test your understanding of formatting output using printf.

To get you started, a portion of the solution is provided for you in the editor; you must format and print the input to complete the solution.

### Input Format

Every line of input will contain a String followed by an integer.

Each String will have a maximum of 10 alphabetic characters, and each integer will be in the inclusive range from 0 to 999.

### Output Format

In each line of output there should be two columns:

The first column contains the String and is left justified using exactly 15 characters.

The second column contains the integer, expressed in exactly 3 digits; if the original input has less than three digits, you must pad your output's leading digits with zeroes.

### Sample Input

```
java 100
cpp 65
python 50
```

### Sample Output

```
=====
java      100
cpp       065
python    050
=====
```

### Explanation

Each String is left-justified with trailing whitespace through the first 15 characters. The leading digit of the integer is the 16<sup>th</sup> character, and each integer that was less than 3 digits now has leading zeroes.

Change Theme Language Java 7

```
1 import java.util.Scanner;
2
3 public class Solution {
4
5     public static void main(String[] args) {
6         Scanner sc=new Scanner(System.in);
7         System.out.println("=====");
8         for(int i=0;i<3;i++)
9         {
10             String s1=sc.next();
11             int x=sc.nextInt();
12             System.out.printf("%-15s%03d\n",s1,x);
13             //Complete this line
14         }
15         System.out.println("=====");
16     }
17 }
18
19
20
21
22
```

Line: 12 Col: 55

Upload Code as File

Test against custom input

Run Code

Submit Code

Problem

Submissions

Leaderboard

Discussions

Editorial

HackerRank

Prepare > Java > Introduction > Java Output Formatting

Exit Full Screen View

Java's System.out.printf function can be used to print formatted output. The purpose of this exercise is to test your understanding of formatting output using printf.

To get you started, a portion of the solution is provided for you in the editor; you must format and print the input to complete the solution.

**Input Format**

Every line of input will contain a String followed by an integer.

Each String will have a maximum of 10 alphabetic characters, and each integer will be in the inclusive range from 0 to 999.

**Output Format**

In each line of output there should be two columns:

The first column contains the String and is left justified using exactly 15 characters.

The second column contains the integer, expressed in exactly 3 digits; if the original input has less than three digits, you must pad your output's leading digits with zeroes.

**Sample Input**

```
java 100
cpp 65
python 50
```

**Sample Output**

```
=====
java      100
cpp       065
python    050
=====
```

**Explanation**

Each String is left-justified with trailing whitespace through the first 15 characters. The leading digit of the integer is the 16<sup>th</sup> character, and each integer that was less than 3 digits now has leading zeroes.


Line: 12 Col: 55

Upload Code as File

Test against custom input

Run Code

Submit Code



You have earned 10.00 points!

You are now 12 points away from the 2nd star for your java badge.

52%

38/50

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

Test case 3

Input (stdin)

Download

1 java 100

2 cpp 65

3 python 50

Expected Output

Download

1 =====


2 java 100

3 cpp 065

4 python 050

5 =====

Type here to search



Desktop

ENG IN

9:45 AM

2/13/2023

## 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:  $N \times i = \text{result}$ .

## 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:

$N \times i = \text{result}$ .

## Sample Input

2

## Sample 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

Change Theme Language Java 7

```
1 import java.io.*;
2 import java.math.*;
3 import java.security.*;
4 import java.text.*;
5 import java.util.*;
6 import java.util.concurrent.*;
7 import java.util.regex.*;
8
9
10
11 public class Solution {
12     public static void main(String[] args) throws IOException {
13
14         Scanner in = new Scanner(System.in);
15         int N = in.nextInt();
16         for(int i = 1; i <= 10; i++)
17         {
18             System.out.println(N + " x " + i + " = " + i * N);
19         }
20
21
22
23
24 }
25
```

Line: 22 Col: 8

Upload Code as File Test against custom input

Run Code Submit Code

## 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:  $N \times i = \text{result}$ .

## 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:  $N \times i = \text{result}$ .

## Sample Input

2

## Sample 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
```

Line: 22 Col: 8

Upload Code as File

Test against custom input

Run Code

Submit Code



You have earned 10.00 points!

You are now 2 points away from the 2nd star for your java badge.

92%

48/50

## Congratulations

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

Next Challenge

### Test case 0

Compiler Message

Success

### Test case 1

### Test case 2

Input (stdin)

1 2

Download

Expected Output

```
1 2 x 1 = 2
2 2 x 2 = 4
3 2 x 3 = 6
4 2 x 4 = 8
5 2 x 5 = 10
```

Download

HackerRank password reset instr

Java Loops II | HackerRank

Java Training - Google Drive

HackerRank/Solution.java at mas

hackerrank.com/challenges/java-loops/problem?isFullScreen=true

HackerRank

Prepare > Java > Introduction > Java Loops II

Exit Full Screen View

•  $1 \leq n \leq 15$

**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**  

```
2
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:  
1. We use  $a = 0$ ,  $b = 2$ , and  $n = 10$  to produce some series  $s_0, s_1, \dots, s_{n-1}$ :

- $s_0 = 0 + 1 \cdot 2 = 2$
- $s_1 = 0 + 1 \cdot 2 + 2 \cdot 2 = 6$
- $s_2 = 0 + 1 \cdot 2 + 2 \cdot 2 + 4 \cdot 2 = 14$

... and so on.  
Once we hit  $n = 10$ , we print the first ten terms as a single line of space-separated integers.  
2. We use  $a = 5$ ,  $b = 3$ , and  $n = 5$  to produce some series  $s_0, s_1, \dots, s_{n-1}$ :

- $s_0 = 5 + 1 \cdot 3 = 8$
- $s_1 = 5 + 1 \cdot 3 + 2 \cdot 3 = 14$
- $s_2 = 5 + 1 \cdot 3 + 2 \cdot 3 + 4 \cdot 3 = 26$
- $s_3 = 5 + 1 \cdot 3 + 2 \cdot 3 + 4 \cdot 3 + 8 \cdot 3 = 50$
- $s_4 = 5 + 1 \cdot 3 + 2 \cdot 3 + 4 \cdot 3 + 8 \cdot 3 + 16 \cdot 3 = 98$

We then print each element of our series as a single line of space-separated values.

Change Theme Language Java 7

```
1 import java.util.*;
2 import java.io.*;
3
4 class Solution{
5     public static void main(String []argh){
6         Scanner in = new Scanner(System.in);
7         int t=in.nextInt();
8         for(int i=0;i<t;i++){
9             int a = in.nextInt();
10            int b = in.nextInt();
11            int n = in.nextInt();
12            int calc = a;
13            for(int j=0;j<n;j++){
14                calc+=(Math.pow(2,j)*b);
15                System.out.print(calc+" ");
16            }
17            System.out.println();
18        }
19        in.close();
20    }
21 }
22
23
```

Line: 18 Col: 34

Upload Code as File

Test against custom input

Run Code

Submit Code

Problem

Submissions

Leaderboard

Discussions

Editorial

Type here to search

Desktop

9:49 AM

2/13/2023



Problem

Submissions

Leaderboard

Discussions

Editorial

HackerRank

Prepare > Java > Introduction > Java Loops II

Exit Full Screen View

•  $1 \leq n \leq 15$

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

```
2
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:

1. We use  $a = 0$ ,  $b = 2$ , and  $n = 10$  to produce some series  $s_0, s_1, \dots, s_{n-1}$ :

- $s_0 = 0 + 1 \cdot 2 = 2$
- $s_1 = 0 + 1 \cdot 2 + 2 \cdot 2 = 6$
- $s_2 = 0 + 1 \cdot 2 + 2 \cdot 2 + 4 \cdot 2 = 14$

... and so on.

Once we hit  $n = 10$ , we print the first ten terms of a single line of space-separated integers.

2. We use  $a = 5$ ,  $b = 3$ , and  $n = 5$  to produce some series  $s_0, s_1, \dots, s_{n-1}$ :

- $s_0 = 5 + 1 \cdot 3 = 8$
- $s_1 = 5 + 1 \cdot 3 + 2 \cdot 3 = 14$
- $s_2 = 5 + 1 \cdot 3 + 2 \cdot 3 + 4 \cdot 3 = 26$
- $s_3 = 5 + 1 \cdot 3 + 2 \cdot 3 + 4 \cdot 3 + 8 \cdot 3 = 50$
- $s_4 = 5 + 1 \cdot 3 + 2 \cdot 3 + 4 \cdot 3 + 8 \cdot 3 + 16 \cdot 3 = 98$

We then print each element of our series as a single line of space-separated values.

Line: 18 Col: 34

Upload Code as File

Test against custom input

Run Code

Submit Code

You have earned 10.00 points!

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

27%

58/80

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

Test case 3

Test case 4

Compiler Message

Success

Input (stdin)

1	2
2	0 2 10
3	5 3 5

Expected Output

1	2 6 14 30 62 126 254 510 1022 2046
2	8 14 26 50 98

Type here to search

Desktop

9:49 AM

2/13/2023

HackerRank password reset instr

Java Datatypes | HackerRank

Java Training - Google Drive

HackerRank/JavaDatatypes.java

hackerrank.com/challenges/java-datatypes/problem?isFullScreen=true

HackerRank

Prepare > Java > Introduction > Java Datatypes

Exit Full Screen View

Problem

Submissions

Leaderboard

Discussions

Editorial

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:** <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

**Input Format**

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

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

**Output Format**

For each input variable *n* and appropriate primitive *dataType*, 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.: *byte < short < int < long*).

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
```

Change Theme

Language

Java 7

```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class JavaDatatypes {
8
9     static String whoCanFitTheNumber(String numString)
10    {
11        String answer = "";
12        try{
13            long num = Long.parseLong(numString);
14            answer = numString + " can be fitted in:\n";
15            if((num<=Byte.MAX_VALUE) && (num>=Byte.MIN_VALUE)){
16                answer = answer.concat("* byte\n* short\n* int\n* long");
17            }else if((num <= Short.MAX_VALUE) && (num >= Short.MIN_VALUE)){
18                answer = answer.concat("* short\n* int\n* long");
19            }else if((num <= Integer.MAX_VALUE) && (num >= Integer.MIN_VALUE)){
20                answer = answer.concat("* int\n* long");
21            }else{
22                answer = answer.concat("* long");
23            }
24        }catch (NumberFormatException e){
25            answer = numString+" can't be fitted anywhere.";
26        }
27        return answer;
28    }
29    public static void main(String[] args) {
30        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be
31        named Solution.
32        */
33        Scanner scanner = new Scanner(System.in);
34        int numTestCases = scanner.nextInt() ;
35        scanner.nextLine();
36        for(int i=0; i<numTestCases;i++){
37            String numString = scanner.nextLine();
```

Upload Code as File

Test against custom input

Run Code

Submit Code

Line: 41 Col: 1

Waiting for www.hackerrank.com...

Type here to search

Desktop

ENG 9:52 AM 2/13/2023

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:** <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

### Input Format

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

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

### Output Format

For each input variable ***n*** and appropriate primitive ***dataType***, 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.: *byte < short < int < long*).

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  
2133333333333333333333333333333333
```

42  
43  
44

Line: 41 Col: 1

 Upload Code as File

☐ Test against custom input

Run Code

**Submit Code**



You have earned 10.00 points!

You are now 12 points away from the 3rd star for your java badge.

60%

68/80

## Congratulations

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

### Next Challenge

✔ Test case 0

### Compiler Message

✓ Test case 1

## Success

✓ Test case 2 

Input (stdin)

✔ Test case 3 

[illegible]

### Expected Output

Download

Problem

Submissions

Leaderboard

Discussions

Editorial

HackerRank

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

Exit Full Screen View

"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  $n$  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.
```

Change Theme Language Java 7

```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be
           named Solution. */
11         Scanner sc = new Scanner(System.in);
12         int lineNumber = 0;
13         while(sc.hasNext()){
14             lineNumber++;
15             System.out.println(lineNumber+" "+sc.nextLine());
16         }
17     }
18 }
```

Line: 18 Col: 2

Upload Code as File

Test against custom input

Run Code

Submit Code

Type here to search

Desktop

9:57 AM

2/13/2023



Problem

Submissions

Leaderboard

Discussions

Editorial

HackerRank

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

Exit Full Screen View

"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  $n$  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.

Line: 18 Col: 2

Upload Code as File ☐ Test against custom input

Run Code Submit Code

You have earned 10.00 points!

You are now 2 points away from the 3rd star for your java badge.

93% 78/80

Congratulations

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

Next Challenge

Test case 0

Test case 1

Compiler Message

Success

Input (stdin)

Download

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

Expected Output

Download

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

Type here to search

Desktop

9:57 AM 2/13/2023



US: formattedPayment  
India: formattedPayment  
China: formattedPayment  
France: formattedPayment

where *formattedPayment* is *payment* formatted according to the appropriate *Locale*'s currency.

**Note:** India does not have a built-in *Locale*, so you must construct one where the language is *en* (i.e., English).

### Input Format

A single double-precision number denoting *payment*.

### Constraints

- $0 \leq \text{payment} \leq 10^9$

### Output Format

On the first line, print US: *u* where *u* is *payment* formatted for US currency.

On the second line, print India: *i* where *i* is *payment* formatted for Indian currency.

On the third line, print China: *c* where *c* is *payment* formatted for Chinese currency.

On the fourth line, print France: *f*, where *f* is *payment* formatted for French currency.

### Sample Input

12324.134

### Sample Output

US: \$12,324.13  
India: Rs.12,324.13  
China: ¥12,324.13  
France: 12 324,13 €

### Explanation

Each line contains the value of *payment* formatted according to the four countries' respective currencies.

Change Theme Language Java 7

```
1 import java.util.*;
2 import java.text.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         double payment = scanner.nextDouble();
9         scanner.close();
10        NumberFormat n = NumberFormat.getCurrencyInstance(Locale.US);
11        String us = n.format(payment);
12        NumberFormat n1 = NumberFormat.getCurrencyInstance(new Locale("en", "IN"));
13        String india = n1.format(payment);
14        NumberFormat n2 = NumberFormat.getCurrencyInstance(Locale.CHINA);
15        String china = n2.format(payment);
16        NumberFormat n3 = NumberFormat.getCurrencyInstance(Locale.FRANCE);
17        String france = n3.format(payment);
18
19        // Write your code here.
20
21        System.out.println("US: " + us);
22        System.out.println("India: " + india);
23        System.out.println("China: " + china);
24        System.out.println("France: " + france);
25    }
26 }
27
```

Line: 27 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

Desktop ENG 1:33 PM  
IN 2/13/2023