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

HackerRank Prepare > Java > Introduction > Java Currency Formatter

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | Editorial

Given a double-precision number, $payment$, denoting an amount of money, use the NumberFormat class' getCurrencyInstance method to convert $payment$ into the US, Indian, Chinese, and French currency formats. Then print the formatted values as follows:

```
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

- $0 \leq payment \leq 10^9$

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 €
```

Change Theme Language: Java 7

```
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) {
        Scanner scanner = new Scanner(System.in);
        double payment = scanner.nextDouble();
        scanner.close();
        // Write your code here.
        String us = NumberFormat.getCurrencyInstance(Locale.US).format(payment);
         String india = NumberFormat.getCurrencyInstance(new Locale("en","in")).format(payment);
          String china = NumberFormat.getCurrencyInstance(Locale.CHINA).format(payment);
           String france = NumberFormat.getCurrencyInstance(Locale.FRANCE).format(payment);

        System.out.println("US: " + us);
        System.out.println("India: " + india);
        System.out.println("China: " + china);
        System.out.println("France: " + france);
    }
}
```

Line: 23 Col: 2

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

Search ENG IN 2:06 PM 2/13/2023

Given a double-precision number, $payment$, denoting an amount of money, use the NumberFormat class' getCurrencyInstance method to convert $payment$ into the US, Indian, Chinese, and French currency formats. Then print the formatted values as follows:

```
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

- $0 \leq payment \leq 10^9$

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 €
```

Line: 23 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

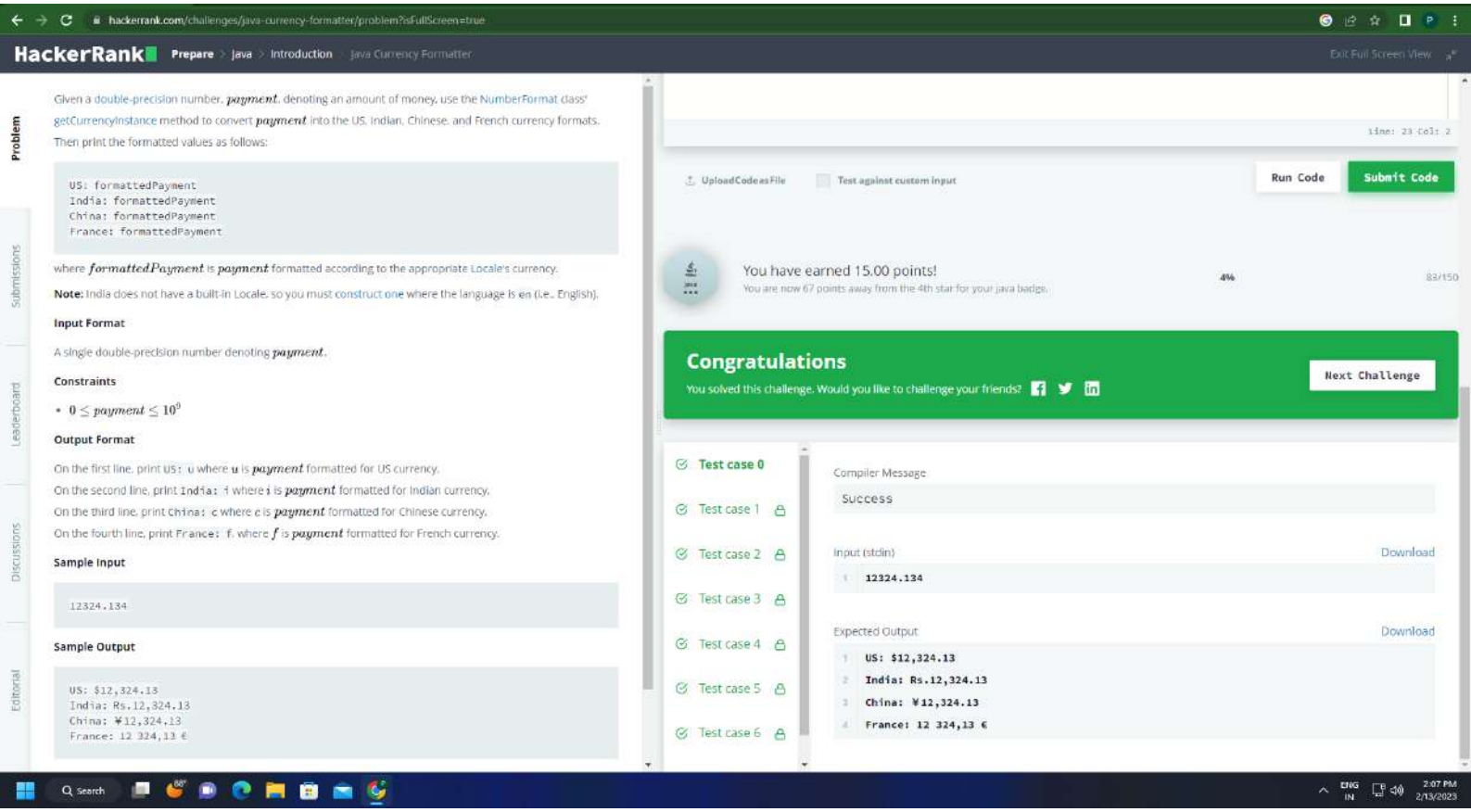
```
12324.134
```

Your Output (stdout)

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

Expected Output Download

```
US: $12,324.13
India: Rs.12,324.13
China: ¥12,324.13
```


hackerrank.com/challenges/java-currency-formatter/problem?isFullScreen=true
HackerRank
Prepare > Java > Introduction > Java Currency Formatter
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
Editorial
Given a double-precision number, payment, denoting an amount of money, use the NumberFormat class' getCurrencyInstance method to convert payment into the US, Indian, Chinese, and French currency formats. Then print the formatted values as follows:
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
$0 \leq payment \leq 10^9$
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 €
Upload Code as File
Test against custom input
Run Code
Submit Code
You have earned 15.00 points!
You are now 67 points away from the 4th star for your java badge.
4%
83/150
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
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
12324.134
Expected Output
Download
US: $12,324.13
India: Rs.12,324.13
China: ¥12,324.13
France: 12 324,13 €
Search
ENG
IN
2:07 PM
2/13/2023