**Java Currency Formatter**

<https://www.hackerrank.com/challenges/java-currency-formatter/problem>

Given a [double-precision](https://en.wikipedia.org/wiki/Double-precision_floating-point_format) number, payment, denoting an amount of money, use the [NumberFormat](https://docs.oracle.com/javase/8/docs/api/java/text/NumberFormat.html) class' [getCurrencyInstance](https://docs.oracle.com/javase/8/docs/api/java/text/NumberFormat.html" \l "getCurrencyInstance-java.util.Locale-) 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 formatted according to the appropriate [Locale](https://docs.oracle.com/javase/8/docs/api/java/util/Locale.html)'s currency.

**Note:** India does not have a built-in Locale, so you must [construct one](https://docs.oracle.com/javase/8/docs/api/java/util/Locale.html#Locale-java.lang.String-java.lang.String-) where the language is en (i.e., English).

**Input Format**

A single double-precision number denoting payment.

**Constraints**

* 0 <= payment <= 109

**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 u is payment formatted for Indian currency.  
On the third line, print China: c where u is payment formatted for Chinese currency.  
On the fourth line, print France: f, where u 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: 12324,13 €

**Explanation**

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