Description

Now your program knows how to get up-to-date rates. Let's make it more interactive! In this stage, you need to read from the input the currency you have, the currency you want to exchange your money for, and the amount of money you want to exchange. Mind that the input number can have a fractional part!

Keep in mind that the currency you have stays the same, you read it only once in the beginning, then you only need to read the currency you want to exchange your money for and the amount of money since they change. If you come across an empty input, the options for exchange are over.

In product development, the performance is key. Let's use a simple way to speed up the program, called **caching**. What if you need to do the math for the same exchange target several times? Isn't it better to save rates at runtime, instead of wasting resources on retrieving the same data from the Internet? If you already did calculations for this exchange target before, you know the rate, so there is no need to connect to the Internet, you only need to refer to the data in cache. You can organize the cache anyway you like, but the easiest way would be to use (as you probably already did in stage 5) a dictionary with currencies as keys and rates as values.

Objectives

Let's go over the steps one more time:

1. Take the first input – the currency that you have. It is default for all the calculations.
2. Retrieve the data from FloatRates as before.
3. Save the exchange rates for USD and EUR (these are the most popular ones, so it's good to have rates for them in advance).
4. Take the second input – the currency code that you want to exchange money for, and the third input – amount of money you have.
5. Check the cache. Maybe you already have what you need?
6. If you have the currency in your cache, calculate the result.
7. If not, get it from the site, and calculate the result.
8. Save the rate to your cache.

9. Print the result.

10. Repeat steps 4-9 until there is no currency left to process.

Examples

The greater-than symbol followed by a space (`> `) represents the user input. Note that it's not part of the input.

Example 1:

```
> ILS
> USD
> 45
Checking the cache...
Oh! It is in the cache!
You received 13.84 USD.
> CHF
> 57
Checking the cache...
Sorry, but it is not in the cache!
You received 13.14 CHF.
> EUR
> 33
Checking the cache...
Oh! It is in the cache!
You received 8.38 EUR.
```

Example 2:

```
> USD
> EUR
> 20
Checking the cache...
Oh! It is in the cache!
You received 16.52 EUR.
> NOK
> 45
Checking the cache...
Sorry, but it is not in the cache!
You received 382.1 NOK.
> SEK
> 75
Checking the cache...
Sorry, but it is not in the cache!
You received 624.66 SEK.
> NOK
> 55
Checking the cache...
```

```
Oh! It is in the cache!
You received 467.02 NOK.
> ISK
> 91
Checking the cache...
Sorry, but it is not in the cache!
You received 11708.38 ISK.
```