

PROBLEM DESCRIPTION

Given a matrix of currency pairs' exchange rates, build an algorithm in Python 2.7.10 that either finds a profitable risk-free trading strategy and presents it, or prints "no risk-free opportunities exist" in case there are no profitable risk-free currency trading strategy.

EXAMPLE

Imagine €1.00 in European currency buys \$1.10 of U.S. currency, and \$1.00 of U.S. currency buys ¥6.00 of Chinese currency, and ¥1.00 of Chinese currency buys €0.16 of European currency. Therefore, a profitable risk-free trading strategy would be the following:

- (1) take €1.00 and purchase \$1.10, then
- (2) take \$1.10 and purchase ¥6.60 (1.10×6.00), then
- (3) take ¥6.60 and purchase €1.056 (6.60×0.16)

Therefore, the strategy started with €1.00 and ended with €1.056, generating 5.60% risk-free profit:

$$1 \times 1.1 \times 6 \times 0.16 = 1.056$$

The task is to determine whether any sequence of currency exchanges can yield a risk-free profit starting and ending with the same currency as per the example above.

INPUTS

The input is the .csv file containing a matrix of n currency exchange rates similar to:

	EUR	USD	CNY	RUB
EUR	1.00	1.10	6.50	70.00
USD	0.90	1.00	6.00	65.00
CNY	0.16	0.17	1.00	10.75
RUB	0.01	0.02	0.08	1.00

Thus the first row and the first column of the table represent the names of the currencies. The corresponding intersections represent the conversion rates between the two currencies.

Each table consists of n+1 rows and n+1 columns in the input file.

OUTPUTS

The algorithm needs to determine whether a sequence of exchanges exists that yields a profit greater than 1.00%. If there is more than one sequence yielding a profit of more than 1.00%, the algorithm needs to print the shortest sequence.

If no sequence yielding a profit of more than 1.00% exists, the algorithm needs to print: "no risk-free opportunities exist yielding over 1.00% profit exist".

All profiting sequences must consist of n or fewer transactions where n is the dimension of the table giving conversion rates. The sequence €, \$, € represents two conversions.

EXAMPLES

Input Example 1:

Input Example 2:

	<u>Curr.1</u>	<u>Curr.2</u>	<u>Curr.3</u>
<u>Curr.1</u>	1.00	1.20	0.89
<u>Curr.2</u>	0.88	1.00	5.10
<u>Curr.3</u>	1.10	0.15	1.00

Output Example 1:

“Curr.1, Curr.2, Curr.1”

	<u>Curr.1</u>	<u>Curr.2</u>	<u>Curr.3</u>	<u>Curr.4</u>
<u>Curr.1</u>	1.00	3.1	0.0023	0.35
<u>Curr.2</u>	0.21	1.00	0.00353	8.13
<u>Curr.3</u>	200	180.559	1.00	10.339
<u>Curr.4</u>	2.11	0.089	0.06111	1.00

Output Example 2:

“Curr.1, Curr.2, Curr.4, Curr.1”