ARBITRAG - Arbitrage

no tags

Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 US Dollar buys 0.5 British pounds, 1 British pound buys 10.0 French francs, and 1 French franc buys 0.21 US dollars. Then, by converting currencies, a clever trader can start with 1 US dollar and buy 0.5 * 10.0 * 0.21 = 1.05 US dollars, making a profit of 5 percent.

Your job is to write a program that takes a list of currency exchange rates as input and then determines whether arbitrage is possible or not.

Input

The input file will contain one or more test cases. On the first line of each test case there is an integer n ($1<=n<=30$), representing the number of different currencies. The next n lines each contain the name of one currency. Within a name no spaces will appear. The next line contains one integer m, representing the length of the table to follow. The last m lines each contain the name c_i of a source currency, a real number r_{ij} which represents the exchange rate from c_i to c_j and a name c_j of the destination currency. Note that c_i and c_j may be the same currency. Exchanges which do not appear in the table are impossible. Test cases are separated from each other by a blank line. Input is terminated by a value of zero (0) for n.

Output

For each test case, print one line telling whether arbitrage is possible or not in the format "Case *case*: Yes", respectively "Case *case*: No".

Example

Input:

```
3
USDollar
BritishPound
FrenchFranc
3
USDollar 0.5 BritishPound
BritishPound 10.0 FrenchFranc
FrenchFranc 0.21 USDollar

3
USDollar
BritishPound
FrenchFranc
6
USDollar 0.5 BritishPound
USDollar 4.9 FrenchFranc
BritishPound 10.0 FrenchFranc
BritishPound 1.99 USDollar
FrenchFranc 0.09 BritishPound
FrenchFranc 0.19 USDollar

0
```

Output:

```
Case 1: Yes
Case 2: No
```

Submit solution! (/submit/ARBITRAG/)

hide comments

< | Previous | 1 | 2 (/problems/ARBITRAG/cstart=10)
3 (/problems/ARBITRAG/cstart=20) | Next (/problems/ARBITRAG/cstart=10)
> (/problems/ARBITRAG/cstart=20)

enigmus (/users/enigmus): 2019-06-06 10:56:26
Printing uppercase answers costed me a couple of WAs :(.
Solvable by basic bellman-ford, taking negative logarithm of edges is not required, just follow the edges by multiplying

Feedback

abhinav_jain02 (/users/abhinav_jain02): 2019-05-23 14:28:31
AC in one go
Apply bellman ford for every weakly connected component

surajkumar_cse (/users/surajkumar_cse): 2019-01-15 23:13:02
The problem demands to product the weight of the edges ,,, after lots of thinking and online help ... I was able to convert the product into sum. Hint: 12th standard maths.

jmr99 (/users/jmr99): 2018-08-15 15:23:05
FloyWar!!

abdelhameedddd (/users/abdelhameedddd): 2018-08-07 10:35:12
AC USING FLOYD :)


anirudnits (/users/anirudnits): 2018-06-18 15:04:47
@siva2697 thanks and my first AC using Floyd warshall :)

ameyanator (/users/ameyanator): 2018-03-30 13:37:37
I see small constraint I use Floyd Warshall :P. AC in 1 go :D

siva2697 (/users/siva2697): 2018-03-17 06:57:45
spoj tool kit wont be helpful after reading edges dont intialize dist[i][i]=1 bcoz input contains self loop.

Last edit: 2018-03-17 07:02:31

siva2697 (/users/siva2697): 2018-03-17 06:57:03
incase if u assume that dollar to dollar intial is 1 it isnt working
dont intialize dist[i][i]=1;
and dont use spoj toolkit for this problem giving wrong results.

sorablaze_11 (/users/sorablaze_11): 2017-11-21 06:15:51
Very weak test cases.

Leave a Comment

Publish

Notes:
1. Don't post any source code here.
2. Please be careful, leave short comments only. Don't spam here.
3. For more discussion (hints, ideas, solutions) please visit our forum (/forum).
4. Authors of the problems are allowed to delete the post and use html code here (e.g. to provide some useful links).

Feedback

Submit solution! (/submit/ARBITRAG/)

Added by:	Vincenzo Bonifaci (/users/voidstar)
Date:	2011-08-07
Time limit:	0.242s
Source limit:	50000B
Memory limit:	1536MB
Cluster:	Cube (Intel G860) (/clusters/)
Languages:	All except: ASM64
Resource:	University of Ulm Local Contest 1996

Evaluate this problem

Nobody has rated this problem yet, maybe you'll be the first?

Concept difficulty

easy normal hard extreme

Implementation difficulty

easy normal hard extreme

Recommend!

Feedback

Own tags

# # # # # # # # # # # # # #

Tag name Add

Feedback