**Arbitrage**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 6518    Accepted Submission(s): 3015**

Problem Description

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 pound, 1 British pound buys 10.0 French francs, and 1 French franc buys 0.21 US dollar. 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. Om 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 ci of a source currency, a real number rij which represents the exchange rate from ci to cj and a name cj of the destination 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".

Sample 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

Sample Output

Case 1: Yes

Case 2: No

Source

[University of Ulm Local Contest 1996](http://acm.hdu.edu.cn/search.php?field=problem&key=University+of+Ulm+Local+Contest+1996&source=1&searchmode=source)

分析：

题意：给几个国家，然后给这些国家之间的汇率。判断能否通过这些汇率差进行套利交易。

Floyd的算法可以求出任意两点间的最短路径，最后比较本国与本国的汇率差，如果大于1，则可以。否则不可以

这道题的做法和2112的做法是一样的 都是最短路加上map容器

容器的用法有在2112中讲过 这里就不多说了

AC：

#include <iostream>

#include <cstdio>

#include <cstring>

#include <map>

#define INF 0x3f3f3f3f

#define MAX 100

using namespace std;

int vis[MAX],dis[MAX];

double map1[MAX][MAX];

char currency[31];

int n,m;

void floyd()

{

for(int k = 0; k < n; k++)

for(int i = 0; i < n; i++)

for(int j = 0; j < n; j++)

if(map1[i][j] < map1[i][k]\*map1[k][j])

map1[i][j] = map1[i][k]\*map1[k][j];

for(int i = 0; i < n; i++)

if(map1[i][i] > 1.0)

{

printf("Yes\n");

return ;

}

printf("No\n");

return ;

}

int main()

{

int case1 = 1;

while(~scanf("%d",&n))

{

if(n==0)

break;

map<string,int>s;

for(int i = 0; i < n; i++)

{

scanf("%s",currency);

s[currency] = i;

}

scanf("%d",&m);

char p[31],q[31];

double c;

memset(map1,0,sizeof(map1));

for(int i = 0; i < m; i++)

{

scanf("%s%lf%s",p,&c,q);

map1[s[p]][s[q]] = c;

}

printf("Case %d: ",case1++);

floyd();

}

return 0;

}