

Problema 3

Exchange

Tempo limite: 0,5s (C/C++), 1,5s (Python), 2,0s(Java)

Limite de memória: 64MB

A volatilidade dos mercados financeiros implica na taxa de variação dos valores de diferentes moedas e ativos financeiros e que pode eventualmente levar à destruição do poder de compra de alguma moeda cuja taxa de câmbio cair de forma rápida em relação a outras moedas, fenômeno que ocorreu com diversas moedas nos últimos anos devido a diferentes possíveis fatores como instabilidade econômica, política ou social, políticas fiscais e monetárias, sanções econômicas, entre outros. Considerando que atualmente é possível ter em carteira diversas moedas e ativos precificados em diferentes moedas, uma forma de evitar o problema de se expor totalmente aos riscos atrelados a uma única moeda ou país seria o de ter em carteira diversas moedas, diversificando o risco da sua carteira.

Além de obter o benefício da diversificação, outra possível vantagem seria a de arbitragem, ou seja, a de encontrar oportunidades em distorções temporárias identificadas de acordo com as taxas de conversão entre diferentes moedas em algum dado instante e realizar uma sequência de trocas entre diversas moedas com a finalidade de se obter lucro com as operações. Essas operações são possíveis uma vez que elas podem ser realizadas em curtos períodos de tempo por meios digitais do sistema financeiro. Porém, não é possível realizar uma sequência muito longa de transações, pois a oportunidade dura apenas enquanto as taxas de conversão não são atualizadas. Um exemplo de tais operações poderia ser: 1 USD vale 5 BRL, 1 BRL vale 25 ARS e 1 ARS vale 0,0081 USD, então com 1 USD, é possível obter 1,0125 USD ($1 \times 5 \times 25 \times 0,0081 = 1,0125$), ou seja, lucro de 1,25%.

A sua tarefa é, dadas as taxas de conversão entre N moedas em um dado instante, encontrar uma oportunidade para se obter lucro de pelo menos 1% com alguma das moedas do conjunto, de forma que a sequência de transações inicie e termine com a mesma moeda e a quantidade de transações seja de no máximo N . Caso mais de uma sequência resulte em lucro de pelo menos 1%, você deve imprimir aquela com o menor número de transações que começa e termina com a moeda de menor identificador. Se para uma dada moeda de início/término houver mais de uma sequência viável de mesmo comprimento, retorne aquela de maior lucro.

Entrada

Uma entrada para o programa inicia-se com uma linha contendo um inteiro N ($2 \leq N \leq 20$), representando o número de moedas no conjunto considerado. Cada uma das próximas N linhas contém valores da tabela de conversão a partir de uma i -ésima moeda ($1 \leq i \leq N$), sendo uma moeda por linha apresentada em ordem crescente do identificador da moeda de origem da transação. As colunas na i -ésima linha da tabela apresentam os valores de conversão da i -ésima moeda para a j -ésima moeda, sendo que as colunas são apresentadas em ordem crescente de identificador ($1 \leq j \leq N, j \neq i$). Os valores de conversão para $i = j$ são iguais a 1 e, portanto, são omitidos da entrada.

Saída

Seu programa deve imprimir a sequência de transações (com os identificadores das moedas envolvidas) que satisfaz o problema, caso ela exista. Caso contrário, o programa deve imprimir “Impossible”.

Exemplo de Entrada

```
3
1.2 0.89
0.88 5.1
1.1 0.15
```

Exemplo de Saída

```
1 2 1
```

Exemplo de Entrada

```
2
2.0
0.45
```

Exemplo de Saída

```
Impossible
```

Exemplo de Entrada

```
6
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.1 0.0 0.0
0.0 0.0 1.0 0.0 0.0
0.0 5.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 9.0
0.0 0.0 0.0 0.0 1.0
```

Exemplo de Saída

```
5 6 5
```