OJ 10113

Exchange Rates

Prof. Edson Alves

Faculdade UnB Gama

Using money to pay for goods and services usually makes life easier, but sometimes people prefer to trade items directly without any money changing hands. In order to ensure a consistent "price", traders set an exchange rate between items. The exchange rate between two items A and B is expressed as two positive integers m and n, and says that m of item A is worth n of item B. For example, 2 stoves might be worth 3 refrigerators. (Mathematically, 1 stove is worth 1.5 refrigerators, but since it's hard to find half a refrigerator, exchange rates are always expressed using integers.)

Your job is to write a program that, given a list of exchange rates, calculates the exchange rate between any two items.

Usar dinheiro para pagar por bens e serviços geralmente torna a vida mais fácil, mas algumas vezes as pessoas preferem torcar itens diretamente, sem qualquer uso de dinheiro. Para garantir um "preço" consistente, os envolvidos na troca definem uma taxa de conversão entre os itens. A taxa de conversão entre os itens A e B é expressa por meio de dois inteiros positivos m e n, e diz que m unidades do item A valem n unidades do item B. Por exemplo, 2 fornos podem valer 3 refrigeradores. (Matematicamente, 1 forno vale um refrigerador e meio, mas como é difícil encontrar meio refrigerador, as taxas de conversão são sempre expressas usando inteiros.)

Sua tarefa á escrever um programa que, dada uma lista de taxas de conversão, calcule a taxa de conversão entre quaisquer dois itens.

Input

The input file contains one or more commands, followed by a line beginning with a period that signals the end of the file. Each command is on a line by itself and is either an assertion or a query. An assertion begins with an exclamation point and has the format

$$! \; m \; itema = n \; itemb$$

where $itema$ *and* $itemb$ *are distinct item names and* m *and* n *are both positive integers less than 100. This command says that* m *of* $itema$ *are worth* n *of* $itemb$*. A query begins with a question mark, is of the form*

$$? \; itema = itemb$$

and asks for the exchange rate between $itema$ *and* $itemb$*, where* $itema$ *and* $itemb$ *are distinct items that have both appeared in previous assertions (although not necessarily the same assertion).*

Input

A entrada contém um ou mais comandos, seguidos de uma linha que começa com um ponto final, o qual indica o fim da entrada. Cada comando está contido em uma linha e ou é uma afirmação ou uma consulta. Uma afirmação começa com um ponto de exclamação e tem o formato

$$! \; m \; itema = n \; itemb$$

onde $itema$ **e** $itemb$ **são nomes de itens diferntes e** m **e** n **são inteiros positivos menores do que 100. Este comando diz que** m **unidades do** $itema$ **valem** n **unidades do** $itemb$**. Uma consulta começa com uma exclamação, e tem a forma**

$$? \; itema = itemb$$

e requer a taxa de conversão entre $itema$ **e** $itemb$**, onde** $itema$ **e** $itemb$ **são itens diferentes e ambos aparecerem em afirmativas anteriores (não necessariamente na mesma afirmação).**

Output

For each query, output the exchange rate between $itema$ and $itemb$ based on all the assertions made up to that point. Exchange rates must be in integers and must be reduced to lowest terms. If no exchange rate can be determined at that point, use question marks instead of integers. Format all output exactly as shown in the example.

Saída

Para cada consulta imprima a taxa de conversão entre $itema$ e $itemb$, baseada em todas as afirmativas feitas até o momento. Taxas de conversão devem ser inteiros e não devem ter divisores comuns. Se não é possível determinar a taxa de conversão neste ponto, substitua os inteiros por pontos de interrogação. Formate a saída conforme o exemplo dado.

Notes

- *Item names will have length at most 20 and will contain only lowercase letters.*
- *Only the singular form of an item name will be used (no plurals).*
- *There will be at most 60 distinct items.*
- *There will be at most one assertion for any pair of distinct items.*
- *There will be no contradictory assertions. For example, “2 pig = 1 cow”, “2 cow = 1 horse”, and “2 horse = 3 pig” are contradictory.*
- *Assertions are not necessarily in lowest terms, but output must be.*
- *Although assertions use numbers less than 100, queries may result in larger numbers that will not exceed 10000 when reduced to lowest terms*

Notas

- Nomes dos itens são compostos por, no máximo, 20 caracteres alfabéticos minúsculos.
- Apenas a forma singular do nome do item será usada (sem plurais).
- Haverão, no máximo, 60 itens diferentes.
- Haverá no máximo uma afirmativa para qualquer par de itens distintos.
- Não haverão afirmativas contraditórias. Por exemplo, “2 porco = 1 vaca”, “2 vaca = 1 cavalo” e “2 cavalo = 3 porco” são contraditórias.
- Afirmativas podem conter divisores comuns, mas a saída não deve conter divisores comuns.
- Embora as afirmativas usem números menores do que 100, as respostas das consultas podem ser números maiores, os quais não excedem 10000 quando eliminados os divisores comuns.

Notas

- Nomes dos itens são compostos por, no máximo, 20 caracteres alfabéticos minúsculos.
- Apenas a forma singular do nome do item será usada (sem plurais).
- Haverão, no máximo, 60 itens diferentes.
- Haverá no máximo uma afirmativa para qualquer par de itens distintos.
- Não haverão afirmativas contraditórias. Por exemplo, “2 porco = 1 vaca”, “2 vaca = 1 cavalo” e “2 cavalo = 3 porco” são contraditórias.
- Afirmativas podem conter divisores comuns, mas a saída não deve conter divisores comuns.
- Embora as afirmativas usem números menores do que 100, as respostas das consultas podem ser números maiores, os quais não excedem 10000 quando eliminados os divisores comuns.

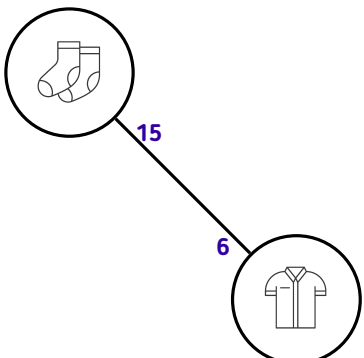
Exemplo de entrada e saída

Exemplo de entrada e saída

```
! 6 shirt = 15 sock
```

Exemplo de entrada e saída

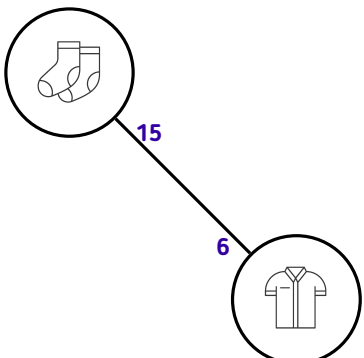
! 6 shirt = 15 sock



Exemplo de entrada e saída

! 6 shirt = 15 sock

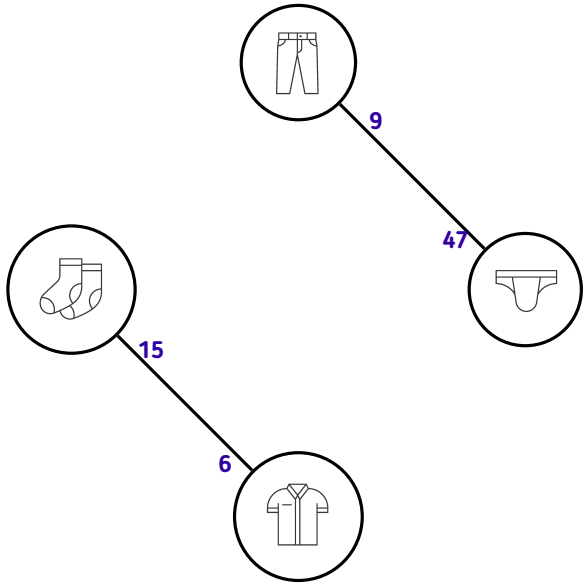
! 47 underwear = 9 pant



Exemplo de entrada e saída

! 6 shirt = 15 sock

! 47 underwear = 9 pant

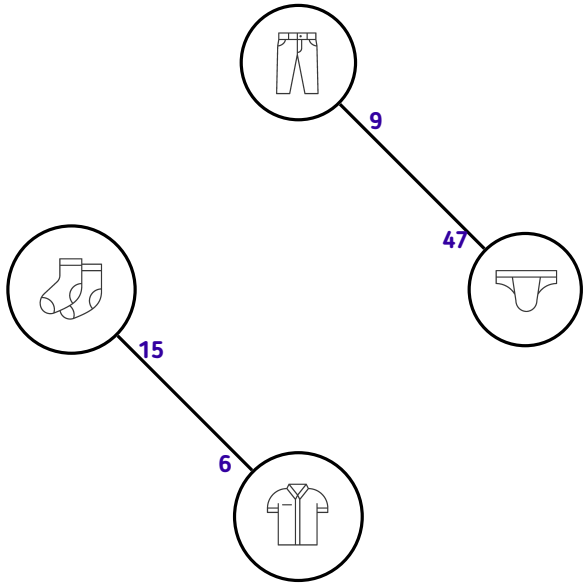


Exemplo de entrada e saída

! 6 shirt = 15 sock

! 47 underwear = 9 pant

? sock = shirt

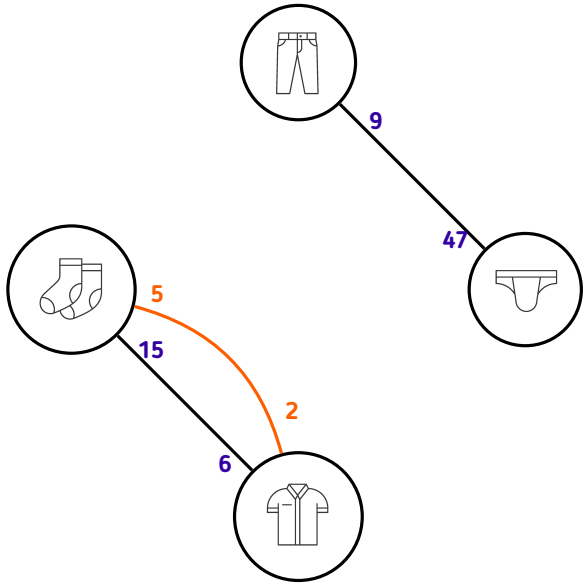


Exemplo de entrada e saída

! 6 shirt = 15 sock

! 47 underwear = 9 pant

? sock = shirt



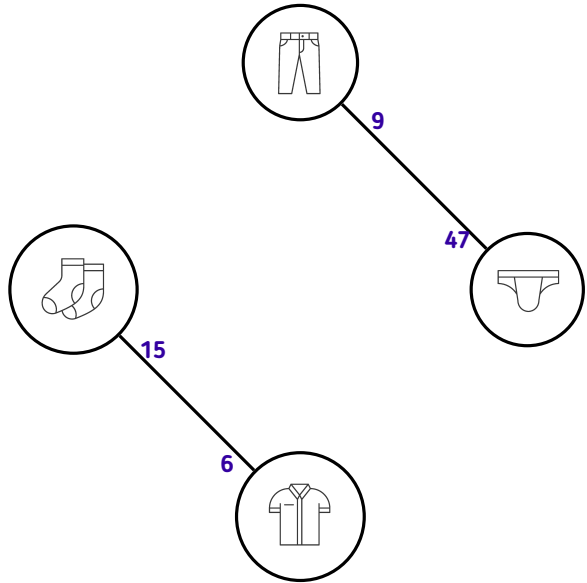
Exemplo de entrada e saída

! 6 shirt = 15 sock

! 47 underwear = 9 pant

? sock = shirt

? shirt = pant



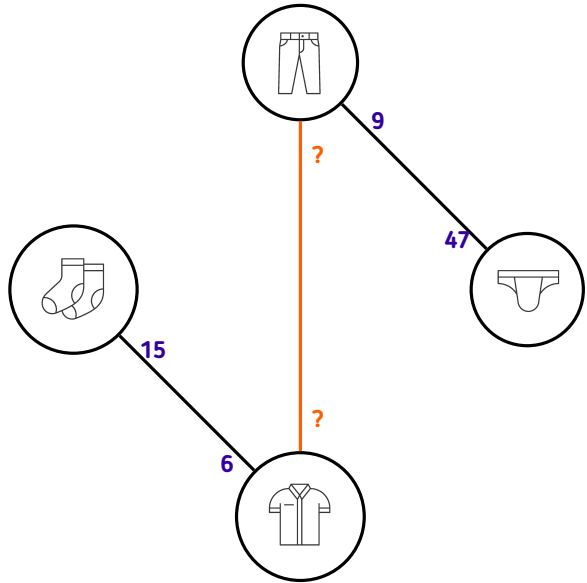
Exemplo de entrada e saída

! 6 shirt = 15 sock

! 47 underwear = 9 pant

? sock = shirt

? shirt = pant



Exemplo de entrada e saída

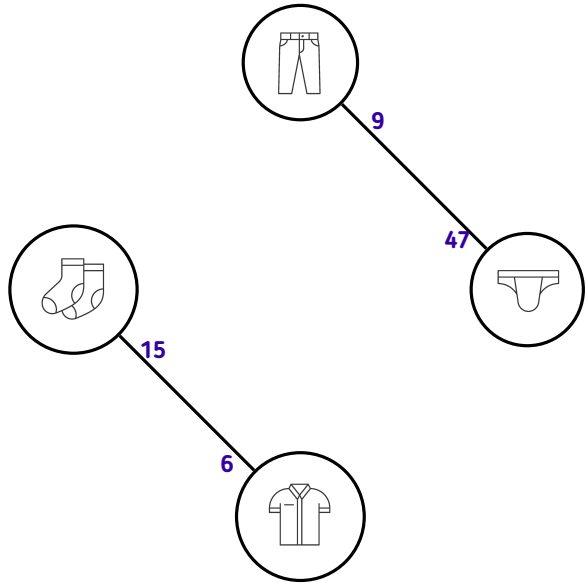
! 6 shirt = 15 sock

! 47 underwear = 9 pant

? sock = shirt

? shirt = pant

! 2 sock = 1 underwear



Exemplo de entrada e saída

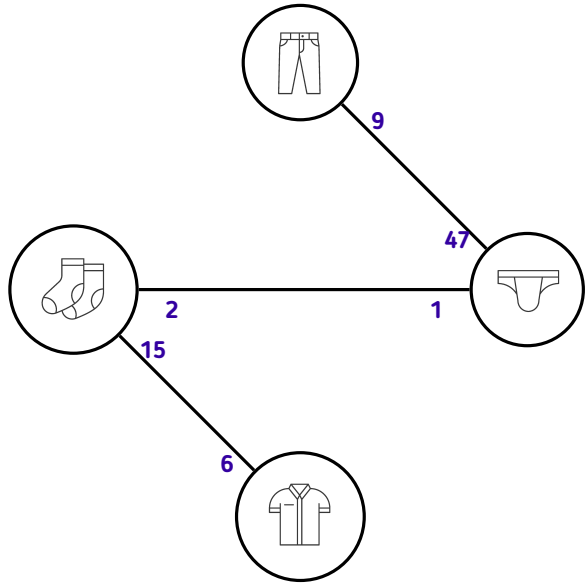
! 6 shirt = 15 sock

! 47 underwear = 9 pant

? sock = shirt

? shirt = pant

! 2 sock = 1 underwear



Exemplo de entrada e saída

! 6 shirt = 15 sock

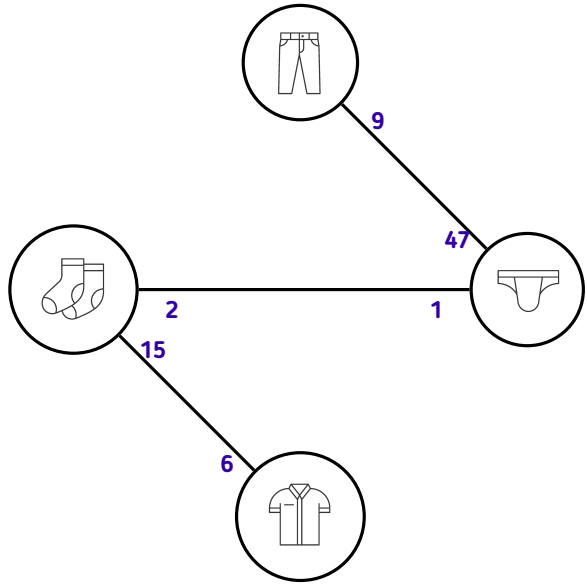
! 47 underwear = 9 pant

? sock = shirt

? shirt = pant

! 2 sock = 1 underwear

? shirt = pant



Exemplo de entrada e saída

! 6 shirt = 15 sock

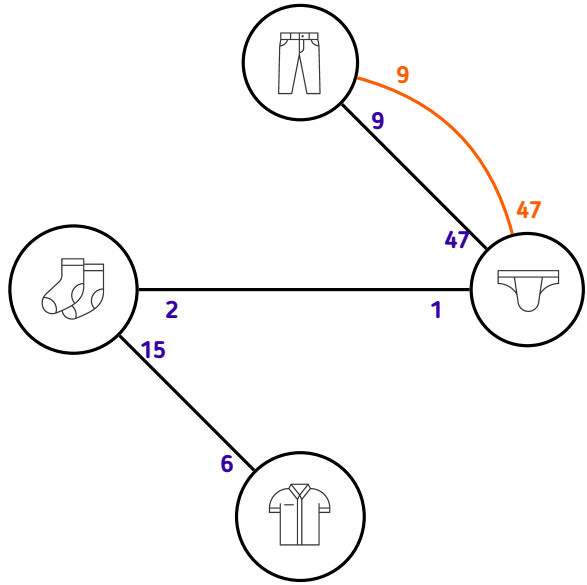
! 47 underwear = 9 pant

? sock = shirt

? shirt = pant

! 2 sock = 1 underwear

? shirt = pant



Exemplo de entrada e saída

! 6 shirt = 15 sock

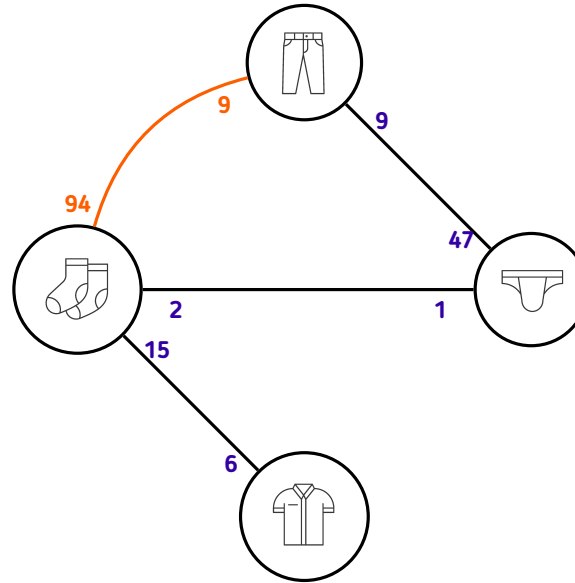
! 47 underwear = 9 pant

? sock = shirt

? shirt = pant

! 2 sock = 1 underwear

? shirt = pant



Exemplo de entrada e saída

! 6 shirt = 15 sock

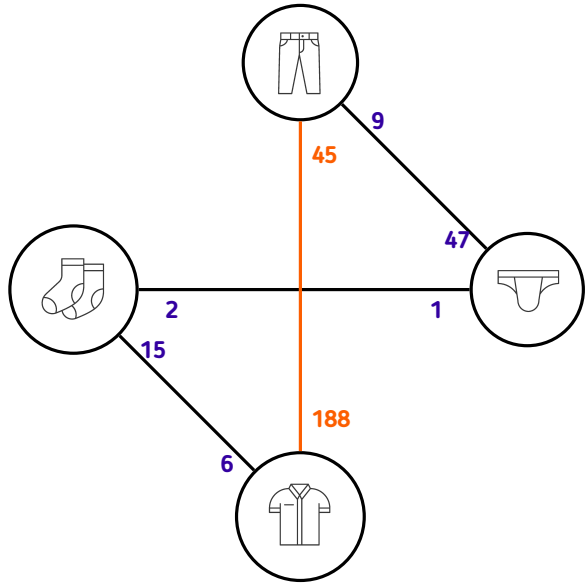
! 47 underwear = 9 pant

? sock = shirt

? shirt = pant

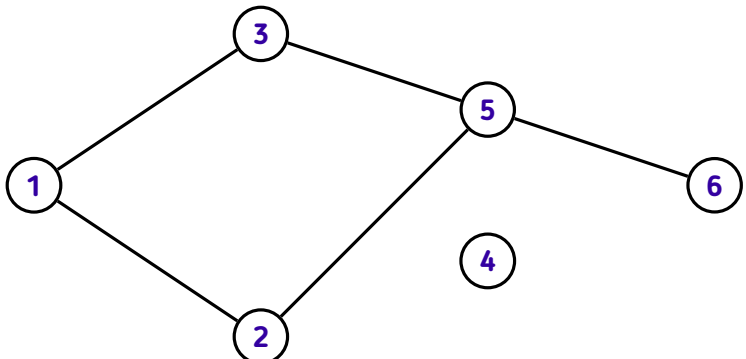
! 2 sock = 1 underwear

? shirt = pant



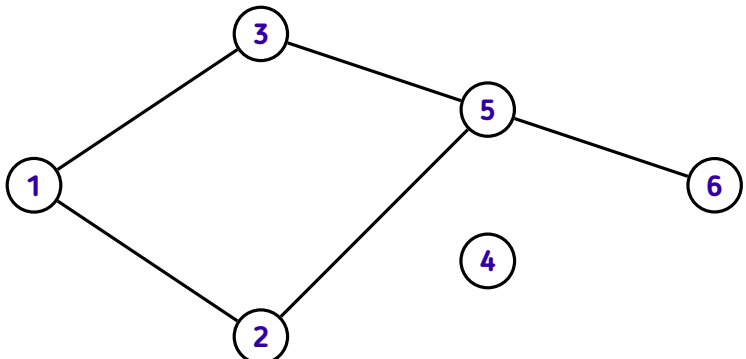
Solução

Solução



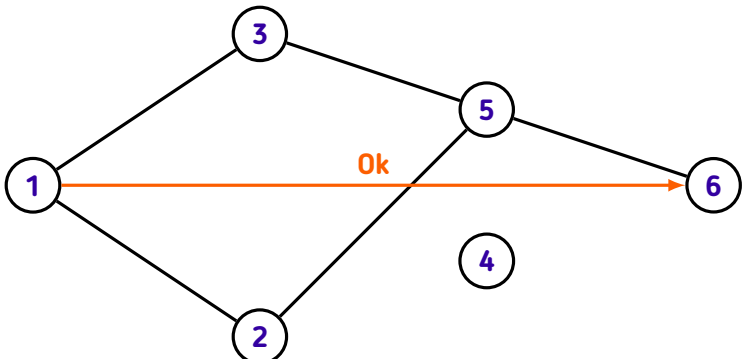
Solução

Só há conversão entre objetos se há ao menos um caminho entre eles



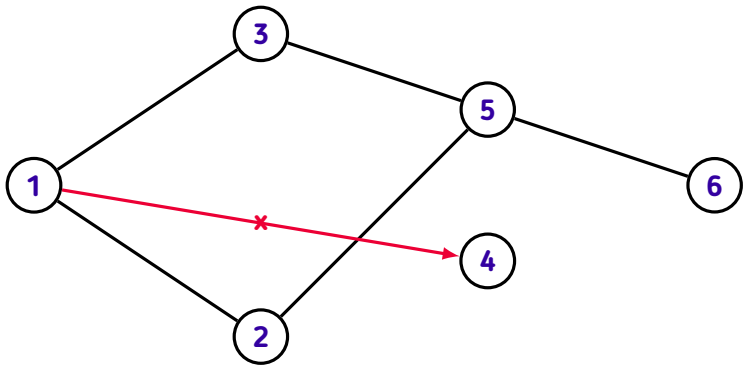
Solução

Só há conversão entre objetos se há ao menos um caminho entre eles



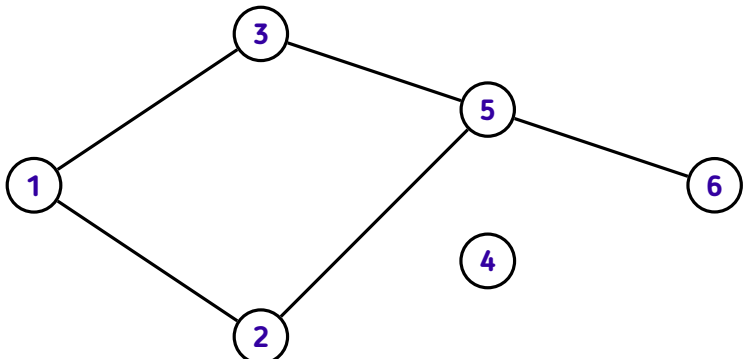
Solução

Só há conversão entre objetos se há ao menos um caminho entre eles



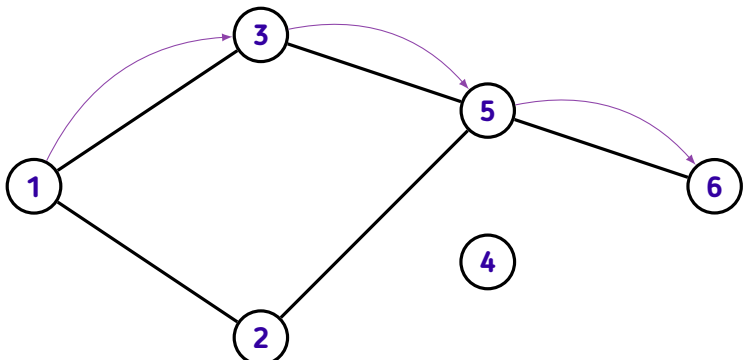
Solução

Pela consistência da entrada este caminho não precisa ser único



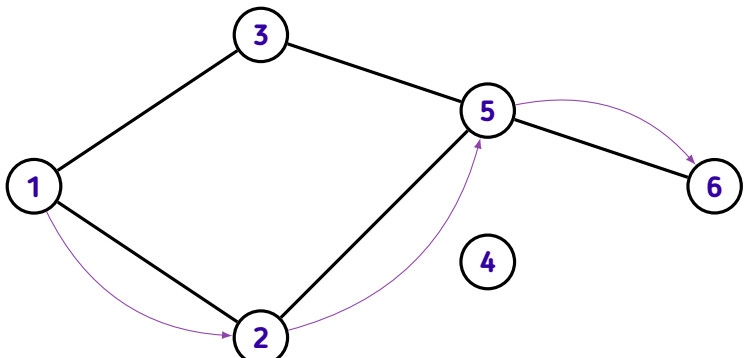
Solução

Pela consistência da entrada este caminho não precisa ser único



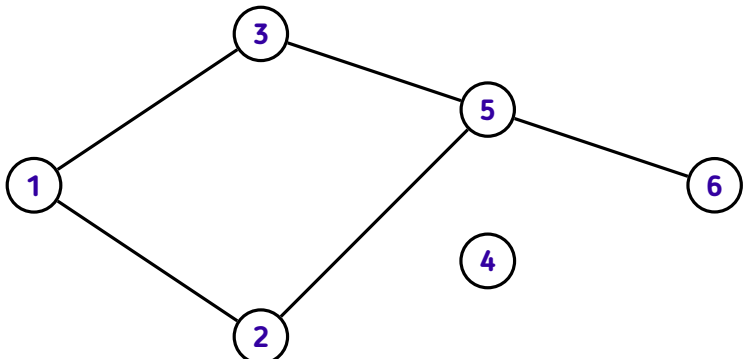
Solução

Pela consistência da entrada este caminho não precisa ser único



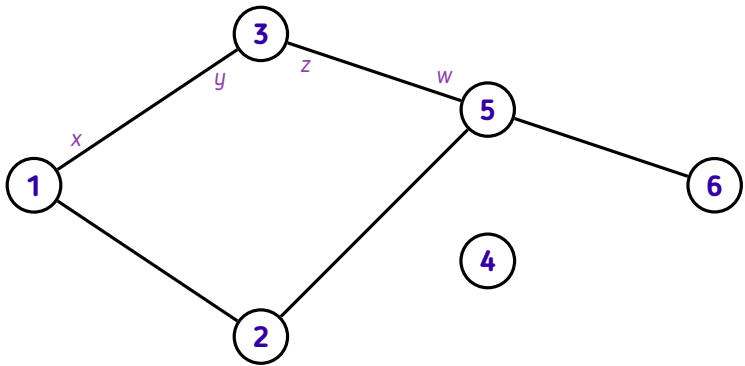
Solução

As taxas de conversão compõem por multiplicação



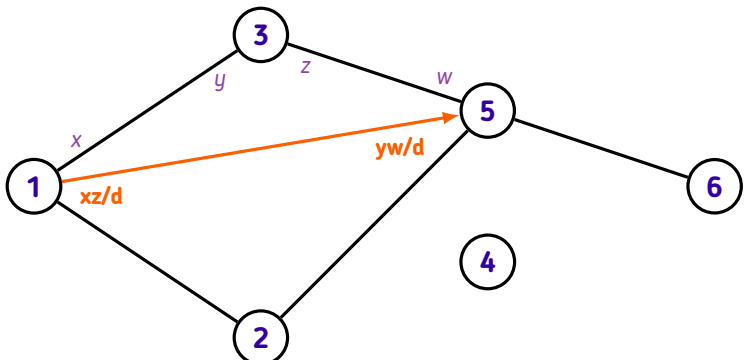
Solução

As taxas de conversão compõem por multiplicação



Solução

As taxas de conversão compõem por multiplicação ($d = gcd(xz, yw)$)



```
ii dfs(const string& u, const string& b, int x, int y, set<string>& found)
{
    if (found.count(u))
        return ii(0, 0);

    if (u == b)
        return ii(x, y);

    found.insert(u);

    for (auto [v, z, w] : adj[u])
    {
        auto p = x*z, q = y*w, d = gcd(p, q);
        auto [r, s] = dfs(v, b, p/d, q/d, found);

        if (r and s)
            return ii(r, s);
    }

    return ii(0, 0);
}
```

```
        switch (cmd.front()) {
        case '!':
        {
            int x, y;
            string a, b, sep;

            cin >> x >> a >> sep >> y >> b;

            auto d = gcd(x, y);
            x /= d;
            y /= d;

            adj[a].emplace_back(b, x, y);
            adj[b].emplace_back(a, y, x);

            break;
        }
```

```
        case '?':
        {
            string a, b, sep;
            cin >> a >> sep >> b;

            set<string> found;

            auto [x, y] = dfs(a, b, 1, 1, found);

            if (x and y)
                cout << x << ' ' << a << " = " << y << ' ' << b << '\n';
            else
                cout << "? " << a << " = ? " << b << '\n';
        }
```