

Mapa sem sentidos únicos

Pretendemos reconstruir um mapa de uma rede viária a partir de um conjunto de trajetos que, **globalmente**, cobrem todas as suas ligações. Para isso, vão ser analisados esses trajetos e vai ser construído um grafo dirigido que corresponde à rede. Cada trajeto é definido por uma sequência de inteiros $v_1 c_1 v_2 c_2 v_3 c_3 \dots v_{k-1} c_{k-1} v_k$, para algum k maior ou igual a 2, em que v_1 é a origem e v_k é o fim do trajeto, e cada segmento $v_i c_i v_{i+1}$ designa o ramo (v_i, v_{i+1}) e c_i o seu comprimento (em Km). Todas as ligações são **bidirecionais**. Como forma de teste, iremos analisar o comprimento dos trajetos dados e o número de adjacentes de **todos** os nós da rede.



Input

Na primeira linha tem o número de trajetos a analisar e o número de nós da rede (ambos são positivos).

Em cada uma das linhas seguintes tem a descrição de um trajeto. No início de cada linha tem o número de nós que constituem o trajeto (isto é, o valor de k correspondente) e a seguir tem a sequência que define o trajeto.

Os nós da rede são identificados por inteiros consecutivos, a partir de 1. A rede pode ter nós isolados. Se for útil, pode assumir que a rede não tem mais do que 20000 nós.

Para CC1024:

- Na implementação em Python, deve usar um **dicionário** para registar a informação sobre os ramos da rede (i.e., as ligações). Para cada ramo (u, v) , a **chave** é o par (u, v) e o **valor** é o seu comprimento. Deve registar o par (u, v) e o par (v, u) .
- Os **adjacentes** de um nó são os nós a que se encontra ligado diretamente.
- Para ler uma linha com vários inteiros, separados por espaços, e os colocar numa lista **P** pode escrever

```
P = list(map(int, input().split()))
```
- Se um percurso tiver k nós então tem $k-1$ ramos.

Output

Será constituído por dois blocos. No primeiro bloco tem a informação sobre o comprimento total de cada um dos trajetos dados: cada linha tem o formato "Trajeto p: c", em que p identifica o trajeto (pelo número de ordem em que ocorreu) e c o seu comprimento. No segundo bloco tem a informação sobre o número de adjacentes de cada nó da rede: cada linha segue o formato "No v: t", em que v identifica o nó e t é o número de adjacentes de v.

Exemplo 1

Input

```
3 7
9 2 4 3 4 2 3 1 20 5 4 6 1 3 10 5 7 2
5 3 8 1 9 6 4 5 7 2
4 1 3 2 7 5 1 4
```

Output

```
Trajeto 1: 53
Trajeto 2: 28
Trajeto 3: 11
No 1: 4
No 2: 3
No 3: 4
No 4: 1
No 5: 5
No 6: 3
No 7: 0
```

Exemplo 2

Input

```
1 9
14 6 4 5 7 2 4 3 4 9 4 3 4 2 3 1 20 5 4 6 1 3 10 5 7 2 3 1
```

Output

```
Trajeto 1: 75
No 1: 2
No 2: 3
No 3: 4
No 4: 0
No 5: 4
No 6: 2
No 7: 0
No 8: 0
No 9: 1
```