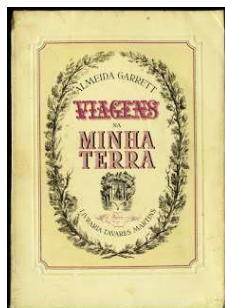


# Problema G

## Viagens na minha terra



Não vamos falar de Carlos, Frei Dinis e Joaninha, personagens desse clássico de Almeida Garrett, editado em 1846, mas de um grupo de pessoas que está a tentar reservar uma viagem no portal da empresa *Viagens na minha terra – Vá para fora cá dentro* (VMT@pt). O grupo pretende viajar até um certo local, devendo a viagem decorrer num certo período de tempo. A VMT@pt efetua várias rotas, algumas com passagem na origem e no destino pretendidos, no período previsto, e outras não. Nem todas as rotas têm já lugares suficientes para o grupo. A VMT@pt tenta cumprir horários, mas o estado de alguns troços dos trajetos tem provocado atrasos. Por isso, tem um aviso no seu portal com informação sobre os troços em que há problemas. O grupo gostaria de evitar problemas que atrasem a sua viagem, o que pode acontecer se os problemas ocorrerem entre a origem da rota e o destino do grupo na rota selecionada para a viagem. Por isso, estão a tentar saber quantas alternativas teriam com número mínimo de problemas (que pode não ser zero).

### Tarefa

Escrever um programa para indicar quantas alternativas o grupo teria com número mínimo de problemas (só interessa os que afetem a sua viagem). As rotas adequadas para o grupo devem respeitar as restrições de horário, de lugares e passar na origem e destino do grupo, claro!

### Input

Na primeira linha tem cinco inteiros: número de elementos do grupo, origem do grupo, destino do grupo, e dois inteiros (entre 7 e 22) que representam o intervalo em que a viagem deverá decorrer (as horas de partida e chegada do grupo terão de pertencer a esse intervalo). Na segunda linha tem um inteiro  $n$  (inferior a 200) que é o número de locais existentes (os locais são identificados por inteiros de 1 a  $n$ ). Segue-se uma tabela quadrada, com  $n$  linhas e  $n$  colunas: o elemento que está na linha  $i$  e na coluna  $j$  é 1 se existir algum problema no troço de  $i$  para  $j$ , é 0 se não existir problema, e é 2 se o troço não existir.

Depois tem a descrição das rotas, seus horários, e lugares disponíveis atualmente. Para cada rota, tem duas linhas com inteiros

$$k \quad h \\ v_1 \quad n_1 \quad d_1 \quad v_2 \quad n_2 \quad d_2 \quad v_3 \quad n_3 \quad d_3 \quad \dots \quad v_{k-1} \quad n_{k-1} \quad d_{k-1} \quad v_k$$

em que  $k$  (maior ou igual a 2) é o número de locais por onde a rota passa e  $h$  é a hora de partida da rota (sempre a horas certas), cada  $v_i$  é um local, cada troço  $(v_i, v_{i+1})$  é sempre válido, o valor  $n_i$  é o número de lugares disponíveis nesse troço e  $d_i$  é a duração prevista para esse troço (em minutos), nessa rota. As rotas são unidirecionais e nenhuma rota passa duas vezes pelo mesmo local. O input termina com 0 0 (dois zeros, separados por espaço).

## Output

Se não for possível efetuar a reserva da viagem, terá uma linha com a palavra **Impossible**. Se for possível, terá uma linha com dois inteiros **c p**, separados por um espaço, em que **c** é o número de alternativas encontradas e **p** é o número de problemas entre a origem da rota e o destino do grupo.

## Exemplo 1

### Input

```
10 5 2 10 15
7
2 1 2 1 1 2 2
0 2 0 2 2 2 1
1 0 2 0 1 2 2
0 2 0 2 0 1 2
0 2 0 0 2 2 2
2 2 2 2 0 2 0
0 0 2 2 2 1 2
7 8
1 20 30 2 13 50 7 15 15 6 25 12 5 17 15 3 12 23 4
4 11
5 30 60 3 2 10 2 15 23 7
7 9
4 20 70 6 13 10 5 18 30 3 10 90 2 25 90 7 2 120 1
5 7
7 13 17 6 12 50 5 20 23 4 17 35 3
7 12
6 2 10 5 10 23 4 12 30 3 15 35 2 3 60 7 18 35 1
4 14
6 20 30 5 11 20 4 2 5 3
7 9
3 7 45 1 18 20 5 15 30 4 10 40 6 13 30 7 12 75 2
4 9
3 12 40 4 10 5 1 16 60 2
0 0
```

### Output

1 0

### Exemplo 2

#### Input

11 5 2 7 20  
7  
2 1 2 1 1 2 2  
0 2 0 2 2 2 1  
1 0 2 0 1 2 2  
0 2 0 2 0 1 2  
0 2 0 0 2 2 2  
2 2 2 2 0 2 0  
0 0 2 2 2 1 2  
7 8  
6 2 90 5 8 30 4 12 35 3 15 45 2 3 25 7 18 35 1  
4 16  
6 20 30 5 11 90 4 2 25 3  
7 18  
6 2 10 5 15 20 4 2 25 3 5 30 2 3 30 7 18 95 1  
6 19  
6 2 30 5 15 30 4 20 25 3 15 30 2 3 30 7  
0 0

### Output

Impossible