

# MC202 — ESTRUTURAS DE DADOS

## Lab 05 — Pilhas

### Problema<sup>1</sup>

Como se sabe, no Brasil, mal acaba um [carnaval](#), as [escolas de samba](#) já começam a se preparar para as festividades do ano seguinte. Neste ano, o Governo Federal decidiu tomar providências para que o carnaval do ano seguinte tenha as festividades distribuídas em localidades menores, como, por exemplo, pequenos distritos, com o objetivo de evitar as aglomerações nas grandes cidades e diminuir o policiamento necessário para garantir a segurança dos foliões.

Em contato com as prefeituras de diversas pequenas cidades, o Governo Federal descobriu as principais dificuldades que estas cidades têm para realizar seus próprios desfiles. A principal delas é a manobra das [alegorias](#) nas estreitas ruas das cidades, pois não é possível que uma alegoria ultrapassasse outra em nenhuma das ruas da cidade. Com isso, é necessário usar as ruas perpendiculares à rua principal de cada cidade, que é onde o desfile ocorrerá. As ruas perpendiculares à principal não têm saída e estão sempre à direita do sentido da via principal. Além do mais, o Governo Federal informou-se de que cada alegoria será construída por um grupo distinto de pessoas, nas diversas zonas rurais próximas à cidade. Por isso, no esperado dia do desfile, essas pessoas não necessariamente chegarão com a alegoria à rua principal da cidade na ordem correta para o desfile.

Como normalmente os organizadores do desfile não têm os conceitos básicos de Ciência da Computação, eles têm dificuldade em reordenar as alegorias e coordenar as tais manobras. Dessa forma, o Governo Federal considerou que você, aluno de MC202, já é capaz de implementar um programa que indique como reordenar as alegorias manobrando-as nas pequenas ruas ou indicar que a reordenação não é possível (neste caso, o desfile será cancelado). Seu programa será distribuído às prefeituras de cada uma dessas pequenas cidades.

Alguns cidadãos ouviram falar de tal projeto e sobre a possibilidade de manobra das grandes alegorias nas pacatas ruas de suas residências (as ruas sem saída perpendiculares à rua principal). Devido aos protestos, as prefeituras entraram em acordo com os cidadãos e decidiram que cada carro alegórico poderia utilizar no máximo uma rua auxiliar (sem saída) para se manobrar. Além disso, as prefeituras também decidiram que sempre que for necessário usar uma rua perpendicular para desviar uma alegoria, ela deverá entrar na primeira rua em que não bloqueie a saída de outra alegoria.

---

<sup>1</sup> Problema inspirado no problema STPAR do Sphere Online Judge (SPOJ).

## Entrada

A entrada do programa conterá duas linhas. Na primeira linha, haverá um número  $n > 0$  e um número  $r \geq 0$  indicando, respectivamente, a quantidade de alegorias esperadas para a festividade e a quantidade de ruas auxiliares que estarão disponíveis aos foliões para manobrar as alegorias. Na segunda linha de entrada, haverá  $n$  números distintos  $x_i$ ,  $1 \leq x_i \leq n$ , que identificam as alegorias e aparecem na ordem em que os carros chegam à rua principal da cidade. Cada  $x_i$  indica a ordem em que a alegoria  $i$  deve entrar no desfile.

## Saída

Caso seja possível ordenar as alegorias utilizando as ruas perpendiculares, o programa deverá imprimir na primeira linha a mensagem `Desfile pode ser realizado:.` Nas linhas seguintes, deverão ser impressos os movimentos das alegorias (com a melhor solução possível) de modo que elas entrem para o desfile na ordem indicada. Essa impressão será de acordo com o especificado a seguir.

- Quando uma alegoria puder seguir/entrar para o desfile sem precisar ser manobrada em uma das ruas auxiliares, deverá ser impressa a mensagem `Alegoria  $n$  desfila.`, onde  $n$  é o número da alegoria.
- Quando uma alegoria precisar ser manobrada em uma das ruas auxiliares, deverá ser impressa a mensagem `Alegoria  $n$  entra na rua  $m$  e aguarda.`, onde  $n$  é o número da alegoria e  $m$  é o número da rua (em ordem de disponibilidade).
- Quando uma alegoria sai de uma das ruas auxiliares e segue/entra para o desfile, deverá ser impressa a mensagem `Alegoria  $n$  sai da rua  $m$  e desfila.`, onde  $n$  é o número da alegoria e  $m$  é o número da rua onde ela estava.

Caso não seja possível ordenar as alegorias com as ruas perpendiculares disponíveis, o programa deverá imprimir somente a mensagem `Sem desfile.`

# Exemplo 1

## Entrada

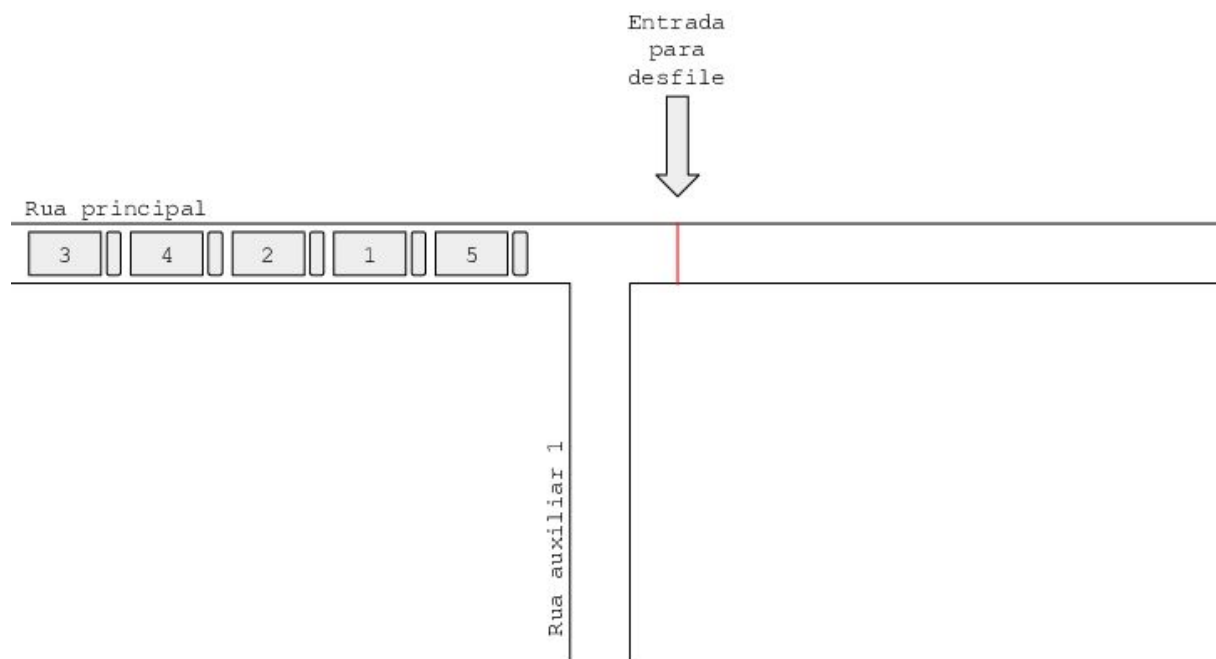
```
5 1
5 1 2 4 3
```

## Saída

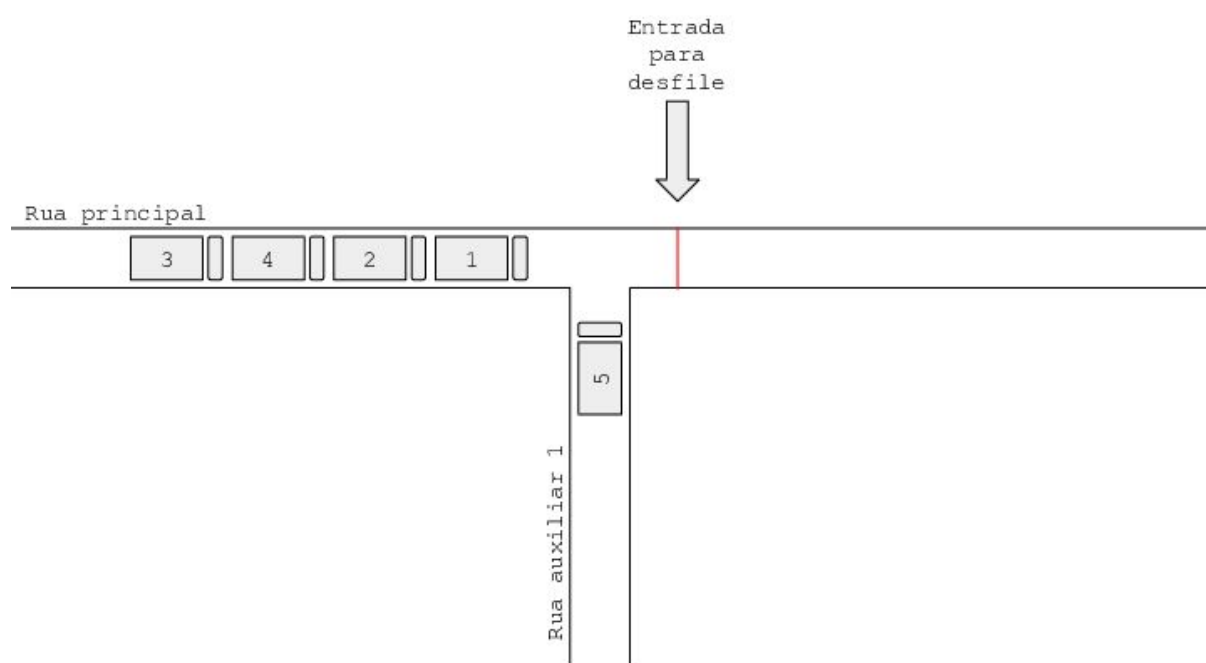
```
Desfile pode ser realizado:
Alegoria 5 entra na rua 1 e aguarda.
Alegoria 1 desfila.
Alegoria 2 desfila.
Alegoria 4 entra na rua 1 e aguarda.
Alegoria 3 desfila.
Alegoria 4 sai da rua 1 e desfila.
Alegoria 5 sai da rua 1 e desfila.
```

## Representação da saída esperada

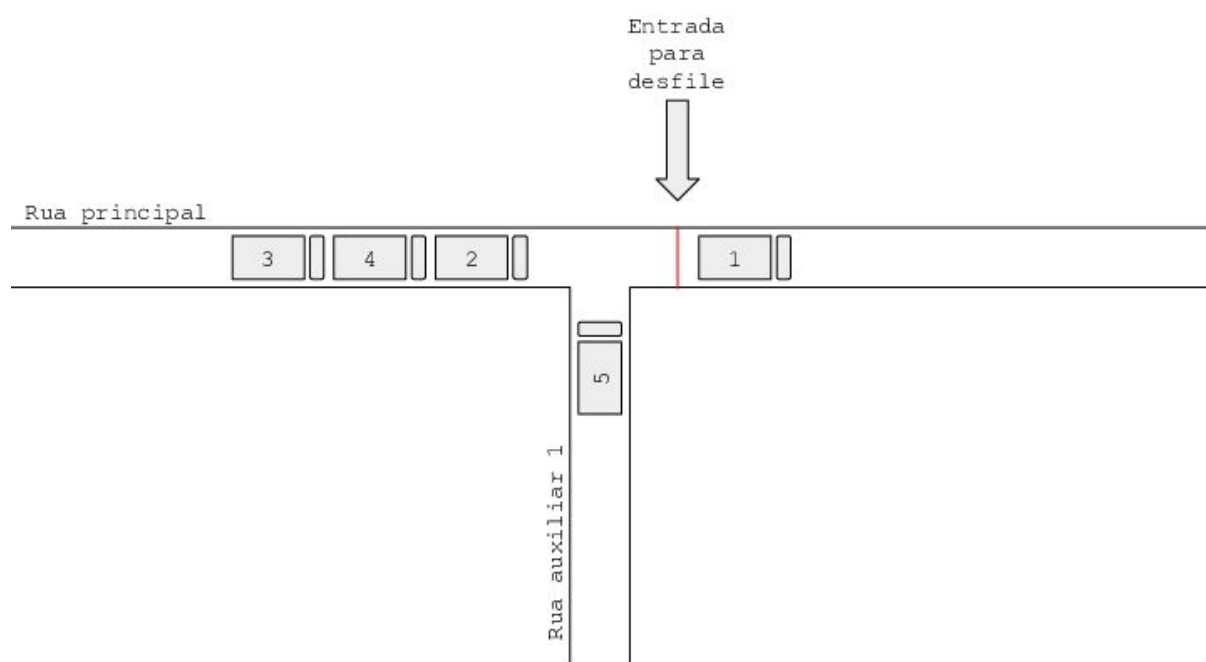
Desfile pode ser realizado:



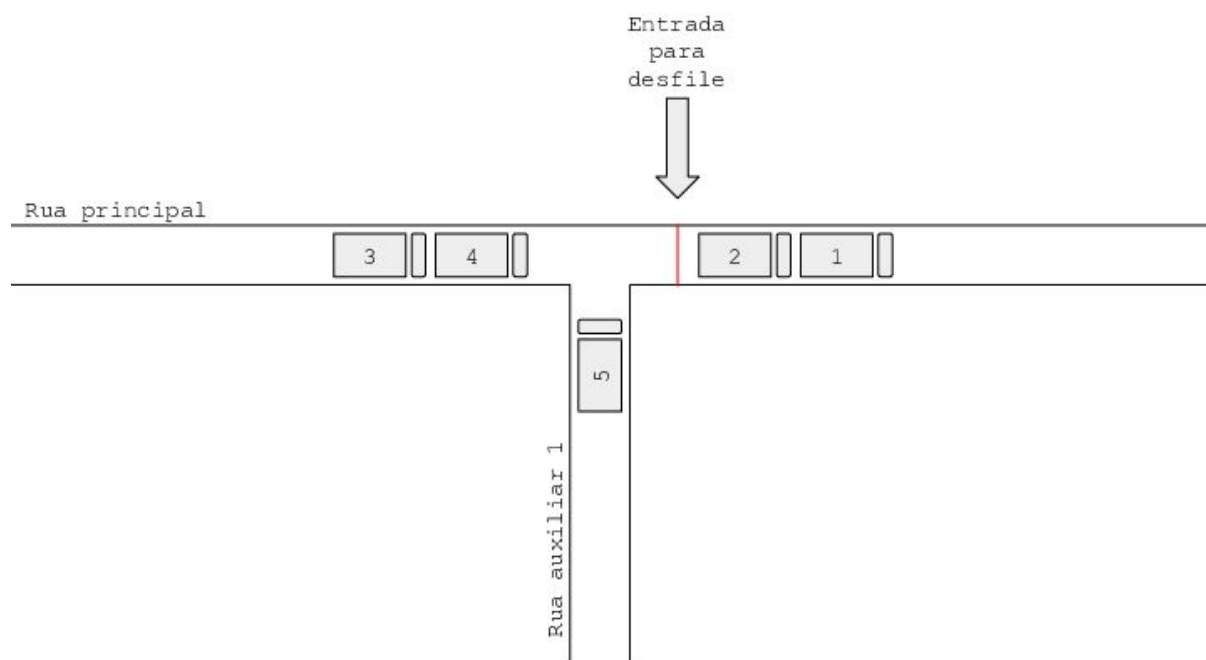
Alegoria 5 entra na rua 1 e aguarda.



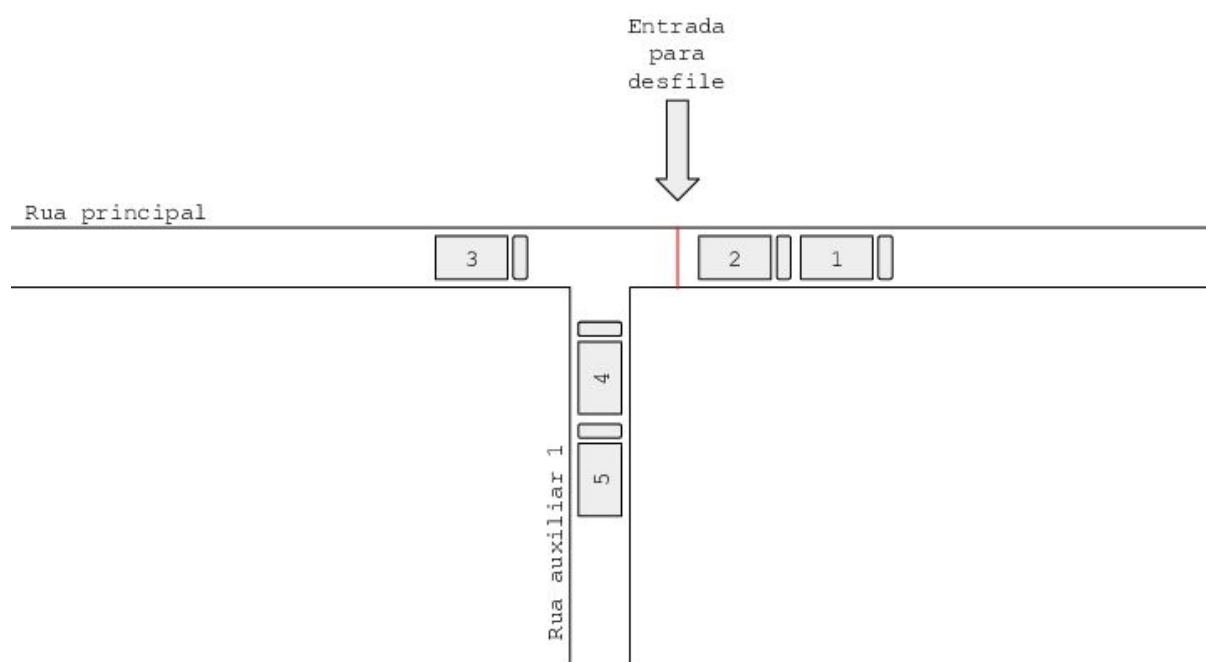
Alegoria 1 desfila.



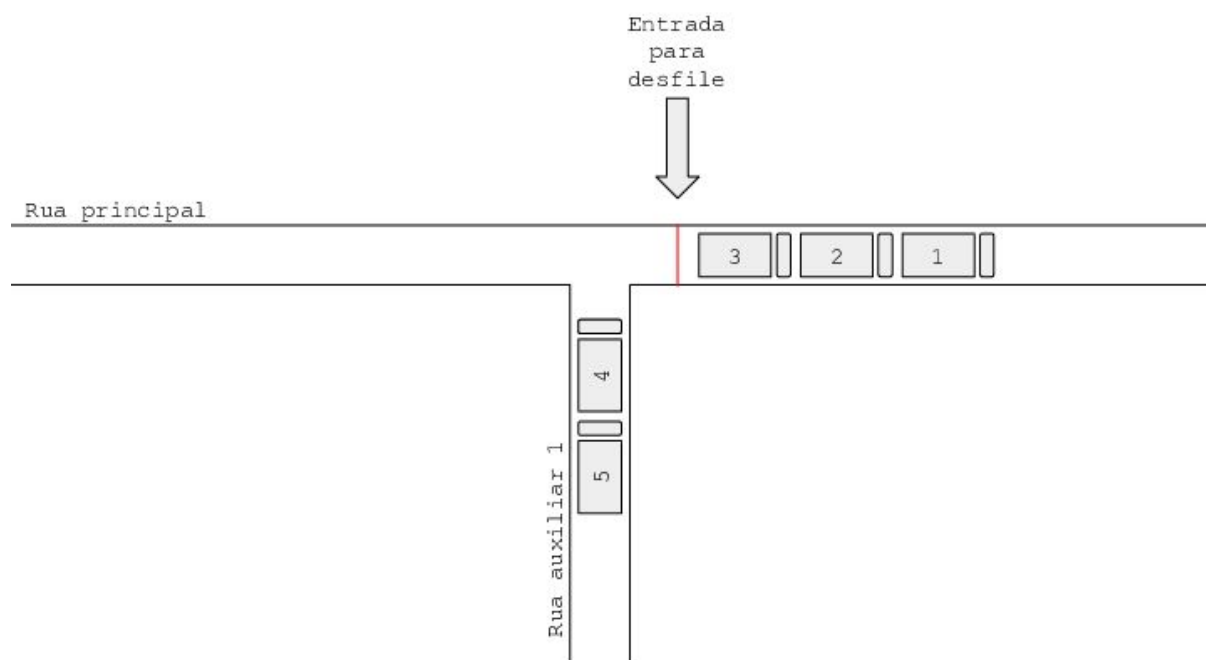
Alegoria 2 desfila.



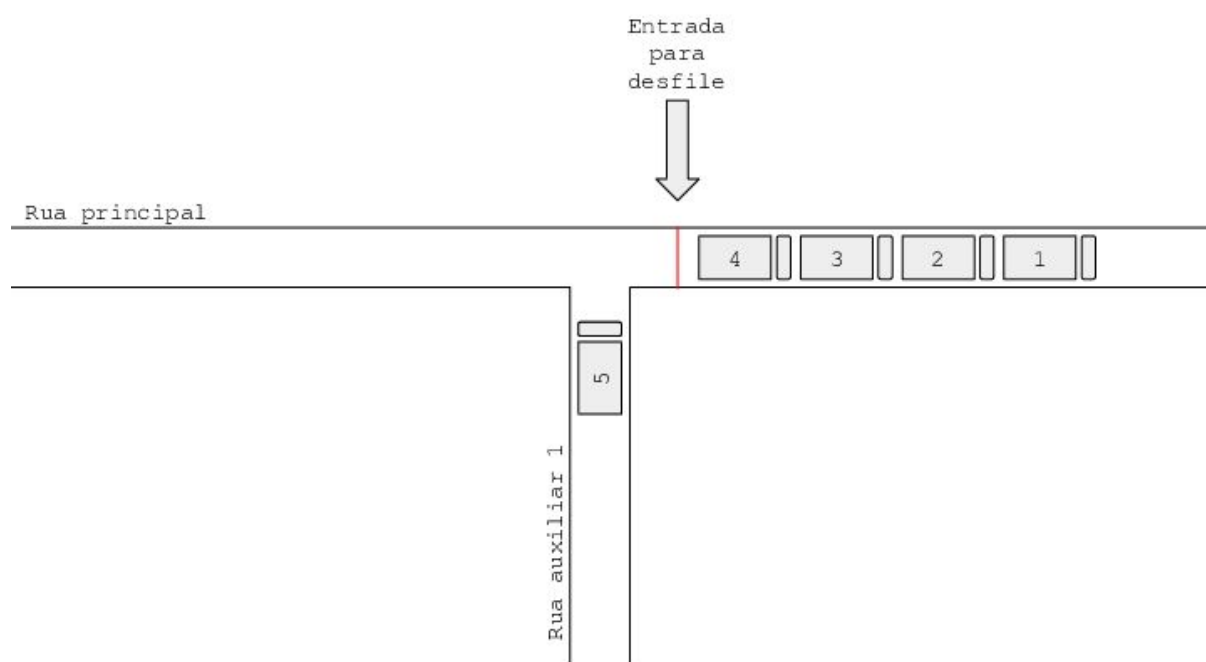
Alegoria 4 entra na rua 1 e aguarda.



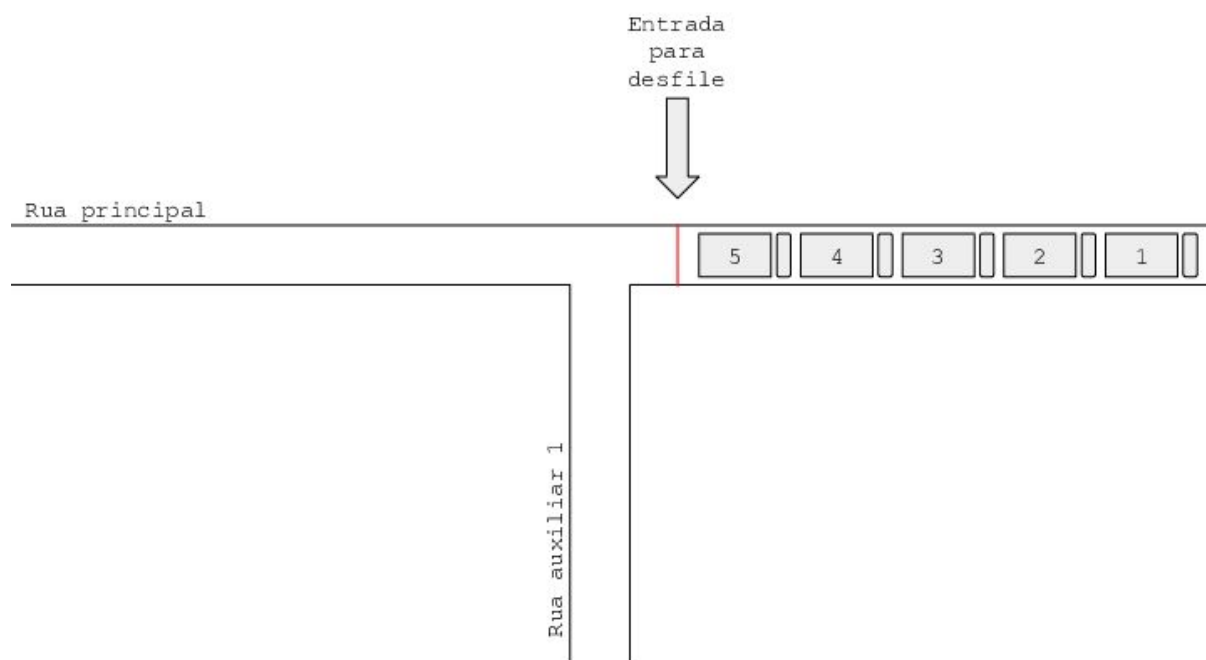
Alegoria 3 desfila.



Alegoria 4 sai da rua 1 e desfila.



Alegoria 5 sai da rua 1 e desfila.



## Exemplo 2

### Entrada

```
8 1
1 4 2 8 6 5 7 3
```

### Saída

```
Sem desfile.
```

## Exemplo 3

### Entrada

```
8 3
1 4 2 8 6 5 7 3
```

### Saída

```
Desfile pode ser realizado:
Alegoria 1 desfila.
Alegoria 4 entra na rua 1 e aguarda.
Alegoria 2 desfila.
Alegoria 8 entra na rua 2 e aguarda.
Alegoria 6 entra na rua 2 e aguarda.
Alegoria 5 entra na rua 2 e aguarda.
Alegoria 7 entra na rua 3 e aguarda.
Alegoria 3 desfila.
Alegoria 4 sai da rua 1 e desfila.
Alegoria 5 sai da rua 2 e desfila.
Alegoria 6 sai da rua 2 e desfila.
Alegoria 7 sai da rua 3 e desfila.
Alegoria 8 sai da rua 2 e desfila.
```



## Exemplo 4

### Entrada

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

### Saída

```
Desfile pode ser realizado:
Alegoria 1 desfila.
Alegoria 2 desfila.
Alegoria 10 entra na rua 1 e aguarda.
Alegoria 5 entra na rua 1 e aguarda.
Alegoria 4 entra na rua 1 e aguarda.
Alegoria 3 desfila.
Alegoria 4 sai da rua 1 e desfila.
Alegoria 5 sai da rua 1 e desfila.
Alegoria 9 entra na rua 1 e aguarda.
Alegoria 8 entra na rua 1 e aguarda.
Alegoria 7 entra na rua 1 e aguarda.
Alegoria 6 desfila.
Alegoria 7 sai da rua 1 e desfila.
Alegoria 8 sai da rua 1 e desfila.
Alegoria 9 sai da rua 1 e desfila.
Alegoria 10 sai da rua 1 e desfila.
```

## Exemplo 5

### Entrada

```
5 1
4 1 5 3 2
```

### Saída

```
Sem desfile.
```

# Observações

- Para as turmas A, B e C, esse laboratório tem peso 2.
- Deverão ser submetidos os arquivos `lab05.c`, `pilha.c`, `pilha.h`, devidamente implementados. Todas as funções responsáveis pela manipulação da estrutura do tipo pilha devem ser implementadas em `pilha.c`.
- Observações sobre SuSy:
  - Versão do GCC: 4.4.7 20120313 (Red Hat 4.4.7-17).
  - *Flags* de compilação:
    - `-ansi -Wall -pedantic-errors -Werror -g -lm`
  - Utilize comentários do tipo `/* comentário */;`  
comentários do tipo `//` serão tratados como erros pelo SuSy
  - Tempo máximo de execução: 1 segundo por teste.
- Além das observações acima, esse laboratório será avaliado pelos critérios:
  - Indentação de código.
  - Organização.
  - Corretude.