

# COMPARAR DUAS LISTAS LIGADAS

CompararLista.[ c | cpp | java | cs ]

Dado o ponteiro para o nó cabeça de duas listas ligadas compare se os dados nos nós das listas ligadas são iguais. As listas ligadas serão iguais se, e somente se tiverem a mesma quantidade de nós e tais nós possuírem as mesmas informações. Atenção: os ponteiros para o nó cabeça das listas ligadas podem ser nulos, o que significa que a lista está vazia.

Realize o exercício com a construção de um procedimento, conforme o seguinte cabeçalho:

```
int compareLists(struct Node *, struct Node *);
```

Que recebe como argumento dois ponteiros para o nó cabeça de cada uma das listas ligadas e retorne um inteiro, sendo 1 se as listas ligadas forem iguais ou 0 caso contrário.

## Entrada

O programa receberá como entrada vários pares de listas ligadas que devem ser comparadas.

A primeira linha da entrada consiste de um inteiro **Q**, representando a quantidade blocos de entrada que a entrada possui.

A seguir haverá uma sucessão de **Q** blocos com quatro linhas cada bloco.

Na primeira linha de cada bloco terá um inteiro **NA**, representando o tamanho da primeira lista ligada. E na segunda linha haverá uma sucessão de **NA** números inteiros separados por um espaço em branco cada, representando os **NA** elementos da primeira lista ligada. Atenção: Se **NA=0**, então a segunda linha do bloco não existirá.

A terceira linha do bloco terá um inteiro **NB**, representando o tamanho da segunda lista ligada. E na quarta linha haverá uma sucessão de **NB** números inteiros separados por um espaço em branco cada, representando os **NB** elementos da segunda lista ligada. Atenção: Se **NB=0**, então a quarta linha do bloco não existirá.

Restrições:

$$1 \leq Q < 100$$

$$0 \leq NA, NB < 10000$$

$$-2^{31} \leq LA_i, LB_j \leq 2^{31}-1, \text{ sendo: } 0 \leq i < NA \text{ e } 0 \leq j < NB$$

### Saída

Após ler cada bloco de entrada seu programa deve imprimir um número inteiro, sendo: 1 quando as listas ligadas forem iguais ou 0 caso contrário. Após a impressão de cada resultado, inclusive o último, salte uma linha.

### Exemplos

Entrada	Saída
4	1
4	0
1 2 3 4	0
4	0
1 2 3 4	
3	
1 2 3	
4	
1 2 3 4	
1	
12	
1	
13	
2	
4 15	
2	
15 4	