

REMOVER NÓS COM VALORES DUPLICADOS EM UMA LISTA LIGADA ORDENADA

`RemoveDuplicidadeLista.[c | cpp | java | cs]`

Dado o ponteiro para o nó cabeça de uma lista ligada ordenada de forma crescente. Remova os nós que contenham valores repetidos, de tal forma que na lista ligada resultante não tenha valores duplicados. Ao final imprima a lista ligada resultante no sentido do nó cabeça para o nó cauda.

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

```
struct Node * removeDuplicates(struct Node *);
```

Que recebe o nó cabeça da lista ligada ordenada e retorna o nó cabeça da lista ligada resultante, que também deve estar ordenada, porém sem as duplicidades.

Entrada

O programa receberá como entrada várias listas ligadas e você deverá tratá-las uma por vez.

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

A seguir haverá uma sucessão de **Q** blocos com até duas linhas cada bloco, representando cada uma das **Q** listas que você deverá tratar. Na primeira linha de cada bloco terá um inteiro **N**, representando o tamanho da lista ligada. E na segunda linha haverá uma sucessão de **N** números inteiros separados por um espaço em branco cada, representando os **N** elementos da lista ligada. Atenção: Se **N=0**, então a segunda linha do bloco não existirá.

Ao ler cada um dos **N** elementos, você deverá inseri-los na cauda da lista ligada.

Restrições:

$$1 \leq Q < 100$$

$$0 \leq N < 10000$$

$$-2^{31} \leq L_i \leq 2^{31}-1, \text{ sendo: } 0 \leq i < N$$

Saída

Após ler cada uma das listas ligadas e trata-las imprima todos nós da lista ligada resultante separados por um espaço em branco cada elemento. Após imprimir o último elemento de cada uma das listas ligadas resultantes, inclusive a última, quebre uma linha.

Exemplos

Entrada	Saída
4	1 2 3 4
6	1
1 2 2 3 3 4	2 3 4 6
7	10
1 1 1 1 1 1 1	
5	
2 3 3 4 6	
1	
10	