

[ED014] Recuperando árvores perdidas

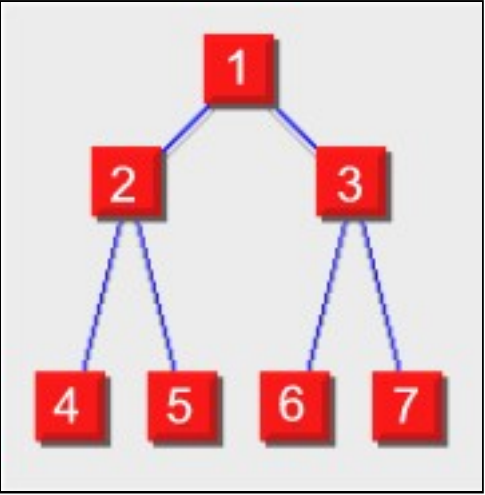
Deve incluir no mesmo ficheiro quaisquer classes que use (nenhum código base será adicionado)

O problema

Um antigo professor de uma cadeira de Estruturas de Dados e Algoritmos tinha muitos exemplos de árvores binárias guardadas para mostrar aos alunos. O que ele fazia era simplesmente guardar a representação **preorder** e **inorder** das árvores (não indicando árvores vazias). **Como usava sempre números diferentes, esta representação bastava para definir completamente a árvore !**

Recorda como escrever uma árvore em **preorder** e **inorder**

- **preorder:** escrever chave da raiz, seguida da representação preorder da subárvore esquerda, seguida da representação preorder da árvore direita
- **inorder:** escrever representação inorder da subárvore esquerda, seguida da chave da raiz, seguida da representação inorder da árvore direita



Por exemplo, para a árvore da figura anterior, temos as seguintes representações:

- **preorder:** 1 2 4 5 3 6 7
- **inorder:** 4 2 5 1 6 3 7

O novo professor da cadeira queria recuperar as árvores e precisa da tua ajuda. A tua tarefa é, dadas as duas reprensentações da árvore (preorder e inorder) reconstruir a árvore. Depois disso, basta mostrares a representação **postorder** da árvore para mostrares que realmente a árvore ficou bem construída. Como o novo professor está muito interessado em saber o número de nós terminais (ou folhas) da árvore, tens também de o indicar.

Recorda como escrever uma árvore em **postorder** ("4 5 2 6 7 3 1" para a árvore da figura)

- **postorder:** escrever representação postorder da subárvore esquerda, seguida da representação postorder da árvore direita, seguida da chave da raiz

Input

A primeira linha contém um número C, indicando o número de casos que se seguem.

Cada um dos casos é descrito por um conjunto de 3 linhas no seguinte formato:

- Uma primeira linha contendo N_NOS (N_NOS>0), indicando o número de nós da árvore
- Uma segunda linha com a representação preorder da árvore (números inteiros positivos separados por espaços)
- Uma terceira linha com a representação inorder da árvore (números inteiros positivos separados por espaços)

Deve ser notado que o programa não deve estabelecer limites para a quantidade de números a receber. Podes assumir que os valores dos nós cabem num *int*.

De notar também que não existem números repetidos numa mesma árvore.

Output

Para cada caso devem ser imprimidas duas linhas de output:

- Uma primeira linha com a representação postorder da árvore (números inteiros positivos separados por espaços)
- Uma segunda linha contendo "**Folhas = NUM**", onde **NUM** é o número de folhas da árvore

Vê o exemplo para clarificar a maneira como deve ser feito o output.

Exemplo de input/output

Input	Output
4	4 5 2 6 7 3 1
7	Folhas = 4
1 2 4 5 3 6 7	3 2 1
4 2 5 1 6 3 7	Folhas = 1
3	2 3 1
1 2 3	Folhas = 2
1 2 3	3 2 1
3	Folhas = 1
1 2 3	
2 1 3	
3	
1 2 3	
3 2 1	