

# Árvore AVL

Neste exercício você deve finalizar sua solução para árvores AVL!

- Você deverá elaborar todas as funções necessárias para inserir elementos na árvore AVL e fazer rotações (simples e duplas) para que a árvore resultante seja simplesmente balanceada.

Escreva um programa principal que cria várias árvores, efetuando rotações para mantê-las simplesmente balanceadas. Para cada árvore, imprima os resultados dos percursos pré-ordem e pós-ordem.

## Entrada e Saída

A entrada conterá vários casos de teste. Cada caso de teste é uma lista contendo uma sequência de chaves que devem ser inseridas em uma árvore inicialmente vazia. A entrada termina quando aparecer a lista vazia (ela não é um caso de teste).

Como saída, imprima primeiro a frase 'Arvore {}', sendo que o especificador '{}' deve ser substituído pelo número da árvore que está sendo processada. Em seguida, imprima os percursos pré-ordem e pós-ordem da árvore obtida após a operação de rotação. Siga a formatação contida nos exemplos. Imprima uma linha em branco após cada caso de teste.

## Exemplos de Entrada e Saída

Entrada	[20, 10, 40, 30, 50, 51] [20, 10, 40, 30, 50, 25] [40, 20, 50, 10, 30, 5] [40, 20, 50, 10, 30, 25] []
Saída	Arvore 1 pre: 40 20 10 30 50 51 pos: 10 30 20 51 50 40  Arvore 2 pre: 30 20 10 25 40 50 pos: 10 25 20 50 40 30  Arvore 3 pre: 20 10 5 40 30 50 pos: 5 10 30 50 40 20  Arvore 4 pre: 30 20 10 25 40 50 pos: 10 25 20 50 40 30
Entrada	[10, 30, 20, 5, 2, 40, 50] [30, 10, 20, 50, 40, 5, 15, 3, 8, 2, 18] []
Saída	Arvore 1

<pre>pre: 20 5 2 10 40 30 50 pos: 2 10 5 30 50 40 20</pre>
--

Arvore 2

```
pre: 10 5 3 2 8 20 15 18 40 30 50
pos: 2 3 8 5 18 15 30 50 40 20 10
```