

# Árvore Binária de Busca [5]

ArvoreBin5.[ c | cpp | java | cs | py ]

Faça um programa que implemente uma Árvore Binária de Busca conforme o padrão de entrada a seguir:

## Entrada

O programa receberá como entrada várias linhas no formato:

Formato do Comando	Descrição
<b>insert</b> <i>valor</i>	Inclusão do <i>valor</i> na Árvore
<b>delete</b> <i>valor</i>	Exclusão do <i>valor</i> na Árvore
<b>pre-order</b>	Mostra os itens da árvore em pré-ordem, um por linha.
<b>in-order</b>	Mostra os itens da árvore em ordem, um por linha.
<b>post-order</b>	Mostra os itens da árvore em pós-ordem, um por linha.

A entrada termina com sinal de final de arquivo (EOF).

Considere as seguintes restrições em sua implementação:

- *valor* será um número inteiro.
- Caso a operação tente incluir um *valor* que já existe na árvore, inclua-o na sub-árvore esquerda do nó com a mesma chave.
- Caso a operação tente excluir um *valor* inexistente na árvore, ignore a operação.
- Caso a operação de exclusão seja em um *valor* com várias ocorrências, exclua a que aparecer primeiro, ou seja, aquela de nível mais próximo da raiz, ou ainda aquela que estiver a uma maior altura na árvore.
- Caso a operação de exclusão recaia sobre um nó com dois filhos, eleja como substituto o sucessor.
- A cada operação de exclusão bem sucedida mostre *valor* na tela e salte uma linha.

## Saída

A saída do programa consistirá em várias linhas, onde em cada linha será a impressão dos valores no percurso solicitado e os valores das chaves excluídas com sucesso, a cada impressão solicitada, inclusive a última, salte uma linha.

## Exemplos

Entrada	Saída
insert 77	77
insert 70	70
insert 83	67
insert 67	73
insert 73	83
insert 80	80
insert 87	87
pre-order	65
insert 65	67
insert 90	70
insert 78	73
in-order	77
delete 80	78
delete 77	80
post-order	83
delete 67	87
delete 73	90
delete 80	80
in-order	77
delete 87	65
delete 78	67
post-order	73
	70
	90
	87
	83
	78
	67
	73
	65
	70
	78
	83
	87
	90
	87
	78
	65
	70
	90
	83