

## Problema B - Operações em Árvores Binárias de Pesquisa

Marcela recebeu como trabalho de Algoritmos a tarefa de fazer um programa que implemente uma Árvore Binária de Pesquisa (ou Busca). O Programa deve aceitar os seguintes comandos:

- I *n*: Insere na árvore binária de pesquisa o elemento *n*.
- INFIXA: lista os elementos já cadastrado segundo o percurso infixo
- PREFIXA: lista os elementos já cadastrado segundo o percurso prefixo
- POSFIXA: lista os elementos já cadastrado segundo o percurso posfixo
- P *n*: pesquisa se o elemento *n* existe ou não.

A qualquer momento pode-se inserir um elemento, visitar os elementos previamente inseridos na ordem infix, prefixa ou posfixa ou ainda procurar por um elemento na árvore para saber se o elemento existe ou não.

### Entrada

A entrada contém N operações utilizando letras (A-Z,a-z) sobre uma árvore binária de Busca, que inicialmente se encontra vazia. A primeira linha de entrada contém a inserção de algum elemento. As demais linhas de entrada podem conter qualquer um dos comandos descritos acima, conforme exemplo abaixo. O final da entrada é determinado pelo final de arquivo (EOF).

Obs: Considere que não serão inseridos elementos repetidos na árvore.

### Saída

Cada linha de entrada, com exceção das linhas que contém o comando "I", deve produzir uma linha de saída. A saída deve ser de acordo com o exemplo fornecido abaixo. Não deve haver espaço em branco após o último caractere de cada linha, caso contrário, sua submissão receberá Presentation Error.

Exemplo de entrada	Exemplo de saída
I c I f I a I h INFIXA PREFIXA POSFIXA P z P h	a c f h c a f h a h f c z nao existe h existe a c f g h

l g INFIXA	
---------------	--

Dica: use a função `strcmp(char A[], char B[])` que compara as strings A e B, e retorna 0 se forem iguais ou um número diferente de 0 se forem diferentes!