

1) Implemente um programa em C que:

- a) Defina uma estrutura (struct) para representar um nó de árvore binária, contendo:
 - i. um valor inteiro (int valor),
 - ii. um ponteiro para o filho esquerdo,
 - iii. um ponteiro para o filho direito.
 - b) Permita inserir novos valores seguindo as regras de uma árvore binária de busca:
 - iv. valores menores vão para a esquerda;
 - v. valores maiores vão para a direita.
 - c) No final, insira os valores 8, 2, 4, 10, 9, 13, 1, 6 e mostre a árvore em ordem (in-order).
-

2) Usando a mesma estrutura do exercício anterior, implemente funções recursivas que:

- imprimam os elementos em pré-ordem (raiz → esquerda → direita),
- em-ordem (esquerda → raiz → direita),
- pós-ordem (esquerda → direita → raiz).

Peça ao usuário para escolher o tipo de percurso e exiba os resultados conforme a escolha.

3) Implemente duas funções recursivas:

- int altura(No *raiz) → retorna a altura da árvore (número de arestas do maior caminho).

- `int peso(No *raiz) →` retorna o número total de nós da árvore.
-

4) Implemente uma função `No* buscar(No* raiz, int valor)` que:

- Retorna o endereço do nó que contém o valor, se encontrado;
- Retorna NULL caso o valor não exista na árvore.

No `main()`, permita ao usuário digitar um número e diga se ele foi encontrado ou não.

5) Implemente duas funções recursivas:

- `int contarFolhas(No* raiz) →` retorna o número de folhas (nós sem filhos);
- `int contarInternos(No* raiz) →` retorna o número de nós internos (nós que possuem pelo menos um filho).

Use a mesma árvore do exercício 1.