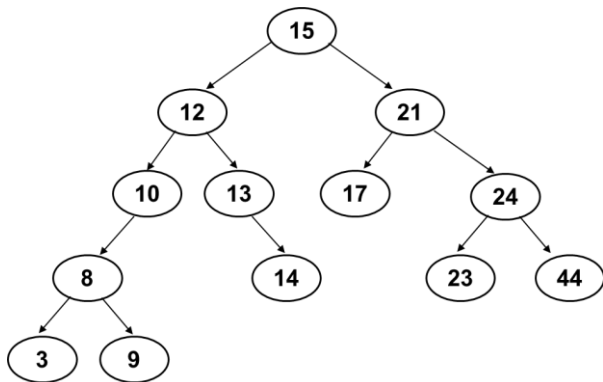


1. Considerando a árvore abaixo, responda:



- Qual é a altura da árvore;
- Qual é o nó raiz da árvore;
- Qual é o valor do pai do nó com valor 8;
- Quais são os valores dos nós ancestrais ao nó 23;
- Quais são os nós descendentes do nó 12;
- Qual é a altura da sub-árvore com nó raiz 10.
- Qual o resultado do caminhamento pré-ordem na árvore;
- Qual o resultado do caminhamento em-ordem na árvore;
- Qual o resultado do caminhamento pós-ordem na árvore;
- Qual o resultado do caminhamento em nível na árvore;

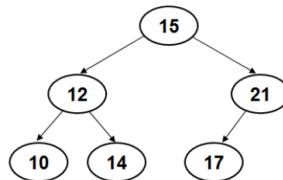
2. Desenhe as árvores binárias geradas pela inserção dos valores na ordem abaixo:

- 50,30,80,20,40,70,90,5,2,25,73,92
- 10,20,30,40,5,6,7,3,4,1,23,21,24,9,87

Para resolver as questões abaixo, considere a implementação de árvore binária disponibilizadas na aula teórica contendo funções para:

- Inserir elementos na árvore;
- Imprimir a árvore;
- Calcular a altura da árvore.

3. Na função main.c, crie a árvore abaixo utilizando as funções disponíveis.



- Implemente uma função que retorna o nó (ponteiro para o nó) do maior elemento da árvore.
- Implemente uma função que retorna o nó (ponteiro para o nó) do menor elemento da árvore.
- Implemente a função `int profundidade(Arvore *arv, int x)` que deverá retornar a profundidade do nó que armazena o valor "x" em uma árvore binária de pesquisa arv.
- Implemente a função `ArvoreBusca apagaFolha(int x, Arvore *arv)` que apaga o nó que armazena o valor "x" em uma árvore binária de pesquisa arv se ele for um nó folha.
- Implemente uma nova função que imprime o valor dos nós da árvore em pré-ordem.
- Implemente uma nova função que imprime o valor dos nós da árvore em em-ordem.
- Implemente uma nova função que imprime o valor dos nós da árvore em pós-ordem.
- Modifique a árvore dos exercícios anteriores para que a árvore armazene strings alocadas dinamicamente ao invés de números.