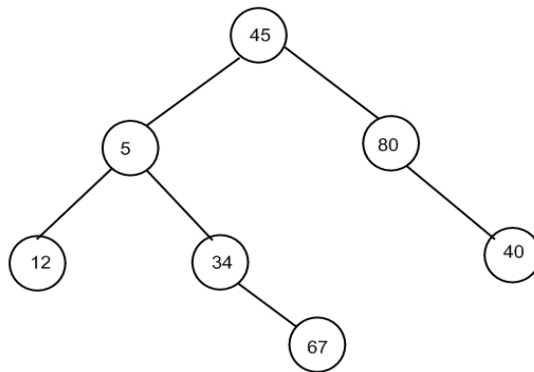


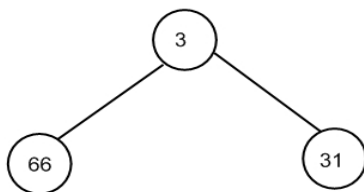
**DEVTITANS**  
**Básico de Estruturas de Dados e Algoritmos**

Entregue um código para cada problema abaixo

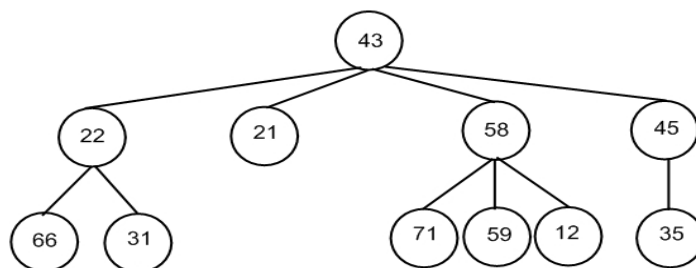
- 1) Baixe o arquivo **listas.c** e implemente a função "igual", que compara duas listas encadeadas e retorna 1 se forem iguais e 0 caso contrário.
- 2) Baixe o código **arvbin.c**. Implemente três funções para imprimir todos elementos da árvore usando os métodos pré-fixado, in-fixado e pós-fixado. Implemente uma função main que, usando as operações inicializa e cria, gere a seguinte árvore e depois imprima usando os três métodos.



- 3) Usando as funções cria e inicializa, enxertar a seguinte sub-árvore como filho esquerdo do nó 80. Imprimir a árvore resultante pelos três métodos. **Dica:** não precisa buscar o valor "80", mas é suficiente gerar a sub-árvore e fazer o enxerto usando, por exemplo, `a->dir->esq = a3;`



- 4) Usando a função libera, podar a sub-árvore que possui raiz com a informação 34. Imprimir a árvore resultante usando os três métodos.
- 5) Baixe o código **arvgen.c**. Implementar uma função main que, usando as operações cria e insere, gere a árvore da figura abaixo, imprima a árvore, e verificar se os elementos 12 e 88 estão presentes na árvore.



- 6) Baixe o código **arvbinbusca.c**. Implementar uma função que imprima a árvore in-fixada. Implementar uma função main que insira os seguintes elementos 1, 2, 3, 5, 15 e 7. Após cada inserção imprima a árvore. Buscar os elementos 2 e 8, imprimindo na tela se cada elemento foi encontrado ou não.