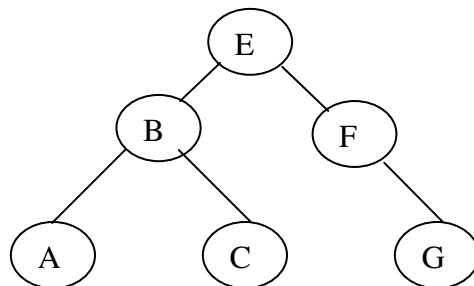


## PRACTICO N°3 – Arboles

1. Dadas las siguientes clases, construya una clase Tree, que implemente la interfase MyTree.  
Dicha clase Tree debe comportarse como un árbol binario.  
Implemente el método find utilizando un loop y de manera recursiva.

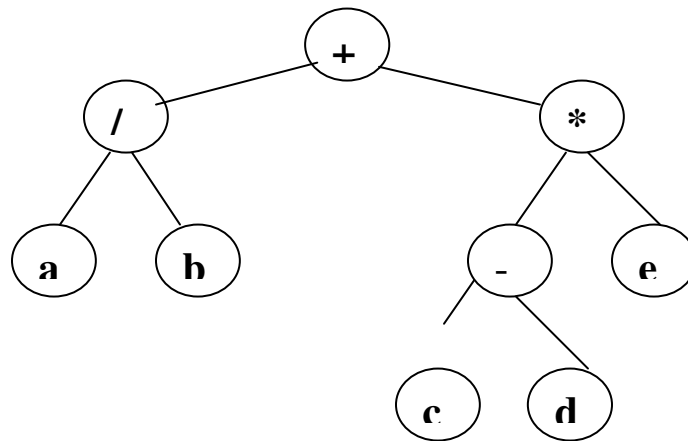
```
public class Node {  
  
    int key;  
    double data;  
    Node leftChild;  
    Node rightChild;  
    .....  
  
}  
  
public interface MyTree {  
  
    public Node find(int key);  
    public void insert (int id, double idata);  
    public void delete (int id);  
  
}
```

2. Se considera el *árbol binario* de la figura:



Si se utiliza un algoritmo “en-orden” como los vistos en el curso para imprimir las claves de los nodos a pantalla. ¿Cuál sería la salida a pantalla del árbol de la figura, donde los valores de la claves son los strings dibujados en cada nodo?

3. Escriba métodos recursivos que calculen para un *árbol binario*:
  - a. El número de nodos de un árbol
  - b. El número de hojas de un árbol
  - c. El número de nodos con subárboles izquierdo y derecho no nulos
4. Implemente en java el algoritmo para recorrer por nivel un árbol visto en el teórico.
5. Construir una aplicación que lea una expresión en notación postfija, construya un árbol binario de la expresión y luego genere una salida con la expresión en notación infija. Las únicas Operaciones válidas son +, -, \*, /. Se supone que la expresión Postfija está dada con la sintaxis correcta.



6. Se considera un *árbol binario ordenado* con la siguiente estructura

```
Class arbol{  
    int key;  
    arbol left;  
    arbol right;  
    ...  
}
```

- a. Escribir un algoritmo recursivo de esta clase que devuelva la suma de todos los valores enteros almacenados en los atributos key del arbol, si la key es mayor que un entero pasado como parámetro.
  - i. public int SumarKey(int dato)
- b. Escribir un algoritmo recursivo de esta clase que devuelva verdadero si existe algún atributo key del árbol que se

encuentre entre dos enteros pasados como parámetro y falso en caso que ninguna key se encuentre en ese rango

i. public boolean BuscarKeys(int min, int max)

7. Escriba un método recursivo en un *árbol binario de búsqueda* que encuentre:
  - a. La clave mayor
  - b. La clave menor
  - c. La menor clave mayor que un valor dado
  
8. Escriba un método recursivo que imprima a pantalla todas las claves de un *árbol binario de búsqueda* tales que se son mayores o iguales que un valor k1 y menores iguales que k2. Los valores k1 y k2 son pasados como parámetros al método.
  
9. Escriba un método que imprima la k-esima clave más pequeña de un *árbol binario de búsqueda*.