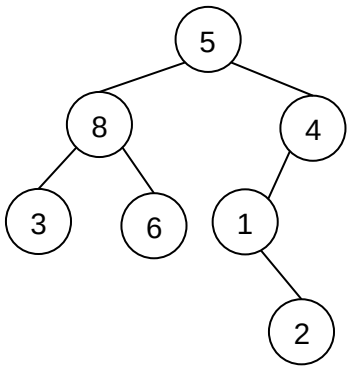
Práctica 7 – Árboles

1. Dado el siguiente árbol:



   a) Mostrar (sin desarrollar un programa):
      1. el contenido de sus nodos según cada recorrido: *preorden, inorden*, *postorden*.
      2. el contenido del nodo raíz, y para cada nodo indicar: grado, nivel, valor del padre y del hermano.
   b) Responder: ¿cuál es la profundidad del árbol?

2. Diagramar el árbol correspondiente a las siguientes expresiones aritméticas y luego, sin desarrollar un programa, recorrerlo en postorden mostrando su contenido.
   a) 5 + 8 * 3 – 6 *3
   b) 6 * (5 + 2) / (6 - 2)

3. Desarrollar funciones para:
   a) devolver la suma de los elementos múltiplos de 3 de un árbol binario.
   b) retornar la cantidad de hojas de un árbol binario.
   c) informar si un valor recibido como parámetro se encuentra en un árbol binario.

4. Corregir los errores en la siguiente función void, la cual genera un arreglo de caracteres con los nodos de grado 2 de un árbol binario.

```
void arrgrado2(arbol a, char v[N], int dimv) {
  if (a != NULL) {
    if(a->izq != NULL && a->der == NULL) {/* grado 2 */
       (*dimv)++;
       v[dimv] = a->dato;
    } else
       arrgrado2(a->izq, v[], dimv);
       arrgrado2(a->der, v[], dimv);

  }
}
```

5. Desarrollar funciones para:
   a) calcular la profundidad de un árbol binario.
   b) retornar la longitud de la cadena más larga de un árbol binario de cadenas.
   c) devolver la cantidad de hijos derechos que contiene un árbol binario

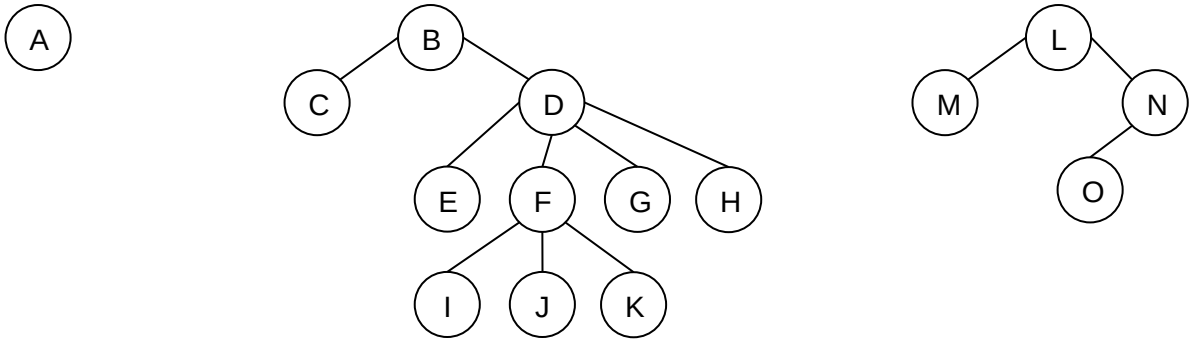
6. Dado un árbol binario de cadenas de caracteres retornar el nivel en el que se encuentra la cadena más larga que comienza con A

7. Desarrollar una función que retorne la suma del contenido de aquellos nodos de un árbol binario que tengan grado 1 y se encuentren en un nivel X que es dato.

8. El draw de un torneo de tenis se representa mediante un árbol binario invertido. Desarrollar subprogramas que muestren:
   a) el nombre del ganador.
   b) los nombres de los 2 finalistas.
   c) los nombres de los 4 semifinalistas.
   d) los nombres de todos los competidores y la cantidad.
9. Dibujar todos los ABB posibles para 3 elementos dados. Ej: 10, 15 y 20.
10. Corregir la siguiente función, la cual retorna si un valor recibido como parámetro se encuentra en un ABB cuyas claves son enteros.

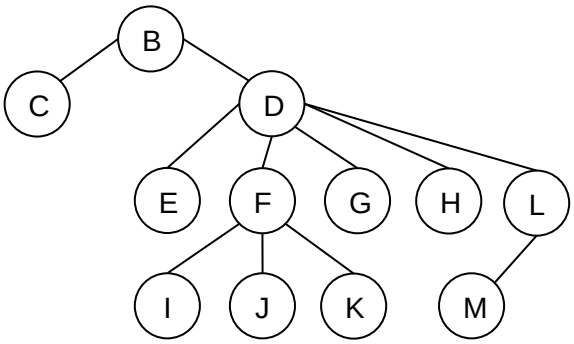
```
int busca(arbol a, int* x) {
  if (a != NULL)
     return 0;
  else
     if(x = a.dato)
         return 1;
     else
       if(x < a.dato)
          return busca(a.der, *x);
       else
          return busca(a->izq, *x);
}
```

11. Retornar el valor mínimo de un ABB.
12. Determinar cuántos elementos de un ABB son mayores que A y menores que B. A y B son parámetros de entrada.
13. Generar un arreglo de caracteres con el contenido de las claves de un ABB. Los elementos del arreglo deben quedar ordenados descendentemente.
14. Desarrollar una función que compruebe que si un árbol binario es un ABB.
15. Graficar la inserción de los siguientes valores, en el orden dado, en un ABB inicialmente vacío: 10, 8, 14, 24, 11, 1, 33, 40, 5, 32, 3, 7, 9 y 25. Eliminar 7, 24, 8, 10.
16. Implementar una función iterativa que inserte un elemento en un ABB.
17. En cada paso de inserción/eliminación del ejercicio 15, determinar si el árbol resultante es AVL (mediante el cálculo del f.e.).
18. Transformar gráficamente el siguiente bosque en un árbol binario:



19. Dado un árbol binario que proviene de la transformación de un bosque, determinar qué cantidad de árboles lo componían.
20. Dado un árbol binario proveniente de la conversión de un árbol general, hallar la cantidad de nodos que habia en niveles impares.

21. Dado un árbol binario proveniente de la conversión de un árbol general, obtener el promedio de las claves cuyo grado era K (dato de entrada).
22. Dado un árbol binario que proviene de la transformación de un árbol general, obtener la altura del árbol original
23. Dado un árbol binario proveniente de la conversión de un árbol general, determinar el grado de este último.
24. Dado un árbol binario que proviene de la transformación de un bosque, hallar la cantidad de árboles del bosque que tenían altura al menos K (dato de entrada)
25. Dado un árbol binario de números naturales que proviene de la transformación de un bosque, devolver un arreglo con la clave mayor de cada uno de los árboles que conforman el bosque.
26. Dado un árbol binario de números naturales que proviene de la transformación de un bosque, verificar que todos los árboles contengan al menos un nodo de grado K (dato de entrada).
27. Dado el siguiente árbol N-ario:



a) Mostrar (sin desarrollar un programa) el contenido de sus nodos según cada recorrido: *preorden, inorden*, *postorden*.
b) ¿cuál es el grado del árbol? ¿Y su profundidad?

28. Dado un árbol N-ario de enteros, desarrollar funciones utilizando TDA N.ARIO para:
    a) retornar la cantidad de nodos que posee.
    b) hallar el porcentaje de claves pares
    c) retornar su grado
    d) hallar la cantidad de nodos de grado impar que hay en niveles impares.
    e) verificar si cumple que para todas las claves salvo las de las hojas, su valor numérico es igual a la cantidad de hijos. (función int y función void)
    f) obtener el promedio de las claves del nivel K (dato)