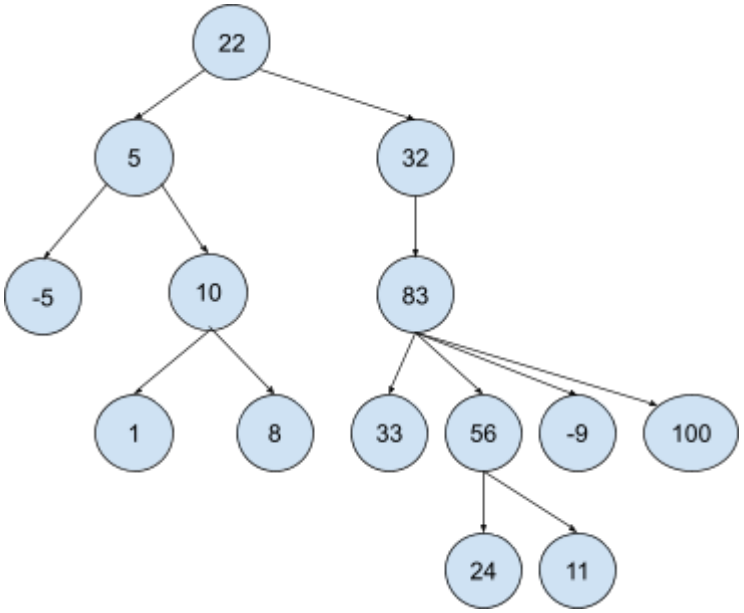
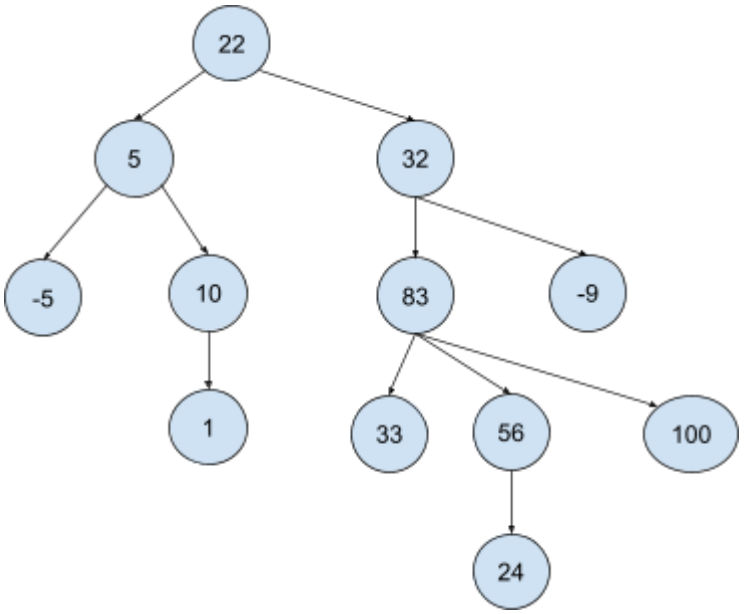
Ejercicio

Implemente en la clase **Parcial** el método:

> ListaGenerica<ArbolGeneral<Integer>>**resolver** (ArbolGeneral<Integer> arbol):int
> que devuelva <u>todos</u> los subárboles que tienen un número par de hijos y no son hojas. Realice un **recorrido en inorden**.

Ejemplo:





Para el árbol dado, se debería retornar una lista con los subárboles cuyos datos en la raíz son:
75, 43, 7, 8, 30 (Resultado obtenido al recorrer el árbol pasado como parámetro en inorden).

Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere
- Todo aquel método que no esté definido en las prácticas debe ser implementado
- Respetar el recorrido solicitado

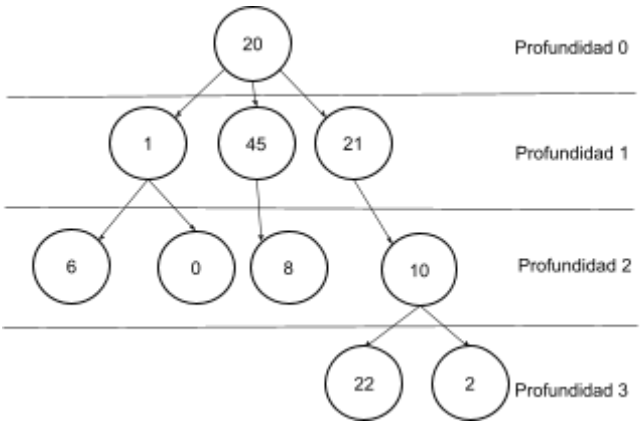
...

Ejercicio

Implemente en la clase **Parcial** un método que reciba un árbol general de enteros. Se debe retornar el producto de las hojas más profundas.
El método debe tener la siguiente firma:

resolver(ArbolGeneral<Integer> arbol):int

Para el siguiente árbol, hay 5 hojas, el nodo 6, el nodo 0, el nodo 8, el nodo 22 y el nodo 2.
El nodo 6, el nodo 0 y el nodo 8 están a profundidad 2, el nodo 22 y el nodo 2 está a profundidad 3. En este caso el método **resolver** debería devolver 44 (del producto de 22*2).



Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.

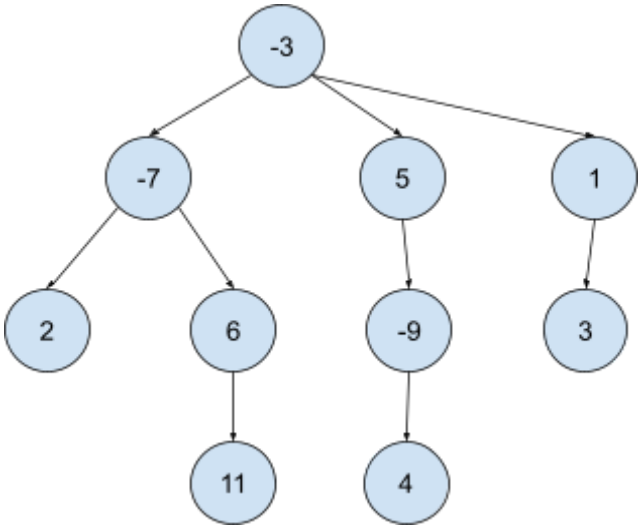
Ejercicio

Implemente en la clase **Parcial** un método que reciba un árbol general de enteros y calcule la suma de todos los nodos. Luego, si la misma es par el método debe devolver la suma de los elementos positivos. En caso contrario, debe devolver la suma de los elementos negativos. Realice un **recorrido en postorden.**

El método debe tener la siguiente firma:

resolver(ArbolGeneral<Integer> arbol):Integer

Para el siguiente árbol, la suma total en recorrido postorden es 2 + 11 + 6 + (-7) + 4 + (-9) + 5 + 3 + 1 + (-3) = 13 un número impar, por lo cual, debe devolver la suma de los números negativos: (-7) + (-9) + (-3)= -19



Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.

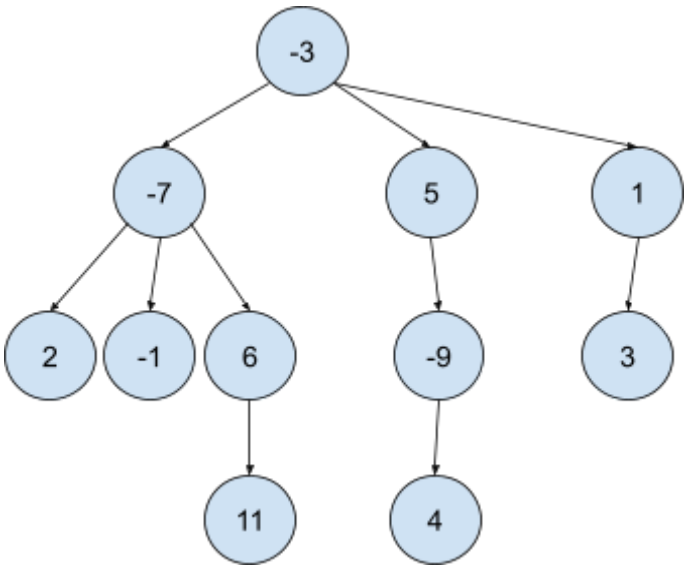
Ejercicio

Implemente en la clase **Parcial** un método que reciba un árbol general de enteros y calcule la cantidad de nodos. Luego, si la misma es par, el método debe devolver la cantidad de elementos positivos. En caso contrario, debe devolver la cantidad de elementos negativos. Realice un **recorrido en inorden.**

El método debe tener la siguiente firma:

resolver(ArbolGeneral<Integer> arbol):Integer

Para el siguiente árbol, la cantidad total es 11, un número impar, por lo cual, debe contar los números negativos, 4.



Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.

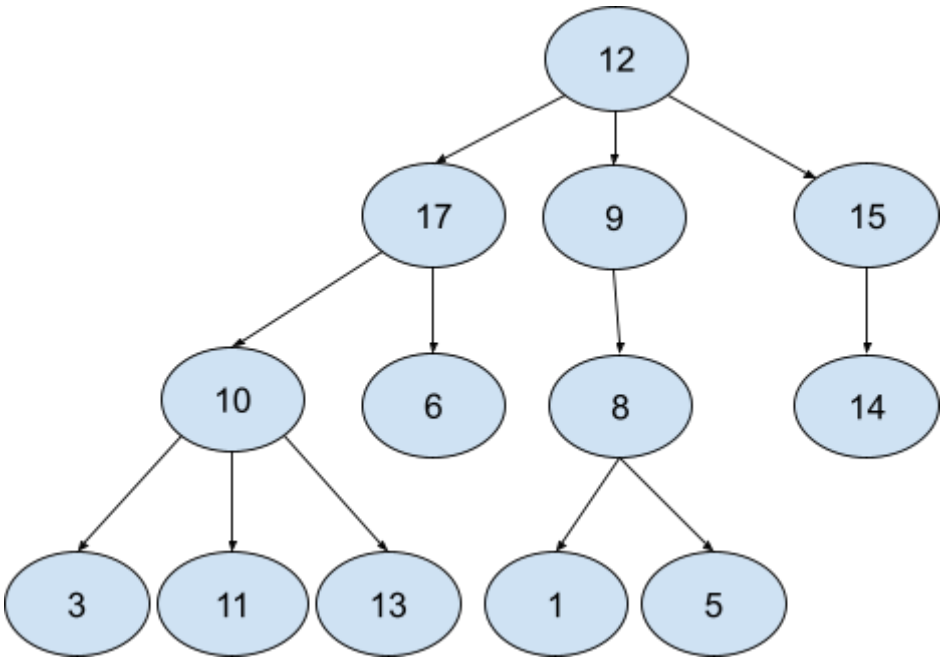
Ejercicio

Implementar en la clase Parcial el método:

resolver (ArbolGeneral<Integer> arbol, Integer min, Integer max): Integer

El cual retorna el promedio de los elementos del arbol, solo considerando aquellos que se encuentran en el rango min y max indicado. Realice un **recorrido en inorden**.

Dado el siguiente árbol general, considerando min=6 y max=12, los elementos a considerar son:
10, 11, 6, 12, 8, 9 (En inorden). La suma da 56. Y dado que son 6 números, el promedio es 9.



Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.

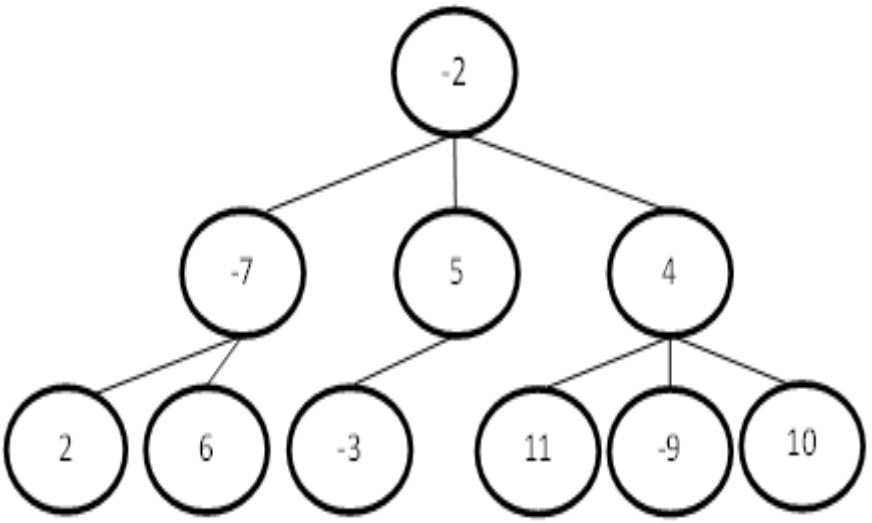
Ejercicio

Implemente en la clase **Parcial** un método que reciba un **árbol general de enteros** y retorne una **lista con los valores mayores a un valor** recibido por parámetro. Para cada valor, retornar el nivel en el que se encuentra. Considerar que el árbol podría estar vacio, que ningún elemento del árbol o todos los elementos del árbol podrían cumplir la condición. Realice un **recorrido inorden**.

El método debe tener la siguiente firma:

resolver(int valor, ArbolGeneral<Integer> arbol)

Para el siguiente árbol, si valor es 5, la lista debería contener los elementos:
6 nivel 2, 11 nivel 2, 10 nivel 2. (En **inorden**).
Para el siguiente árbol, si valor es 11, la lista debería ser vacía.
Para el siguiente árbol, si valor es -10, la lista debería contener todos los elementos del árbol:
2 nivel 2, -7 nivel 1, 6 nivel 2, -2 nivel 0, -3 nivel 2, 5 nivel 1, 11 nivel 2, 4 nivel 1, -9 nivel 2, 10 nivel 2 (En **Inorden**).



Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.

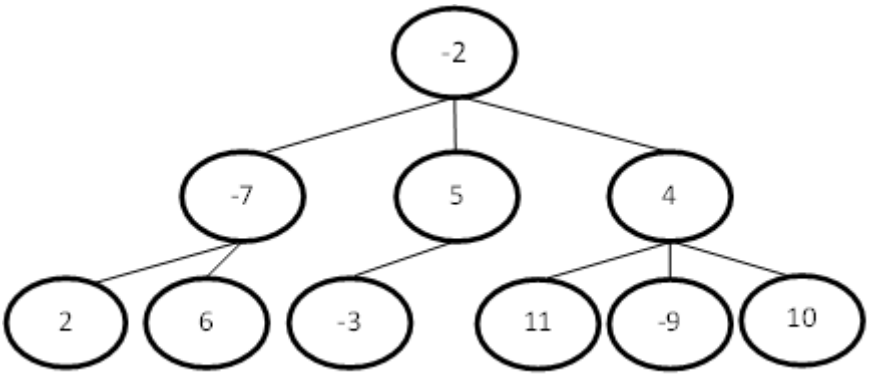
Ejercicio

Implemente en la clase **Parcial** un método que reciba un **árbol general de enteros** y retorne una lista con los valores que se encuentran en un rango de valores recibido por parámetro. Para cada valor, retornar el nivel en el que se encuentra.
El rango de valores es cerrado, por lo cual, si el elemento del árbol es igual a uno de los extremos, también se lo debe incluir en el resultado. Considerar que el árbol podría estar vacío, que ningún elemento del árbol o todos los elementos del árbol podrían cumplir la condición**. Realice un recorrido en postorden**.

El método debe tener la siguiente firma, por lo cual, siempre debe retornar una lista, ya sea que haya o no elementos para retornar.

resolver(int menor, int mayor, ArbolGeneral<Integer> arbol)

Para el siguiente árbol, si los valores son 5 y 10, la lista debería contener los elementos 6 nivel 2, 5 nivel 1, 10 nivel 2. (En **postorden**).
Para el siguiente árbol, si valor es 20 y 100, la lista debería ser vacía.
Para el siguiente árbol, si valor es -9 y 20, la lista debería contener todos los elementos del árbol:
2 nivel 2, -7 nivel 1, 6 nivel 2, -2 nivel 0, -3 nivel 2, 5 nivel 1, 11 nivel 2, 4 nivel 1, -9 nivel 2, 10 nivel 2. (En **postorden**).



Debe respetar la clase y el método indicado.
- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.

Ejercicio

Implemente en la clase **Parcial** que tiene como variable de instancia un **ArbolGeneral<Integer>** ,el método:

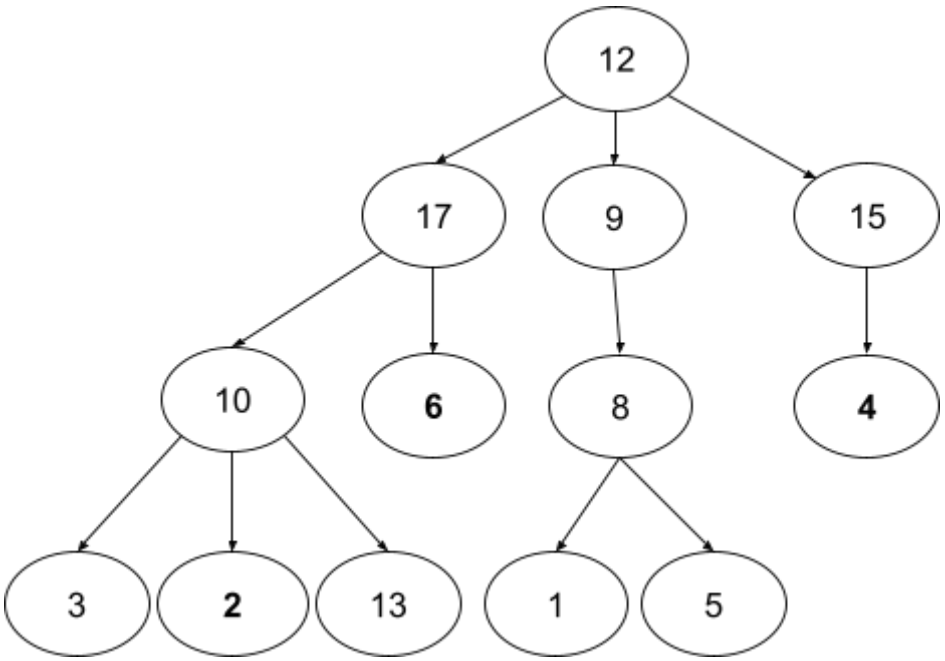
ListaEnlazadaGenerica<Integer> **resolver()**

Que retorna una lista con los elementos de la **frontera** del árbol con valor par.

Se define como **frontera** de un árbol general, a las hojas del árbol recorridas de izquierda a derecha.

Por Ejemplo, para el siguiente árbol

- **resolver() devuelve una lista de enteros con 2, 6 y 4.**



Debe respetar la clase y el método indicado.

- Puede definir todos los métodos y variables auxiliares que considere.
- Todo aquel método que no esté definido en las prácticas debe ser implementado.
- Respetar el recorrido solicitado.