



Nombre y Apellido:

Legajo:

Examen Parcial

Tiempo mínimo para el examen: 1 hora reloj.

Tiempo máximo para el examen: 2 horas reloj.

Práctica

1. Se necesita desarrollar un programa para gestionar una lista de estudiantes de una escuela. Se desea implementar una clase llamada **Estudiante** que tenga los métodos **obtener_legajo** y **obtener_nombre** para obtener el legajo y la edad del estudiante, respectivamente.

Además, se debe implementar una clase llamada **Escuela** con los siguientes métodos:

- A. **agregar_estudiante**: recibe un objeto **Estudiante** y lo agrega a la lista de estudiantes de la escuela.
- B. **eliminar_estudiante**: recibe el legajo de un estudiante y lo elimina de la lista de estudiantes de la escuela.
- C. **buscar_estudiante**: recibe el legajo de un estudiante y devuelve **True** si el estudiante está en la lista de la escuela, **False** en caso contrario.

Nota Consideramos que el legajo es un identificador único dentro de cada escuela.

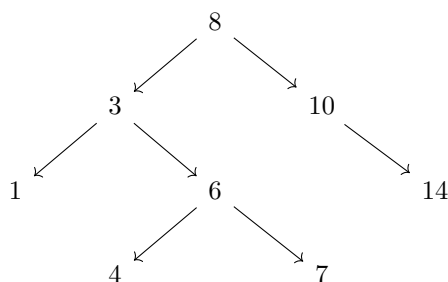
2. Implemente una función que reciba una lista enlazada L que contiene números enteros y un número entero k y devuelva una nueva lista enlazada en la que para cada posición i de dicha lista enlazada, contenga el resultado de la suma de i y los $k - 1$ elementos anteriores a i . En las primeras $k - 1$ posiciones, devolver la suma parcial hasta dicho momento. La función debe ser pura, es decir, no debe modificar la lista recibida por parámetro.

Por ejemplo, si se llama a la función con la lista $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ y $k = 3$, el resultado debería ser $1 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 12$.

Nota: Puede considerar que están definidos todos los métodos de la interfaz de Lista (**__str__**, **__len__**, **append**, **insert**, **remove**, **pop** e **index**).

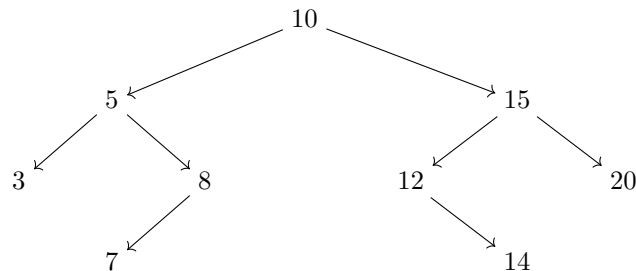
3. Implemente una función que reciba un árbol binario A y que dado un valor entero M , una clave inicial **inicio** y una clave final **fin**, se devuelva una lista con todos los datos cuyas claves estén entre **inicio** y **fin**, que estén dentro de los primeros M niveles del árbol (considerando a la raíz en nivel 0).

Por ejemplo, dado el siguiente árbol, **inicio** = 5, **fin** = 15 y $M = 2$, debería dar la lista [6, 8, 10, 14] (1 y 3 quedan afuera porque son menores que **inicio**, 4 y 7 quedan afuera porque están en el nivel 3).



Ayuda Defina la función de manera recursiva e incorpore un parámetro para control del nivel actual.

4. Dado el siguiente árbol binario.



1. Si se trata de un árbol binario de búsqueda, inserte el valor 13 en el mismo. Si no se trata de un árbol binario de búsqueda, justifique porqué.
2. Muestre el recorrido pre-orden del mismo

Teoría

5. Explique que es una clase, un objeto, un método y un atributo.
6. El TAD Deque (también llamado Cola De Doble Final) es un TAD que se comporta como una pila y como una cola a la vez. Formalmente, esta definido por la siguiente interfaz:

- `__len__` que devuelve cuantos elementos contiene la deque.
- `insertar_al_principio(x)` para agregar un elemento al principio de la deque.
- `obtener_al_inicio()` para obtener el elemento en la posición inicial.
- `insertar_al_final(x)` para agregar un elemento al final de la deque.
- `obtener_al_final()` para obtener el elemento en la posición final.
- `__init__()` crea una nueva deque vacía.
- `is_empty()` chequea si la deque esta vacía.

Explique como implementaría este TAD de forma que todas las operaciones se ejecuten eficientemente, es decir, con complejidad temporal constante

7. Explique que es el caso base y el caso recursivo en una definición recursiva.