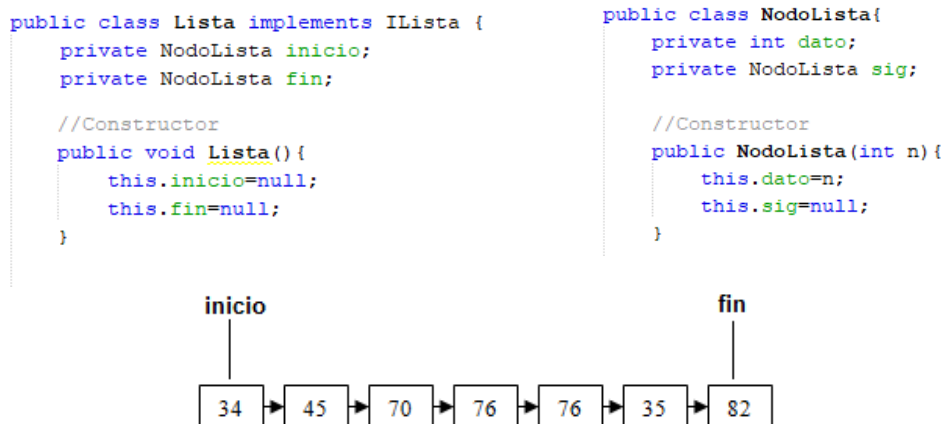


EVALUACIÓN	Parcial	GRUPO	N3C	FECHA	27/06/2022
MATERIA	Aed1				
CARRERA					
CONDICIONES	<p>El parcial será entregado en los librillos proporcionadas, con letra clara, entendible.</p> <p>IMPORTANTE:</p> <ul style="list-style-type: none"> - Duración 2 hs. - Sin material <p>Puntaje Mínimo: 1 Puntaje Máximo: 45</p>				
Docente con el que curso	Estudiante NRO	Estudiante Nombre	Nota		

Ejercicio 1 (10 ptos)

A partir de una implementación de una lista simplemente enlazada, como se muestra a continuación



Se dispone de las siguientes funciones de la Clase Lista: Inicio(), getInicio() , agregarInicio()
Y de la clase NodoLista: NodoLista (int n), getSig(), getDato()

Se solicita implementar:

- Realizar un algoritmo que reciba una Lista como parámetro y la imprima por consola en orden inverso.
Ejemplo: Para la lista que se muestra en figura, imprimiría: 82,35,76,76,70,45,34
Firma: **public static void mostrarAlVerre(Lista l) (5 ptos)**

Se solicita implementar, detallando pre y post condiciones, uno de los dos siguientes algoritmos a elección:

- b. Realizar un algoritmo que, dada una lista y dado un entero, cuente todos los nodos con números menores e iguales al mismo. Ej: para la lista dispuesta como ejemplo, si el número dado fuera 76, se debería retornar: 6.

Firma: **int contarMenorEl igual(Lista lista, int num) (5 ptos)**

- c. Realizar un algoritmo recursivo que, dado un entero dado como parámetro, indique si el mismo existe o no en la lista,

Firma: **boolean existe(NodoLista nodo, int num) (5 ptos)**

Ejercicio 2 (15 ptos)

Dado el siguiente vector ordenado:

10	14	19	25	30	45	49	51
----	----	----	----	----	----	----	----

Se solicita

- a. implementar un método recursivo que, dado un entero que indica la posición tope del array hasta la que hay que sumar, retorne la suma desde la posición 0 a esa posición tope. Para el ejemplo del vector dispuesto a continuación con tope 3, el retorno debería ser: 68

Firma: **int suma(int vec[], int tope) (10 ptos)**

- b. Realizar el diagrama de llamadas para el array dispuesto anteriormente con tope 4. **(5 ptos)**

Ejercicio 3 (10 ptos)

Dada una pila desordenada de elementos y un dato entero, implementar un método que permita eliminar todas las ocurrencias de ese entero en la pila.

Firma: **void eliminarEnPila(Pila p, int valor) (10 ptos)**

Se disponen de las siguientes operaciones de Pila:

new Pila(), esVacia(), cima(), apilar(), desapilar()

Es posible crear estructuras adicionales (Lista, Cola, etc.) disponiendo de sus operaciones si fuera necesario.

Pila antes	Pila después de borrar elemento valor = 4
4	
4	8
8	105
105	45
4	28
45	5
4	
28	
4	
5	
4	

Ejercicio 4 (10 pts)

Matriz Original:

```
int mat[][] = {{0,0,0,0,0,0,0},
               {0,0,0,0,0,0,0},
               {0,0,0,0,0,0,0},
               {0,0,0,0,0,0,0}}
```

Matriz que devuelve la función:

```
int mat[][] = {{1,1,1,1,1,1,1},
               {1,0,0,0,0,0,1},
               {1,0,0,0,0,0,1},
               {1,1,1,1,1,1,1}}
```

Realizar un algoritmo que, dado una matriz de enteros inicializada en 0 (cero), retorne otra matriz con los bordes completados con el valor 1.

Firma: `int[][] cuadroMatriz(int mat[][])`