

EVALUACIÓN	Parcial	GRUPO	N3D	FECHA	28/6/22
MATERIA	Aed1				
CARRERA	AP-ATI				
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 pts)

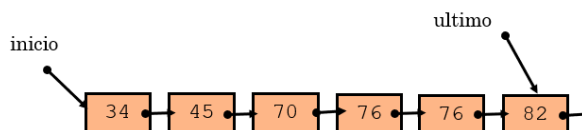
A partir de una implementación de una lista simplemente enlazada - ordenada en forma ascendente - como se muestra a continuación:

```
Public class Lista implements ILista {
    Private Nodo inicio;
```

```
// Constructor
Public Lista() {
    Inicio=null;
}
```

```
public class Nodo {
    private int dato;
    private Nodo sig;
```

```
//Constructor
public Nodo(int elDato){
    this.setDato(elDato);
    this.setSig(null);
}
```



Se solicita implementar:

- Un algoritmo que, dado un nodo inicial de la lista y un dato n, lo elimine de la lista y retorne un booleano según lo haya encontrado y eliminado. Ej: para la lista dispuesta como ejemplo, si se diera el entero 76, la lista resultante sería: 34-45-70-76-82 y el retorno es true.
Firma: boolean eliminar(Nodo nodoLista, int num) **(5 pts)**
- Un algoritmo recursivo que, dado un nodo inicial de la lista y un dato entero, retorne la cantidad de elementos mayores al dato dado. Ej: para la lista dispuesta como ejemplo, si el número dado fuera 76, se debería retornar 2:
Firma: int contarMayores(int num) **(5 pts)**

Ejercicio 2 (10 pts)

Dada la siguiente pila de ejemplo y suponiendo disponibles los métodos `new Pila()`, `esVacia()`, `cima()`, `apilar()`, `desapilar()`:

10
20
12
78
40

Escribir un algoritmo que, dado una pila, un nivel inferior y un nivel superior, elimine todos los elementos que se encuentran entre los niveles indicados (inclusive). Se considera que el nivel 1 es donde se encuentra el elemento tope de la pila. Firma: `void eliminarElementos(Pila p, int nivelInf, int nivelSup)`

Ej: para la pila dispuesta inicialmente, con nivel inferior: 2 y nivel superior: 4, la pila resultante debería ser:

10
40

Se podrán crear Pilas, Listas y Colas alternativas para su implementación, suponiendo disponible los métodos vistos en clase.

Ejercicio 3 (10 pts)

Realizar un algoritmo que, dado una matriz de enteros, imprima la suma de filas impares.

Firma: `void sumarFilasImpares(int mat[][])`

Ejercicio 4 (15 pts)

Dado el siguiente vector ordenado en forma ascendente,

- a) Implementar un algoritmo recursivo de búsqueda por bipartición utilizando la técnica divide y vencerás.
(10 puntos)

Vector v=

1	5	34	45	70	80	85	100
---	---	----	----	----	----	----	-----

Pre condición: el vector está ordenado, inicialmente parte con `desde=0`, `hasta=v.length-1`.

Firma: `private static boolean algoritmo(int[] v, int desde, int hasta, int num)`

- b) Realizar el diagrama de llamadas para Llamada inicial: `algoritmo(v,0,7,70)` (5 puntos)