

1)

Implementar un método que dada una matriz de nxm devuelva un vector que contiene el máximo de cada columna.

Ej:

Matriz

5	1	8	4
2	9	7	1
1	4	3	5

Resultado

5	9	8	5
---	---	---	---

2)

Implementar un método que reciba una matriz (nxm) y devuelva una pila que contiene la suma de cada fila de la matriz (respetar el orden de impresión, en el tope de la pila debe estar la suma de la primera fila)

5	1	8	4	18
2	9	7	1	19
1	4	3	5	13

3)

Implemente en un método que, dada una lista y un elemento, en caso de encontrarse, elimine ese elemento y el anterior.

Se borra la primera ocurrencia en caso de estar más de una vez.

Se debe implementar este método dentro del TAD

Si no hay un elemento anterior para borrar, solamente elimina el encontrado.

*public void eliminar2Elementos(int elemento)*

Ejemplos:

5->7->3->6->9  
(eliminar 3)  
5->6->9  
5->7->3->6->9  
(eliminar 5)  
7->3->6->9

## ROMS

### Ejercicio 1 (10 pts)

Escriba un algoritmo que, dado una matriz de enteros de largo  $m \times n$  indique (mediante un valor booleano) si las diagonales de la matriz suman lo mismo, como se muestra en el ejemplo.

28	3	62	18
15	39	47	5
8	57	45	6
25	16	71	35

Dada la siguiente lista simplemente enlazada y ordenada:



- a) Implemente un algoritmo recursivo que muestre los últimos  $n$  elementos impares de la lista

**Firma sugerida: mostrarUltimosNImpares (Lista L, int n)**

Ejemplo: para la lista anterior y un  $n = 2$ , el resultado sería: 69 43

- b) Realice el diagrama de llamadas para el ejemplo anterior

### Ejercicio 4 (20 pts)

Escribir una función que permita eliminar de una pila todos los elementos mayores a un valor indicado:

Firma a utilizar: **boolean eliminarMayores(pila P, int valor)**

### Ejercicio 2

Implemente un método de búsqueda binaria visto en el curso

Firma: public boolean buscarB (int vec[], int inf, int sup, int n)

#### Ejercicio 4 (10 pts)

Realizar un algoritmo que reciba una matriz y un número, y retorne una pila donde se encuentren los índices de aquellas columnas de la matriz donde se encuentra el número dado al menos 2 veces. Los índices en la pila deben estar ordenados de forma ascendente como se muestra en la figura.

Sugerencia de firma: `public Pila pilaDeIndices(int [][] mat, int num);`

Ejemplo:

Matriz				Pila	
23	1	3	32	1	
11	32	1	32	3	
32	32	43	32		

Explicación: El 32 se encuentra 2 veces en la columna de índice 1 y 3 veces en la columna de índice 3.

Se asumen disponible los métodos de Pila: `tope()`, `push()`, `pop()`, `esVacia()`, `esLlena()`. En caso de utilizar algún otro TAD, se podrán asumir disponibles las operaciones vistas durante el curso.

- a) Implemente un método en la clase Lista (implementada con un NodoLista simplemente encadenado) que reciba un número `pos1` y otro número `pos2` y borre todos los elementos entre la posición `pos1` y la posición `pos2` inclusive. Asuma las posiciones como indizadas en 1, y que `pos1` es menor o igual que `pos2`.

Firma: `public void nosVimoEnTokyo(int pos1, int pos2)`

Por ejemplo:

1891,5,3,1,1891	-> pos1: 1, pos2: 3	-> 1891,1891
1891,5,3,1,1891	-> pos1: 1, pos2: 200	-> 1891
1891,5,3,1,2,3,4	-> pos1: 9, pos2: 200	-> 1891,5,3,1,2,3,4
Vacia	-> pos1: 1, pos2: 200	-> Vacia

- b) Dado el siguiente vector: `int v[] = {35,7,67,52,31,28}`  
Indique el algoritmo de ordenación utilizado en la vista parcial y realice su implementación:(5 pts)
- 35 - 07 - 67 - 52 - 31 - 28
  - 07 - 35 - 67 - 52 - 31 - 28
  - 07 - 28 - 67 - 52 - 31 - 35
  - 07 - 28 - 31 - 52 - 67 - 35

#### Ejercicio 4 (10 ptos)

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[][])`