

EVALUACION	EXAMEN AED1	GRUPO	TODOS	FECHA	16/05/2025
MATERIA	Algoritmos				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	- Puntos: 100 - Duración: 3 horas - Sin material				

Ejercicio 1 (25 pts)

Dado una matriz cuadrada de enteros:

1	4	2	3
2	4	7	4
5	1	1	5
3	2	1	7

a) Escriba un algoritmo que retorne la suma de los elementos de la diagonal secundaria.
Firma: **public static int sumaDiagonalSecundaria(int[][] mat) (5 pts)**

b) Escriba un algoritmo que retorne un boolean, indicando si existe en la matriz dos columnas consecutivas que sumen los mismo.

Firma: **public boolean int filaMenorSuma(int[][] mat) (10 pts)**

Nota: en el ejemplo dado, la columna de índice 1 y 2 suman los mismo y son consecutivas (11)

c) Escriba un método recursivo que retorne la cantidad de números pares presentes en una columna específica.

Firma: **public static int contarParesColumna(int[][] mat, int col) (10 pts)**

Nota: es posible cambiar la firma del método si lo desea

Ejercicio 2 (20 pts)

Dado un vector de enteros desordenado y teniendo a disposición los métodos posMinVec(int vec[], int desde, int hasta) y posMaxVec(int vec[], int desde, int hasta) vistos en el curso, que retornan el índice en donde se encuentra el menor y mayor elemento del vector entre las posiciones desde y hasta respectivamente, implemente un método que, utilizando los métodos anteriores, permita ordenar un vector en forma ascendente.

Firma: **public static void ordenarAsc (int[] vec)**

Ejercicio 3 (55 pts)

Se ha implementado una Lista simplemente enlazada que cuenta con un puntero al inicio, puntero al final y un entero para almacenar la cantidad de elementos.

```
public class Lista {          public class Nodo {
    private Nodo inicio;      private int dato;
    private Nodo fin;         private Nodo sig;
    private int cantidad;
    //.....                //Métodos de acceso y modificación disponibles
}                             }
```

Implemente las siguientes operaciones en el TAD Lista:

a) Implemente una operación **agregarFinal(int dato)**, que agrega un elemento al final de la lista.

Firma: **public void agregarFinal (int dato) (10 pts)**

b) Implemente una operación de instancia **eliminarIndice** que elimina el elemento de la lista que se encuentra en el índice indicado (el primer elemento se encuentra en el índice 0). Definir las pre y post condiciones.

Firma: **public void eliminarIndice (int indice) (15 pts)**

c) Implemente la operación **sumarDesde** de forma recursiva, que reciba un índice desde el cual comenzar a sumar los valores de la lista (el primer elemento se encuentra en el índice 0)
Pre: desde < cantidad de elementos

Firma: **public int sumarDesde(int desde) (15 pts)**

d) Implemente la operación **insertarOrdenado**, que inserte un dato manteniendo la lista ordenada en forma ascendente. Se cuenta con los métodos disponibles **agregarInicio(int dato)** y **agregarFinal(int dato)**. Se tomará en cuenta la eficiencia.

Firma: **public void insertarOrdenado(int dato) (15 pts)**

Notas:

- Para todas las operaciones solicitadas se dispone de gets y sets
- Se pueden utilizar funciones o métodos auxiliares, pero se deben implementar.
- Indicar claramente que parte se está resolviendo.
- Escribir con letra legible ya que se considerará durante la corrección.