

<b>EVALUACION</b>	Parcial Matutino (TODOS)	<b>GRUPO</b>	TODOS	<b>FECHA</b>	09/12/2024
<b>MATERIA</b>	Algoritmos 1				
<b>CARRERA</b>	Analista Programador / Analista en Tecnologías de la Información				
<b>CONDICIONES</b>	<b>- Puntos: 45</b> <b>- Duración: 2 horas</b> <b>- Sin material</b>				

### Ejercicio 1 (10 pts)

- Desarrollar un algoritmo que reciba la matriz de enteros, una columna y un valor, e indique mediante un valor boolean si se encuentra dicho valor en la columna indicada de la matriz.
- Desarrollar un algoritmo que verifique si existe en la matriz alguna fila donde se encuentren 3 números consecutivos iguales. **(6 puntos)**

**Ejemplo de matriz de 6 X 5: tanto en fila 1, como en fila 4 se cumple la condición**

10	1	34	34	22
34	5	7	7	7
6	5	45	3	4
45	8	45	45	0
44	66	66	66	8
78	9	0	32	11

### Ejercicio 2 (8 pts)

A continuación, se detalla una versión con errores del algoritmo insert sort utilizado para ordenar el siguiente vector en forma ascendente: [34,11,5,1,3,90]

```
public static void insertSort(int[] arr) {
    for (int i = 1; i < arr.length; i++) {
        int actual = arr[i];
        int j = i;
        while (j > 0 && arr[j] < arr[j - 1]) {
            arr[j] = arr[j - 1];
            j--;
        }
        arr[i] = actual;
    }
}
```

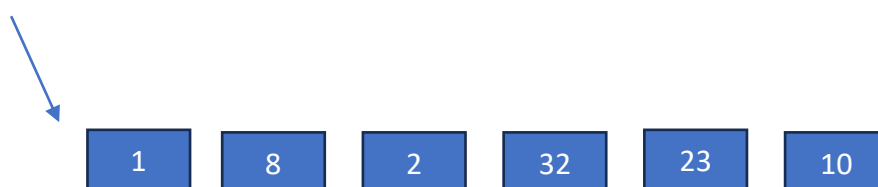
- Escribir el algoritmo con los errores solucionados **(4 puntos)**
- Tomando en cuenta el array inicial y el algoritmo corregido, indicar como quedaría el vector en las sucesivas pasadas (estado del vector luego de cada iteración) **(4 puntos)**

### Ejercicio 3 (12 pts)

Dada una lista simplemente enlazada de nodos con valores enteros y un valor N:

- a) Implemente un algoritmo recursivo que retorne la suma de los valores de la lista que sean mayores a N. **(10 pts)**.

inicio



Ej: para un valor N= 10 debería retornar: 55

- b) Realice el diagrama de llamadas para el ejemplo dado **(2 pts)**

### Ejercicio 4 (15 pts)

Dada una implementación de un TAD Lista con nodos simplemente enlazados **ordenados en forma ascendente sin números repetidos** con puntero al inicio como se muestra a continuación:

inicio



- a) Implemente una nueva operación en el TAD que inserte en forma ordenada un valor indicado en la lista, únicamente si ya no existe dicho número en la lista. Se deberá retornar un boolean indicando si se pudo agregar o no el valor.

**public boolean insertarOrdenado (int valor) (7 pts)**

- b) Implemente una nueva operación en el TAD que retorne una cola con los elementos que se encuentran en la lista. La cola debe estar dispuesta de forma de que el elemento que se encuentra al principio de la lista sea el último elemento de la cola y el elemento que se encuentra al final de la lista sea el frente de la cola.

**public Cola listaACola() (8 pts)**

Ejemplo de retorno para lista dada: Cola resultante: 4->5->7->23->30->39

Si se utilizan operaciones del TAD no solicitadas, deben implementarse. Se asumen disponibles los métodos: getDato(), setDato(), getSig() y setSig(), así como todas las operaciones del TAD Cola (frente(), encolar(), desencolar(), esVacia()) y las de TAD Pila (top(), apilar(), desapilar(), esVacia())