

EVALUACION	Parcial Nocturno	GRUPO	TODOS	FECHA	XX/07/2024
MATERIA	Algoritmos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	<b>- Puntos: 45</b> <b>- Duración: 2 horas</b> <b>- Sin material</b>				

### Ejercicio 1 (10 pts)

Sea M una matriz de enteros positivos:

Ejemplo de matriz de 4 X 5:

	0	1	2	3	4
0	56	56	3	54	3
1	11	12	6	40	14
2	11	34	4	37	1
3	8	34	35	30	38
4	7	23	1	5	45
5	6	88	2	4	56

- a) desarrollar un algoritmo que verifique si existe una columna en donde los elementos estén ordenados de manera estrictamente decreciente. En caso de existir, retornar el índice de dicha columna (cualquiera de las que cumpla la condición); en caso contrario retornar -1. **(5 pts)**

### Ejercicio 2 (8 pts)

A continuación, se detalla una versión del algoritmo insert sort, la cual cuenta con algunos errores en la sección marcada en amarillo.

```
public static void insertionSort(int[] arr) {
    for (int i = 1; i < arr.length; i++) {
        int key = arr[i];
        int j = i - 1;
        // Completar el siguiente bucle while
        while (j > 0 && arr[j] > key) {
            arr[j] = arr[j+1];
            j = j + 1;
        }
        arr[j] = key;
    }
}
```

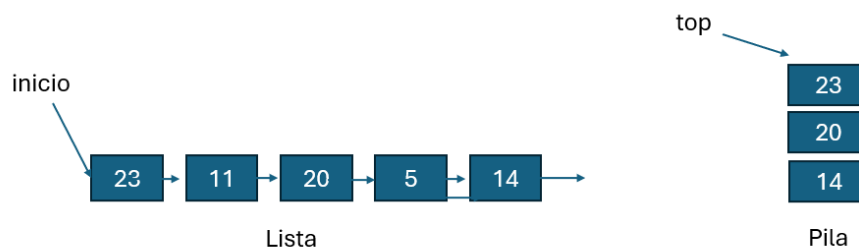
- a) Corrija únicamente lo que considere que es incorrecto del algoritmo. **(5 pts)**  
b) Para un vector con los valores [34,1,22,18,33], muestre como quedaría el vector luego de cada inserción. **(3 pts)**

### Ejercicio 3 (12 pts)

Dado una lista simplemente enlazada de nodos con valores enteros positivos, un valor C y una pila vacía:

- a) Implemente un algoritmo recursivo que coloque todos los elementos de la lista que sea mayores que C en la pila, de forma de que el primer elemento de la lista que cumpla con la condición se sitúe en el tope de la pila. Defina las pre y post condiciones. **(10 pts)**. Se sumen disponibles los métodos gets y sets, así como todas las operaciones del TAD Pila (top(), apilar(), desapilar(), esVacia())

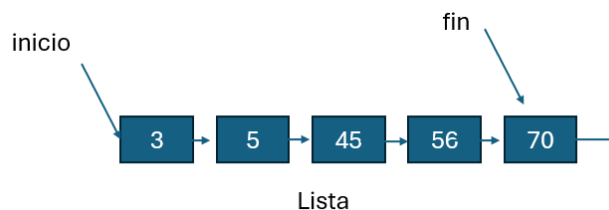
Ejemplo para un valor C = 12



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

### Ejercicio 4 (15 pts)

Dada una implementación de un TAD Lista con nodos simplemente enlazados ordenados en forma ascendente, con puntero al inicio y puntero al final como se muestra a continuación:



- a) Implemente una nueva operación en el TAD que dado un número, retorne la posición en donde se encuentra dicho número en la lista, o retorne -1 en caso de no encontrarlo. Se valorará la eficiencia de la implementación. Ejemplo: para la lista anterior y el número 56, debería retornar 4.

**Firma sugerida: public int posiciónDeNumero(int num) (8 pts)**

- b) Implemente una nueva operación en el TAD que dada una posición, elimine de la lista el elemento que se encuentra en dicha posición y el siguiente (en caso de no existir un siguiente solamente eliminará el de la posición dada). Para un pos=2 la lista quedaría con: 3->56->70

**Firma sugerida: public void eliminar2Elementos(int pos) (8 pts)**

Si se utilizan operaciones del TAD no solicitadas, deben implementarse. Se cuentan disponibles los métodos gets y sets.