

<b>EVALUACION</b>	Examen	<b>GRUPO</b>		<b>FECHA</b>	03/08/2022
<b>MATERIA</b>	Algoritmos 1				
<b>CARRERA</b>	Analista Programador / Analista en Tecnologías de la Información				
<b>CONDICIONES</b>	- Puntos: Máximo: Mínimo: - Duración: 2 horas - Sin material				
Indique docente con el que curso					
<b>Nombre</b>	<b>Nro estudiante</b>		<b>Nota</b>		

### Ejercicio 1 (20 pts)

Dado el siguiente vector `int v[] = {35,7,67,52,31,28}` y algoritmo:

```

public static int algoritmo1(int[] vector) {
    int i, aux, j, contador;
    contador=0;
    for (i = 1; i < vector.length; i++) {
        aux = vector[i];
        j = i-1;
        while ( (j >= 0) && (vector[j] > aux) ) {
            vector[j+1] = vector[j];
            j--;
            contador++;
        }
        vector[j+1] = aux;
    }
    return contador;
}

```

Responda las siguientes preguntas, una vez ejecutado el algoritmo

- Cuántas recorridas completas hace del vector dado **(5 pts)**
- Que representa el contador y cuál es el valor de retorno **(5 pts)**
- Complete la secuencia utilizando el método, hasta que quede ordenado **(10 pts)**

- 35 – 7 – 67 – 52 – 31 - 28

....

....

....

....

....

-7 – 28 – 31 – 35 – 52 – 67

---

## Ejercicio 2 (15 pts)

Implemente un método de ordenamiento ascendente (menor a mayor) utilizando el método auxiliar:

```
int Buscposminimo(int v[],posdesde,poshasta)
```

que retorna la posición donde se encuentra el valor mínimo del vector entre 2 posiciones dadas como se muestra en el siguiente ejemplo

**Ejemplo:** para el vector dado en el ejercicio 1 (35 – 7 – 67 – 52 – 31 - 28

Buscposminimo(int v[],0,5) retorna 1 (posición del 7, mínimo valor)

Buscposminimo(int v[],2,5) retorna 5 (posición del 28, mínimo valor)

## Ejercicio 3 (35 pts)

Se desea implementar una bandeja de entrada de correos (ej Gmail), donde se almacenen mensajes de correo electrónico con una vista en la que los correos recientes se visualizan primero.

La idea es entonces tener una estructura dinámica donde almacenar los correos que van llegando, con la fecha de recibido, el origen (String), el asunto (String), y el cuerpo del mensaje (String).

Se solicita:

- a) Definir la clase Nodo y estructura a utilizar para satisfacer los requerimientos mencionados y la estructura **(5 pts)**

Implemente las operaciones básicas de la estructura, para

- b) Consultar si hay algún correo de un remitente dado **(10 pts)**
- c) Eliminar todos los correos que coincidan con determinado asunto **(10 pts)**
- d) Consultar si la bandeja está vacía **(5 pts)**
- e) Justifique la estructura elegida en términos de eficiencia para los métodos mencionados. **(5 pts)**

### **Ejercicio 4 (30 ptos)**

Escribir una función:

// el método debe ser eficiente

**void Reemplazar(pila P,int nuevo,int viejo)**

que tenga como argumentos una pila P de enteros un valor int: nuevo y un valor int viejo de forma que, si el viejo valor aparece en algún lugar de la pila, sea reemplazado por el nuevo.

**Nota:** se permite utilizar una pila auxiliar y se asume disponibles los métodos:

- pop() ,
- push()
- tope()
- esvacía()
- esllena()