UNIVERSIDAD ORT
Uruguay

**Facultad de Ingeniería**
***Bernard Wand-Polak***
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

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

## Ejercicio 1 (20 ptos)

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

a) Cuantas recorridas completas hace del vector dado **(5 ptos)**
**Una**

**b)** Que representa el contador y cuál es el valor de retorno **(5 ptos)**
**La cantidad de intercambios de elementos dentro del vector**

c) Complete la secuencia utilizando el método, hasta que quede ordenado **(10 ptos)**

- 35 – 7 – 67 – 52 – 31 - 28
….
….
….

**Facultad de Ingeniería**
***Bernard Wand-Polak***
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

….
….
-7 – 28 – 31 – 35 – 52 – 67

```
35 7 67 52 31 28 antes
7 35 67 52 31 28 i=1
7 35 67 52 31 28 i=2
7 35 52 67 31 28 i=3
7 31 35 52 67 28 i=4
7 28 31 35 52 67 i=5
```

**Ejercicio 2 (15 ptos)**

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

int Buscoposminimo(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

Buscoposminimo(int v[],0,5) retorna 1 (posición del 7, mínimo valor)
Buscoposminimo(int v[],2,5) retorna 5 (posición del 28, mínimo valor)

```
public static void ordenarPosMinimo (int v[]){

      for (int i=0;i<v.length;i++){
            int posmin = Buscoposminimo(v,i,v.length-1);
            if (v[posmin]<v[i]){
                  int aux = v[i];
                  v[i]= v[posmin];
                  v[posmin]= aux;
            }
      }
}
```

**Ejercicio 3 (35 ptos)**

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).

**Facultad de Ingeniería**
***Bernard Wand-Polak***
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

Se solicita:

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

```
public class NodoLista{

   private Date fechaRecibido;
   private String remitente;
   private String asunto;
   private String cuerpoMensaje;
   private NodoLista sig;

   public Date getFechaRecibido() {
          return fechaRecibido;
   }
   public void setFechaRecibido(Date fechaRecibido) {
          this.fechaRecibido = fechaRecibido;
   }
   public String getRemitente() {
          return remitente;
   }
   public void setRemitente(String remitente) {
          this.remitente = remitente;
   }
   public String getAsunto() {
          return asunto;
   }
   public void setAsunto(String asunto) {
          this.asunto = asunto;
   }
   public String getCuerpoMensaje() {
          return cuerpoMensaje;
   }
   public void setCuerpoMensaje(String cuerpoMensaje) {
          this.cuerpoMensaje = cuerpoMensaje;
   }

  public NodoLista(Date fechaRecibido,String remitente,String asunto,
                String cuerpoMensaje){

   this.fechaRecibido=fechaRecibido;
   this.remitente=remitente;
   this.asunto=asunto;
   this.cuerpoMensaje=cuerpoMensaje;
  this.sig=null;
  }
  public void setSig(NodoLista s){
      this.sig=s;
  }
  public NodoLista getSig(){
      return this.sig;
  }
}
```

**Facultad de Ingeniería**
***Bernard Wand-Polak***
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

```
public class Lista  {

 private NodoLista inicio;
 //Constructor
 public void Lista(){
     this.inicio=null;
 }
 //Inicio
 public void setInicio(NodoLista nodo){
     inicio=nodo;
 }
 public NodoLista getInicio(){
     return inicio;
 }

 public void agregarInicio(NodoLista nodo){
     NodoLista nuevo= nodo;
     nuevo.setSig(inicio);
     this.inicio=nuevo;
 }

 }
```

Implemente las operaciones básicas de la estructura, para

b) Consultar si hay algún correo de un remitente dado **(10 ptos)**

```
public boolean existeRemitente(String remitente){
    NodoLista aux=this.inicio;
    while (aux!=null ){
         if (aux.getRemitente().equals(remitente)){
               return true;
         }
        aux=aux.getSig();
    }
    return false;
}
```

**c)** Eliminar todos los correos que coincidan con determinado asunto **(10 ptos)**

```
public void borrarAsunto (String asunto) {

  if( this.esVacia()){
      return;
  }

  NodoLista aux =  this.inicio;

  if (aux.getAsunto().equals(asunto)){
       if (this.inicio.getSig()==null){
          this.inicio = null;
      } else {
          this.inicio=this.inicio.getSig();
      }
      return;
  }

  while (aux!=null && aux.getSig()!=null){
      if (aux.getSig().getAsunto().equals(asunto)){
          aux.setSig(aux.getSig().getSig());
      }
      aux = aux.getSig();
  }
}
```

d) Consultar si la bandeja está vacía **(5 ptos)**

```
public boolean esVacia(){
    if (this.inicio==null)
        return true;
    else
        return false;
}
```

**e)** Justifique la estructura elegida en términos de eficiencia para los métodos mencionados. **(5 ptos)**

**La estructura elegida es una lista simplemente encadenada donde se agregan elementos solo al inicio a los efectos de mantener los mails recibidos ordenados por orden de llegada.**

**Si ese hubiera sido el único requisito podría haber sido una Pila pero al tener que recorrer la estructura para buscar elementos se eligió una Lista.**

**Facultad de Ingeniería**
***Bernard Wand-Polak***
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

**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 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()
- esvacia()
- esllena()

```
public static void reemplazar (Pila p, int nuevo, int viejo){

    Pila aux = new Pila();
    int elementoTope;

    while (! p.esVacia()){
          elementoTope = p.tope();
          if (elementoTope==viejo){
                aux.push(nuevo);
          } else {
                aux.push(elementoTope);
          }
          p.pop();
   }

    while (! aux.esVacia()){
          p.push(aux.tope());
          aux.pop();
   }

 }
```