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	SOLUCION Examen		**FECHA**	20/10/2017
MATERIA	Algoritmos y Estructuras de Datos1			
CARRERA	Analista Programador / Analista en Tecnologías de la Información			
CONDICIONES	**Especifique las pre y post condiciones de los métodos que implemente.** **Serán tenidos en cuenta ejercicios o partes de ellos completas y que el estilo y metodología de desarrollo se ajuste al curso.** **IMPORTANTE** - Duración **3 horas** - **NO** se puede consultar material			

ESTUDIANTE **(nombre y número)**		**Nota**

Ejercicio 1 (35 puntos)

Dado el siguiente array **V**:

1	5	17	11	21	13	15	19	23	25
0	1	2	3	4	5	6	7	8	9

a) Implementar un algoritmo que haga una transposición entre los k primeros términos del array y los n – k términos restantes sin utilizar un vector auxiliar.

Firma para el método a implementar :

- public void cambiar(int[] array, int n, int k)

Si ejecutamos el método con los valores del ejemplo, Cambiar(V, 3, k) luego de ejecutado el algoritmo, el vector resultante es el siguiente:

11	21	13	15	19	23	25	1	5	17
0	1	2	3	4	5	6	7	8	9

b) Ordenar el vector utilizando el método de inserción

c) Implemente un algoritmo recursivo que sume los elementos del vector
Utilizar ls siguiente firma:
int suma_vec(int v [], int n) // n es la cantidad de elementos del vector

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 2 (25 puntos)

Dados los siguientes seudocódigos correspondientes a los métodos de ordenamiento

```
void _____Sort (int* A, int n){
int i, j, First;
for (i = 1; i < n; i++){
First = A[i];
j = i-1;
while (j >= 0 && First < A[j]){
A[j+1] = A[j];
j--;
}
A[j+1] = First;
}
}
```

```
void _____Sort (int* A, int n){
int i, j, posmin, tmp;
for (i = 0; i < n-1; i++){
posmin = i;
for (j = i+1; j < n; j++){
if (A[j] < A[posmin])
posmin = j;
}
// Intercambio de elementos
tmp = A[i];
A[i] = A[posmin];
A[posmin] = tmp;
}
}
```

a) Indique a que algoritmo corresponde cada seudocódigo.
b) Aplique los mismos al siguiente vector de enteros 2, 3, 5, 1, 4. Y:
    a. Muestre la secuencia generada hasta ordenar el vector.
    b. Determine cual algoritmo emplea menos *esfuerzo* en ordenarlo.

Para ello se consideran las siguientes reglas:

- La comparación entre elementos del arreglo implica *una unidad de esfuerzo*
- El intercambio de elementos en el arreglo implica *dos unidades de esfuerzo* (solo si son elementos diferentes).

Justifique su respuesta mostrando paso a paso la ejecución de los algoritmos, especificando en cada paso qué elementos se comparan, cuáles intercambios se realizan y cómo queda el arreglo al final de cada paso.

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

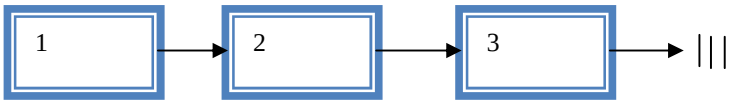
Ejercicio 3 (40 puntos)

Dada la siguiente programa en el que se esta probando una lista simplemente enlazada:

```
public static void main(String[] args) {

    System.out.println("Creamos lista vacia:");
    Lista miLista=new Lista();
    System.out.println("Agregamos al inicio valores 3, 2, 1:");
    miLista.agregarInicio(3);
    miLista.agregarInicio(2);
    miLista.agregarInicio(1);
    System.out.println("Mostrar Lista: ");
    miLista.mostrar (miLista.getInicio());  // se debe mostrar en orden inverso del ultimo al primero
}
```

Con la siguiente estructura



A) Defina la estructura de nodo y lista para una lista de enteros.
B) Implemente un método agregar inicio
C) Implemente un método que muestre la lista del ultimo al primero
Es decir, si se ingresan los datos como se muestra en la figura y el código el método mostrar debe mostrar 3 – 2 - 1

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

soluciones

Solución Ejercicio 1

A)
```
void cambiar(int[] array, int n, int k) {
    if (k>0) {
        int aux= array[0];
        for(int i=1;i<n; i++)
            array[i-1]=array[i];
        array[n-1]=aux;
        cambiar(array, n, k-1);   }
}
```

B)
```
ordenoporInsercion(int array[]){
    int i, temp, j;
    for (i = 1; i < array.length; i++){
        temp = array[i];
        j = i - 1;
        while ( (array[j] > temp) && (j >= 0) ){
            array[j + 1] = array[j];
            j--;       }
        array[j + 1] = temp;
    }
}
```

C)
```
int suma_vec(int v [], int n) {

    if (n == 0)
      return v [n];

    else

      return suma_vec(v, n - 1) + v [n];
```

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

Solución Ejercicio 2

INSERTION SORT	SELECTIONSORT
Primer paso	**Primer paso**
First = 2	Posmin = 0
Arreglo: 2,3,5,1,4	3 < 2 NO
Segundo paso:	5 < 2 NO
First = 3	1 < 2 SI
3 < 2 NO	Posmin = 3
Arreglo: 2,3,5,1,4	4 < 1 NO
Tercer paso	Intercambian 1 y 2
First = 5	Arreglo: 1,3,5,2,4
5 < 3 NO	**Segundo paso:**
Arreglo: 2,3,5,1,4	Posmin = 1
Cuarto paso	5 < 3 NO
First = 1	2 < 3 SI
1 < 5 SI	Posmin = 3
Intercambian 5 y 1	4 < 2 NO
Arreglo: 2,3,1,5,4	Intercambian 2 y 3
1 < 3 SI	Arreglo: 1,2,5,3,4
Intercambian 3 y 1	**Tercer paso:**
Arreglo: 2,1,3,5,4	Posmin = 2
1 < 2 SI	3 < 5 SI
Arreglo: 1,2,3,5,4	Posmin = 3
Quinto paso:	4 < 3 NO
First = 4	Intercambian 3 y 5
4 < 5 SI	Arreglo: 1,2,3,5,4
Intercambian 5 y 4	**Cuarto paso:**
Arreglo: 1,2,3,4,5	Posmin = 3
4 < 3 NO	4 < 5 SI
	Intercambian 4 y 5
	Arreglo: 1,2,3,4,5

Se hacen 7 comparaciones y 4 intercambios, por lo que el esfuerzo total del insertion sort para ordenar de forma ascendente el arreglo es de 15 unidades.

Se hacen 10 comparaciones y 4 intercambios, por lo que el esfuerzo total del selection sort para ordenar de forma ascendente el arreglo es de 18 unidades.

En resumen, insertionsort lo hace con menos esfuerzo.

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

Solucion Ejercicio 3
Parte a)

```
//Definición de la estructura
public class NodoLista{
        private int dato;
        private NodoLista sig;

    //Constructor
    public NodoLista(int n){
        this.dato=n;
        this.sig=null;
    }

    //Dato
    public void setDato(int d){
        this.dato=d;
    }
    public int getDato(){
        return this.dato;
    }

    //Siguiente
    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 i){
        inicio=i;
    }
    public NodoLista getInicio(){
        return inicio;
    }
}
```

Parte b

```
public void agregarInicio(int n){
        NodoLista nuevo= new NodoLista(n);
        nuevo.setSig(inicio);
        this.inicio=nuevo;
        if(this.fin==null)//estoy insertando el primer nodo
            this.fin=nuevo;
      }
```

Parte c)

```
  public void mostrar (NodoLista l){

       if(l!=null){
           mostrarl.getSig());

        System.out.println(l.getDato());

}
```

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