

EVALUACION	EXAMEN AED1	GRUPO	TODOS	FECHA	
MATERIA	Algoritmos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	- Puntos: 100 - Duración: 2 horas - Sin material				
Nombre	Nro estudiante		Nota		

Ejercicio 1 (30 pts)

Dada una matriz de mxm como se muestra en el ejemplo:
Te vamos a pedir que realices los siguientes algoritmos.

Ejemplo:

12	1	3	4
3	3	2	5
4	2	8	7
11	3	4	6

- a) Hacer un algoritmo que intercambie la primera columna con la última columna de una matriz

Firma sugerida: `public static void intecambiocol (int[][] M)`

Solución

```
public static void intecambiocol (int[ ][ ] M){
    int ultimafila=M.length;
    int ultimacol=M[0].length-1;
    for (int fila=0;fila<ultimafila;fila++){
        int aux=M[fila][0];
        M[fila][0]=M[fila][ultimacol];
        M[fila][ultimacol]=aux;
    }
}
```

- b) Escribir un algoritmo recursivo que sume los elementos de la diagonal principal. Realice el diagrama de llamadas para el ejemplo dado.

solucion

```
public static int sumadiagonalM (int f, int c, int[ ][ ] M ) {
    if (f==M.length-1 && c==M[0].length-1)
        return M[M.length-1][M[0].length-1];
    else
        return M[f][c] +sumadiagonalM (f+1, c+1, M ) ;
}
```

```
sumadiagonalM (0, 0, mat)
return 12 + sumadiagonalM (1, 1, mat) = 29
    sumadiagonalM (1, 1, mat)
    return 3 + sumadiagonalM (2, 2, mat) = 17
        sumadiagonalM (2, 2, mat)
        return 8 + sumadiagonalM (3, 3, mat) = 14
            sumadiagonalM (3, 3, mat)
            return 6
```

Ejercicio 2 (30 ptos)

Asumiendo que se dispone del TAD Lista de enlace doble con los métodos **esVacia()**, **agregarInicio(int dato)**, **eliminarInicio()**, **Listar()** ya implementados, se solicita:

- a) **Hagas un algoritmo** Realizar un algoritmo que reciba la matriz del ejercicio 1 y retorne una lista (de enlace doble) que contenga los elementos de la matriz en forma ordenada (15 ptos)

Utilizar la siguiente firma:

public static Ldoble cargarMatrizenLista(int[][] m)

solución

```
public static Ldoble cargarMatrizenLista(int [][] m){
    Lista l = new Lista();
    filas=m.length();
    columnas=m[0].length();
    for (int i=0; i<filas; i++){
        for (int j=0; j<columnas; j++){
            l.agregarOrdenado(m[i][j])
        }
    }

    return l;
}
```

En TAD Lista:

```
public void agregarOrdenado(int num){  
  
    if(this.getIncio().getDato() >= num){  
  
        this.agregarInicio(num);  
    }  
    else{  
  
        Nodo aux = this.getIncio();  
  
        while(aux.getSiguiente() != null && aux.getSiguiente().getDato() <= num){  
            aux = aux.getSiguiente();  
        }  
  
        Nodo nuevo = new Nodo(num);  
  
        If(aux.getSiguiente() != null){  
            nuevo.setSiguiente(aux.getSiguiente());  
            aux.getSiguiente().setAnterior(nuevo);  
        }  
  
        nuevo.setAnterior(aux);  
        aux.setSiguiente(nuevo);  
  
    }  
}
```

b) Utilizando el TAD lista, se solicita modificar el siguiente algoritmo, de forma que retorne un String que contenga los datos impares de la lista

```
public String mostrarRec1(NodoDoble primero, NodoDoble ultimo) {  
  
    if (primero==ultimo){  
        if(primero.getDato() %2 != 0){  
            return " - " + primero.getDato() ;  
        }  
        else{  
            return "";  
        }  
    }else{  
  

```

```
        if(primero.getDato() %2 != 0){
            return primero.getDato()+ " - " +
                mostrarRec1(primero.getSiguiente(),ultimo);
        }
        else{
            return mostrarRec1(primero.getSiguiente(),ultimo);
        }
    }
}
```

Ejercicio 3 (20 ptos)

Implementa una función que tome una cadena como entrada y la devuelva en orden inverso utilizando una Pila.

```
public String reverso(String cadena) {

    Pila p = new Pila();
    Int largoCadena = cadena.length();

    for(int i=0; i< largoCadena; i++){
        p.push(cadena.charAt(i))
    }

    String ret = "";

    while(!p.esVacia()){
        ret += p.top();
        p.pop();
    }

    return ret
}
```

Ejercicio 4 (20 ptos)

Asumiendo disponibles los métodos del TAD Lista del ejercicio 2, implemente un algoritmo que reciba una lista y retorne sus datos en un array ordenado, utilizando cualquier de los métodos de ordenación vistos en clase.

Implementar Bubble Sort, InsertSort, Selection Sort., Quick Sort o Merge Sort.