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	Solución EXAMEN AED1	**GRUPO**	TODOS	**FECHA**	12/02/2025
MATERIA	Algoritmos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	**- Puntos: 100** **- Duración: 3 horas** **- Sin material**				

Ejercicio 1 (25 ptos)

Dado una matriz cuadrada de enteros, no repetidos:

1	6	7	31
10	4	11	40
3	4	5	43
51	2	8	9

a) Escriba un algoritmo que retorne el máximo de los elementos de la matriz, entre dos filas dadas. **(5 ptos)**

//pre: 0 <= fila1 && fila1 < mat.length && 0 <= fila2 && fila2 < mat.length
Firma: **public static int maximoEntreFil (int[ ] [ ] mat, int fila1, int fila2)**
Ej: para fila1: 1 y fila2:2, el resultado debería ser: 43

```
public static int maximoEntreFil(int[][] mat, int fila1, int fila2) {

    int max = Integer.MIN_VALUE;

    for (int i = fila1; i <= fila2; i++) {
        for (int j = 0; j < mat[i].length; j++) {
                if(mat[i][j] > max){
                        max = mat[i][j];
                }
    }

    return max;
}
```

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

b) Escriba un algoritmo que retorne un valor booleano, indicando si existe en la matriz alguna fila cuyos valores se encuentran dispuestos en forma estrictamente ascendente **(10 ptos)**
Ej: para el ejemplo dado, la fila 0 y la fila 2 cumplen la condición.

```
public static boolean existeFilaAscendente(int[][] mat) {

    for (int i = 0; i < mat.length; i++) {
      boolean ascendente = true;
      for (int j = 1; j < mat[i].length && ascendente; j++) {
        if (mat[i][j] <= mat[i][j - 1]) {
          ascendente = false;
        }
      }
      if (ascendente) {
        return true;
      }
    }
    return false;
  }
```

c) Escriba un método recursivo que muestre los elementos de la diagonal principal de la matriz, en forma inversa. Realice el diagrama de llamadas para la matriz dada.
(10 ptos)
Ej: para el ejemplo de la matriz dada, se debería mostrar 9 5 4 1

```
public static void mostrarDiagonalInversa(int[][] mat, int indice) {
  if (index >= 0) {
        mostrarDiagonalInversa(mat, index - 1);
        System.out.print(mat[index][index] + " ");

  }
}
```

```
mostrarDiagonalInversa(mat, 3)
    SOUT “9”
    mostrarDiagonalInversa(int[][] mat, 2)
        SOUT “5”
        mostrarDiagonalInversa(int[][] mat, 1)
            SOUT “4”
            mostrarDiagonalInversa(int[][] mat, 0)
                SOUT “1”
```

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 (15 ptos)

Dado un vector ordenado en forma ascendente de números enteros y un valor, implemente el método de búsqueda por punto medio (en forma iterativa o recursiva)

Firma: **public static boolean buscar (int[ ] vec, int valor)**

```
public static boolean buscar(int[] vec, int valor) {

    int inicio = 0,
    int fin = vec.length - 1;

    while (inicio <= fin) {
        int medio = inicio + (fin - inicio) / 2;

        if (vec[medio] == valor) {
            return true;
        } else if (vec[medio] < valor) {
            inicio = medio + 1;
        } else {
            fin = medio - 1;
        }
    }

    return false;
}
```

Ejercicio 3 (60 ptos)

Se ha implementado una Lista simplemente enlazada que cuenta con un puntero al inicio, un entero para almacenar la cantidad de elementos y un entero que limita su capacidad máxima.

```
public class Lista {
      private Nodo inicio;
      private int cantidad;
      private int capMax;
      //…….
}
```

```
public class Nodo {
      private int dato;
      private Nodo sig;

//Métodos de acceso y modificación disponibles
}
```

Implemente las siguientes operaciones en el TAD 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

a) Implemente una nueva operación de instancia que retorne un vector con los elementos de la lista, conservado el mismo orden y posiciones **(10 ptos)**

Firma: **`public int[] darVectorDeLista()`**

```
public int[] darVectorDeLista(Lista l) {
   int[] vector = new int[l.getCantidad()];
   Nodo aux = l.getInicio();
   for (int i = 0; i < lista.getCantidad(); i++) {
      vector[i] = aux.getDato();
       aux = aux.getSiguiente();
   }
   return vector;
}
```

b) Implemente la operación de instancia agregarInicio, que inserta el elemento al comienzo de la lista, retornando un boolean que indica si se pudo efectivamente agregar **(10 ptos)**

Firma: **`public boolean agregarInicio(int dato)`**

```
public boolean agregarInicio(int dato) {

    if(cantidad != capMax ){
          Nodo nuevo = new Nodo(dato);
          if(inicio == null){
                inicio = nuevo;
          }
          else{
                nuevo.setSiguiente(inicio);
                inicio = nuevo;
          }
          cantidad++;
          return true;
    }
    else{
        return false;
    }

}
```

c) Implemente la operación de instancia **colaMayoresDe** (de forma recursiva), que retorne una cola con todos los elementos mayores al parametro. **(20 ptos)**

Firma: **`public Cola colaMayoresDe (int dato)`**
Ej: para la lista 10-3-5-76-11-4-2 y el dato 4, debería retornar la siguiente cola: 10 5 76 11 (frente)
Nota: se pueden crear métodos auxiliares si lo considera necesario pero se debe implementar

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 Cola colaMayoresDe(int dato) {
   Cola cola = new Cola();
   colaMayoresDeAux(cola, dato, inicio);
   return cola;
}

private void colaMayoresDeAux(Cola cola, int dato, Nodo actual) {


   if (actual != null) {
        if(actual.getDato() > dato){
                colaMayoresDeAux(cola, dato, actual.getSiguiente());
                cola.encolar(actual.getDato())
        }

   }
}
```

d) Implemente la operación de instancia eliminarValor que elimina (en caso de existir) dicho valor de la lista **(20 ptos)**

Firma: **`public boolean eliminarValor (int dato)`**

```
public boolean eliminarValor eliminarValor(int dato){

      if(incio != null){

            if(inicio.getDato() == dato){

                  inicio = inicio.getSiguiente();
                  cantidad++;
            }
            else{
                  Nodo aux = inicio;
                  while(aux.getSiguiente() != null && aux.getSiguiente().getDato() != dato){
                        aux = aux.getSiguiente()
                  }

                  If(aux.getSiguiente() == null){ //no esta el dato
                        return false;
                  }else{
                        aux = aux.getSiguiente().getSiguiente();
                        return true;
                  }
            }
      }

}
```

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

Notas:

- **Para todas las operaciones solicitadas se dispone de gets y sets**
- **Se cuentan con las operaciones de Cola y Pila vistas en el curso.**
- **Se pueden utilizar funciones o métodos auxiliares, pero se deben implementar.**
- **Indicar claramente que parte se está resolviendo.**
- **Escribir con letra legible ya que se considerará durante la corrección.**