**Algoritmo A22Matriz**

Const

MAX\_REN = 5

MAX\_COL = 5

TipoTaMatriz = Arreglo[1..MAX\_REN, MAX\_COL] de entero

Var

aMatriz : TaMatriz

eOpcion, eCol, eRen, eElemento : entero

**Procedimiento LeerMatriz(TaMatriz: aMatriz)**

Var

eRen, eCol: entero

Inicio

Desde eRen = 1 hasta MAX\_REN hacer

Desde eCol = hasta MAX\_COL hacer

Escribir (“Ingrese matriz [ ”, eRen “,”, eCol, “]:”)

Leer aMAtriz [eRen, eCol]

Fin Desde

Fin Desde

Fin Procedimiento//LeerMatriz

**Procedimiento EscribirMatriz(EaMatriz : TaMatriz)**

Var

eRen, eCol: entero

Inicio

Desde eRen = 1 hasta MAX\_REN hacer

Desde eCol = 1 hasta MAX\_COL hacer

Escribir (“Matriz [ ”, eRen, “,”, eCol, “] = ”, aMatriz [eRen. eCol])

Fin Desde

Fin Desde

Fin Procedimiento // EscribirMatriz

**Función BuscarMatriz(EaMatriz : TaMatriz, EeElemento : entero, SeRen :entero, SeCol : entero): Lógico**

Inicio

Hacer eRen = 1

Hacer eCol = 1

Mientras (eElemento <> aMatriz[eRen, eCol] y eRen <= MAX\_REN) hacer

Hacer eCol = 1

Mientras (eElemento <> aMatriz[eRen, eCol], y eCol <= MAX\_COL) hacer

Hacer eCol = eCol + 1

Fin Mientras

Si (eElemento <> aMatriz[eRen, eCol]) entonces

Hacer eRen = eRen + 1

Fin Si

Fin Mientras

Si (eElemento) = aMatriz [eRen, eCol] entonces

Regresar Verdadero

De lo contrario

Regresar Falso

Fin Si

Fin Procedimiento // BuscarMatriz

**Procedimiento TransponerMatriz(EaMatriz : TaMatriz)**

Var

eCol, eRen, eTemp : entero

Inicio

Desde eRen = hasta MAX\_REN – 1 hacer

Desde eCol = eRen +1 hasta MAX\_COL hacer

Hacer eTem = aMatriz [eRen, eCol]

Hacer aMatriz[eRen, eCol] = aMatriz [eCol, eRen]

Hacer aMatriz[eCol, eRen] = eTemp

Fin Desde

Fin Desde

Fin Procedimiento // TransponerMatriz

**Función EsSimetrica(EaMatriz : TaMatriz): Lógico**

Var

lSimetrica: lógico

eRen, eCol: entero

Inicio

Si (MAX\_REN <> MAX\_COL) entonces

Regresar Falso

De lo contrario

Hacer lSimetrica = Verdadero

Hacer eRen =1

Hacer eCol = 1

Mientras (lSimetrica = Verdadero y eRen < MAX\_REN) hacer

Hacer eCol = eRen

Mientras (lSimetrica y eCol <= MAX\_COL) hacer

Hacer eCol = eCol + 1

Si (aMatriz[eRen, eCol] <> aMatriz[eCol, eRen])

Hacer lSimetrica = No lSimetrica

Fin Si

Fin Mientras

Hacer eRen = eRen + 1

Fin Mientras

Fin Si

Regresar lSimetrica

Fin Función // EsSimetrica

**//Programa Principal**

Inicio

Repetir

Escribir (“Programa que realiza manejo básico de matrices bidimensionales.”)

Escribir (“1.- Leer 2.- Escribir 3.- Buscar 4.- Transponer 5.- Verificar simetría 6.- Salir”)

Escribir (“Seleccione una opción: ”)

Leer eOpcion

En caso de (eOpinion) hacer

1: Ejecutar LeerMatriz(aMatriz)

2.- Ejecutar EscribirMAtriz (aMatriz)

3: Escribir (“Ingrese el elemento que desee buscar:”)

Leer (eElemento)

Si (BuscarMatriz(aMatriz, eElemento, eRen, eCol)) entonces

Escribir (“El elemento se encuentra en la posición [“, eRen, “,”, eCol, “[”)

De lo contrario

Escribir (“No se encontró el elemento”, eElemento)

Fin Si

4.- Ejecutar TransponerMatriz(aMatriz)

5.- Si (EsSimetrica(aMatriz)) entonces

Escribir (“La matriz si es simétrica.”)

De lo contrario

Escribir (“La matriz no es simétrica.”)

Fin Si

6: Escribir (“Adiós”)

De lo contrario: ….

Fin En caso de

Hasta que (eOpcion =6)

**Fin**