



## Solucionario - Práctica N° 4

### Arreglos Unidimensionales y Multidimensionales

#### Solucionario 1:

##### VARIABLES

ENTERO : \*v, i, mayor, menor

REAL : promedio, suma, cantidad

ACCION Hallar\_mayor\_menor\_promedio\_numeros

ESCRIBIR("Ingrese la cantidad de elementos del vector : ") LEER(cantidad)

mayor  $\leftarrow$  0

menor  $\leftarrow$  999

suma  $\leftarrow$  0

v  $\leftarrow$  reservar\_espacio ENTERO [ cantidad ]

PARA i DESDE 0 HASTA cantidad-1 HACER

ESCRIBIR("Ingrese el numero : ") LEER(v[ i ])

FIN\_PARA

PARA i DESDE 0 HASTA cantidad-1 HACER

SI ( v[ i ] > mayor )

ENTONCES

mayor  $\leftarrow$  v[ i ]

FIN\_SI

SI ( v[ i ] < menor )

ENTONCES

Menor  $\leftarrow$  v[ i ]

FIN\_SI

suma  $\leftarrow$  suma + v[ i ]

FIN\_PARA

Promedio  $\leftarrow$  suma / cantidad

ESCRIBIR("El numero mayor es : ", mayor )

ESCRIBIR("El numero menor es : ", menor )

ESCRIBIR("El promedio de los numeros : ", promedio )

FIN\_ACCION

#### Solucionario 2:

##### VARIABLES

ENTERO : \*v, i, cantidad, estado, numero

ACCION Hallar\_posicion\_numero

ESCRIBIR("Ingrese la cantidad de elementos del vector : ") LEER(cantidad)

estado  $\leftarrow$  0

v  $\leftarrow$  reservar\_espacio ENTERO [ cantidad ]



```
PARA i DESDE 0 HASTA cantidad-1 HACER
    ESCRIBIR("Ingrese el numero : ") LEER(v[ i ])
FIN_PARA

ESCRIBIR("Ingrese el numero a buscar : ") LEER(numero)

i ← 0

MIENTRAS ( i < cantidad y estado = 0 )
    HACER
        SI ( v [ i ] = numero )
            ENTONCES
                Estado ← 1
                ESCRIBIR("El numero ", numero ," se encuentra en la posicion ", i ," del vector")
                i ← i + 1
            FIN_SI
        FIN_MIENTRAS

SI ( estado = 0 )
    ENTONCES
        ESCRIBIR("El numero no se encuentra en el vector ")
    FIN_SI

FIN_ACCION
```

### Solucionario 3:

```
VARIABLES
    ENTERO : *v, i, j, cantidad, aux
ACCION Ordenar_numero_sin_repeticiones
    ESCRIBIR("Ingrese la cantidad de elementos del vector : ") LEER(cantidad)

v ← RESERVAR_ESPACIO ENTERO [ cantidad ]

PARA i DESDE 0 HASTA cantidad-1 HACER
    ESCRIBIR("Ingrese el numero : ") LEER(v[ i ])
FIN_PARA

PARA i DESDE 0 HASTA cantidad-1 HACER
    PARA j DESDE 0 HASTA cantidad-2 HACER
        SI ( v[ j ] > v[ j+1 ] )
            ENTONCES
                aux ← v[ j ]
                v[ j ] ← v[ j+1 ]
                v[ j+1 ] ← aux
            FIN_SI
        FIN_PARA
    FIN_PARA

PARA i DESDE 0 HASTA cantidad-1 HACER
    aux ← v[ i ]
    PARA j DESDE i+1 HASTA cantidad-1 HACER
        SI ( aux = v[ j ] )
            ENTONCES
                v[ j ] ← -99
            FIN_SI
    FIN_PARA
```



```
FIN_PARA
FIN_PARA

ESCRIBIR("Los elementos del vector son : ")

PARA i DESDE 0 HASTA cantidad-1 HACER
    SI ( v[ i ] < > -99 )
        ENTONCES
            ESCRIBIR( v[ i ] , " " )
    FIN_SI
FIN_PARA

FIN_ACCION
```

#### **Solucionario 4:**

```
VARIABLES
    ENTERO : i, fibo, fibo1, fibo2
    ARREGLO ENTERO : v[ 20 ]
ACCION Crear_vector_serie_fibonacci

    fibo1 ← 0
    fibo2 ← 1
    v[ 0 ] ← 1

    MIENTRAS ( i < 20 )
        HACER
            fibo ← fibo1 + fibo2
            v[ i ] ← fibo
            fibo1 ← fibo2
            fibo2 ← fibo
            i ← i +1
    FIN_MIENTRAS

    ESCRIBIR("Los elementos del vector es : ")

    PARA i DESDE 0 HASTA 19 HACER
        ESCRIBIR( v[ i ] , " " )
    FIN_PARA

FIN_ACCION
```

#### **Solucionario 5:**

```
VARIABLES
    ENTERO : i, estado
    ARREGLO ENTERO : v[11]
    CARÁCTER : caracter
ACCION Buscar_palabra
    ESCRIBIR("Ingrese el texto : ")

    i ← 0
    estado ← 0

    HACER
        carácter ← ESPERAR( )
        ESCRIBIR(carácter)
```



```
SI ( v[ i ] = caracter )
  ENTONCES
    i ← i + 1
FIN_SI
```

```
SI ( i = 11 )
  ENTONCES
    estado ← 1
FIN_SI
```

```
MIENTRAS ( estado = 0 )
```

```
  ESCRIBIR("La palabra ALGORITMICA ha sido detectada...")
```

```
FIN_ACCION
```

### Solucionario 6:

#### VARIABLES

```
  ENTERO : *V1, *V2, *V3
```

```
  ENTERO : i, j, k, h
```

```
  ENTERO : g1, g2, g3
```

```
ACCION Hallar_producto_polinomios
```

```
  ESCRIBIR("Ingrese el grado del primer polinomio : ") LEER(g1)
```

```
  v1 ← RESERVAR_ESPACIO ENTERO [ g1+1 ]
```

```
  PARA i DESDE 0 HASTA g1 HACER
```

```
    ESCRIBIR("Ingrese el coeficiente : ") LEER(v1[ i ])
```

```
  FIN_PARA
```

```
  ESCRIBIR("Ingrese el grado del segundo polinomio : ") LEER(g2)
```

```
  v2 ← RESERVAR_ESPACIO ENTERO [ g2+1 ]
```

```
  PARA i DESDE 0 HASTA g2 HACER
```

```
    ESCRIBIR("Ingrese el coeficiente : ") LEER(v2[ i ])
```

```
  FIN_PARA
```

```
  g3 ← g1 + g2
```

```
  v3 ← RESERVAR_ESPACIO ENTERO [ g3+1 ]
```

```
  PARA i DESDE 0 HASTA g3 HACER
```

```
    v3[ i ] ← 0
```

```
  FIN_PARA
```

```
  h ← 0
```

```
  PARA i DESDE g2 HASTA 0 HACER
```

```
    k ← 0
```

```
    PARA j ← g1 HASTA 0 HACER
```

```
      v3[ g3-k-h ] ← v3[ g3-k-h ] + v2[ i ]*v1[ j ]
```

```
      k ← k + 1
```

```
    FIN_PARA
```

```
    h ← h + 1
```

```
  FIN_PARA
```

```
  ESCRIBIR("El producto de los polinomios es : ")
```

```
  PARA i DESDE 0 HASTA g3-1 HACER
```



```
    ESCRIBIR(v3[i]," x",g3-I," + ")
  FIN_PARA
  ESCRIBIR(v3[g3]," x0")

FIN_ACCION
```

### Solucionario 7:

```
VARIABLES
  ENTERO : *v, n, i, j, aux, med
ACCION Hallar_mediana_numeros
  ESCRIBIR("Ingrese la cantidad de números a ingresar : ") LEER(n)

  PARA i DESDE 0 HASTA n-1 HACER
    ESCRIBIR("Ingrese el numero : ") LEER(v[ i ])
  FIN_PARA

  PARA i DESDE 0 HASTA n-1 HACER
    PARA j DESDE 0 HASTA n-2 HACER
      SI ( v[ j ] > v[ j+1 ] )
        ENTONCES
          aux ← v[ j ]
          v[ j ] ← v[ j+1 ]
          v[ j+1 ] ← aux
        FIN_SI
    FIN_PARA
  FIN_PARA

  SI ( n MOD 2 = 0 )
    ENTONCES
      med ← ( v[ n/2-1 ] + v[ n/2 ] )/2
    SINO
      med ← v[ (n-1)/2 ]
    FIN_SI

  ESCRIBIR("La mediana de los numeros es : ", med )

FIN_ACCION
```

### Solucionario 8:

```
VARIABLES
  ENTERO : *v, n, i, j, cont1, cont2, aux, mayor, moda
ACCION Hallar_moda
  ESCRIBIR("Ingrese la cantidad de elementos del vector : ") LEER(n)

  V ← RESERVAR_ESPACIO ENTERO [ n ]

  ESCRIBIR("Ingrese los elementos del vector : ")

  moda ← 0
  PARA i DESDE 0 HASTA n-1 HACER
    ESCRIBIR("V[" , i ,"] = ") LEER(v[ i ])
  FIN_PARA

  PARA i DESDE 0 HASTA n-1 HACER
    aux ← v[ i ]
```



```
cont1 ← 0
PARA j DESDE 0 HASTA n-1 HACER
    SI ( aux = V[ j ] )
        ENTONCES
            cont1 ← cont1 + 1
        FIN_SI
    FIN_PARA

    SI ( cont1 > mayor )
        ENTONCES
            mayor ← cont1
            moda ← aux
        FIN_SI
    FIN_PARA

cont2 ← 0

PARA i DESDE 0 HASTA n-1 HACER
    aux ← V[ i ]
    SI ( aux < > '' )
        ENTONCES
            cont1 ← 0
            PARA j 0 DESDE n-1 HACER
                SI ( aux = V[ j ] )
                    ENTONCES
                        cont1 ← cont1 + 1
                        V[ j ] ← ''
                FIN_SI
            FIN_PARA

            SI ( cont1 = mayor )
                ENTONCES
                    cont2 ← cont2 + 1
                FIN_SI
        FIN_PARA

    SI ( cont2 = 1 )
        ENTONCES
            ESCRIBIR("La moda es : ", moda )
        SINO
            ESCRIBIR("No existe moda ")
        FIN_SI

FIN_ACCION
```

### Solucionario 9:

#### VARIABLES

ENTERO : \*\*m, n, i, j, cont

#### ACCION Verificar\_matriz\_simetrica

ESCRIBIR("Ingrese el numero de filas de la matriz cuadrada : ") LEER(n)

m ← RESERVAR\_ESPACIO ENTERO \*[ n ]

PARA i DESDE 0 HASTA n-1 HACER

m[ i ] ← RESERVAR\_ESPACIO ENTERO [ n ]

FIN\_PARA



```
ESCRIBIR("Ingrese los elementos de la matriz : ")

PARA i DESDE 0 HASTA n-1 HACER
  PARA j DESDE 0 HASTA n-1 HACER
    ESCRIBIR("m[" , i , "][" , j , " ] = ")  LEER(m[ i ][ j ])
  FIN_PARA
FIN_PARA

PARA i DESDE 0 HASTA n-1 HACER
  PARA j DESDE 0 HASTA n-1 HACER
    SI ( m[ i ][ j ] = m[ j ][ i ] )
      ENTONCES
        cont ← cont + 1
      FIN_SI
    FIN_PARA
  FIN_PARA

SI ( cont = n*n )
  ENTONCES
    ESCRIBIR("La matriz es simetrica")
  SINO
    ESCRIBIR("La matriz no es simetrica")
  FIN_SI

FIN_ACCION
```

### Solucionario 10:

```
VARIABLES
  ENTERO : **A, **B, n, i, j
ACCION Hallar_transpuesta_matriz
  ESCRIBIR("Ingrese el numero de filas de la matriz cuadrada : ")  LEER(n)

  A ← RESERVAR_ESPACIO ENTERO *[ n ]
  PARA i DESDE 0 HASTA n-1 HACER
    A[ i ] ← RESERVAR_ESPACIO ENTERO [ n ]
  FIN_PARA

  B ← RESERVAR_ESPACIO ENTERO *[ n ]
  PARA i DESDE 0 HASTA n-1 HACER
    B[ i ] ← RESERVAR_ESPACIO ENTERO [ n ]
  FIN_PARA

  ESCRIBIR("Ingrese los elementos de la matriz : ")

  PARA i DESDE 0 HASTA n-1 HACER
    PARA j DESDE 0 HASTA n-1 HACER
      ESCRIBIR("A[" , i , "][" , j , " ] = ")  LEER(A[ i ][ j ])
    FIN_PARA
  FIN_PARA

  PARA i DESDE 0 HASTA n-1 HACER
    PARA j DESDE 0 HASTA n-1 HACER
      B[ i ][ j ] ← A[ j ][ i ]
    FIN_PARA
  FIN_PARA
```



```
ESCRIBIR("La matriz transpuesta de A es : ")
```

```
PARA i DESDE 0 HASTA n-1 HACER  
  PARA j DESDE 0 HASTA n-1 HACER  
    ESCRIBIR(B[i][j] , " ")  
  FIN_PARA  
FIN_PARA
```

```
FIN_ACCION
```

### Solucionario 11:

VARIABLES

```
ENTERO : **A, **B, **C, f1, c1, f2, c2, i, j  
ENTERO : suma, h, k
```

ACCION Producto\_matrices

```
ESCRIBIR("Ingrese el numero de filas de la primera matriz cuadrada : ") LEER(f1)  
ESCRIBIR("Ingrese el numero de columnas de la primera matriz cuadrada : ") LEER(c1)
```

```
A ← RESERVAR_ESPACIO ENTERO *[ c1 ]  
PARA i DESDE 0 HASTA c1-1 HACER  
  A[ i ] ← RESERVAR_ESPACIO ENTERO [ f1 ]  
FIN_PARA
```

```
ESCRIBIR("Ingrese los elementos de la matriz A : ")
```

```
PARA i DESDE 0 HASTA f1-1 HACER  
  PARA j DESDE 0 HASTA c1-1 HACER  
    ESCRIBIR("A[" , i , "][" , j , " ] = ") LEER(A[ i ][ j ])   
  FIN_PARA  
FIN_PARA
```

```
ESCRIBIR("Ingrese el numero de filas de la segunda matriz cuadrada : ") LEER(f2)  
ESCRIBIR("Ingrese el numero de columnas de la segunda matriz cuadrada : ") LEER(c2)
```

```
B ← RESERVAR_ESPACIO ENTERO *[ c2 ]  
PARA i DESDE 0 HASTA c2-1 HACER  
  B[ i ] ← RESERVAR_ESPACIO ENTERO [ f2 ]  
FIN_PARA
```

```
ESCRIBIR("Ingrese los elementos de la matriz B : ")
```

```
PARA i DESDE 0 HASTA f2-1 HACER  
  PARA j DESDE 0 HASTA c2-1 HACER  
    ESCRIBIR("B[" , i , "][" , j , " ] = ") LEER(B[ i ][ j ])   
  FIN_PARA  
FIN_PARA
```

```
C ← RESERVAR_ESPACIO ENTERO *[ c2 ]  
PARA i DESDE 0 HASTA c2-1 HACER  
  C[ i ] ← RESERVAR_ESPACIO ENTERO [ f1 ]  
FIN_PARA
```

```
SI ( c1 = f2 )  
  ENTONCES  
    h ← 0
```





```

HACER
  k ← 0
  HACER
    suma ← 0
    PARA i DESDE 0 HASTA c1-1 HACER
      suma ← suma + A[ h ][ i ]*B[ i ][ k ]
    FIN_PARA
    C[ h ][ k ] ← suma
    k ← k + 1
  MIENTRAS ( k < c2 )
  h ← h + 1
MIENTRAS ( h < f1 )

ESCRIBIR("El resultado de la multiplicacion de matrices es : ")

PARA i DESDE 0 HASTA f1-1 HACER
  PARA j DESDE 0 HASTA c2-1 HACER
    ESCRIBIR(C[ i ][ j ], " ")
  FIN_PARA
FIN_PARA

SINO
  ESCRIBIR("No se puede realizar el producto de las matrices")
FIN_SI

FIN_ACCION

```

## Solucionario 12:

```

VARIABLES
  ENTERO : **M, fila, columna, i, j, h, opcion
  ENTERO : x, y, aux1, aux2
ACCION Rotar_matriz_90°
  x ← 27
  y ← 7

  HACER
    HACER
      LIMPIAR_PANTALLA( )
      UBICAR( x,y ) ESCRIBIR("*****")
      UBICAR( x,y+1 ) ESCRIBIR("*          MENU          *")
      UBICAR( x,y+2 ) ESCRIBIR("* Ingresar datos.....[1] *")
      UBICAR( x,y+3 ) ESCRIBIR("* Mostrar datos.....[2] *")
      UBICAR( x,y+4 ) ESCRIBIR("* Rotar 90g Matriz.....[3] *")
      UBICAR( x,y+5 ) ESCRIBIR("* Mostrar.....[4] *")
      UBICAR( x,y+6 ) ESCRIBIR("*****")
      UBICAR( x,y+7 ) ESCRIBIR("* Opcion.....[ ] *")
      UBICAR( x,y+8 ) ESCRIBIR("*****")
      UBICAR( x+24,y+7 ) LEER(opcion)
    MIENTRAS( opcion <1 y opcion >4)
  LIMPIAR_PANTALLA

  SEGÚN_SEA ( opcion )
  HACER
    CASO 1: UBICAR( x,y ) ESCRIBIR("Ingrese la cantidad de filas : ") LEER(Fila)
            UBICAR( x,y+1 ) ESCRIBIR("Ingrese la cantidad de columnas : ") LEER(columna)

```



```
M ← RESERVAR_ESPACIO ENTERO *[ columna ]
PARA i DESDE 0 HASTA columna-1 HACER
    M[ i ] ← RESERVAR_ESPACIO ENTERO [ fila ]
FIN_PARA

LIMPIAR_PANTALLA( )

PARA i DESDE 0 HASTA fila-1 HACER
    PARA j DESDE 0 HASTA columna-1 HACER
        SI ( i = 0 o i = fila-1 )
            ENTONCES
                UBICAR( x,y ) ESCRIBIR("M[" , i , "]" , j , "]" = ") LEER(M[ i ][ j ])
            SINO
                SI ( j = 0 o j = columna-1 )
                    ENTONCES
                        UBICAR( x,y ) ESRIBIR("M[" , i , "]" , j , "]" = ") LEER(M[ i ][ j ])
                    SINO
                        M[ i ][ j ] ← 0
                FIN_SI
            FIN_SI
        FIN_PARA
    FIN_PARA

CASO 2: PARA i DESDE 0 HASTA fila-1 HACER
    PARA j DESDE 0 HASTA columna-1 HACER
        UBICAR( x+10+j,y+i ) ESCRIBIR(M[ i ][ j ])
    FIN_PARA
FIN_PARA

CASO 3: PARA h DESDE 0 HASTA columna-1 HACER
    aux1 ← M[0][0]
    PARA i DESDE 0 HASTA columna-2 HACER
        M[0][ i ] ← M[0][ i+1 ]
    FIN_PARA
    aux2 ← M[ fila-1 ][0]

    PARA i ← fila-1 HASTA 2 HACER
        M[ i ][0] ← M[ i-1 ][0]
    FIN_PARA
    M[1][0] ← aux1

    aux1 ← M[ fila-1 ][ columna-1 ]
    PARA i DESDE columna-1 HASTA 2 HACER
        M[ fila-1 ][ i ] ← M[ fila-1 ][ i-1 ]
    FIN_PARA
    M[ fila-1 ][1] ← aux2

    PARA i DESDE 0 HASTA fila-3 HACER
        M[ i ][ columna-1 ] ← M[ i+1 ][ columna-1 ]
    FIN_PARA
    M[ fila-2 ][ columna-1 ] ← aux1

FIN_PARA

UBICAR( x,y ) ESCRIBIR("¡ Se ha realizado la rotacion !")
```



FIN\_SEGUN

MIENTRAS ( opcion < > 4 )

FIN\_ACCION