

1. EJERCICIO SUMA DE LA MATRIZ CON UN SUBPROCESO

Subproceso `mostrarMatriz(M,A,B)`

```
Para i<-1 Hasta A Con Paso 1 Hacer
    Para j<-1 Hasta B Con Paso 1 Hacer
        Escribir M[I,J], " " Sin Saltar
    FinPara
    Escribir ""
FinPara
```

FinSubProceso

Proceso SUMAMATRIZ

```
Dimension M[50,50]
Escribir "INGRESE CANTIDAD DE FILAS Y COLUMNAS DE LA MATRIZ";
Escribir "NO. DE FILAS:"
Leer A
Escribir "NO. DE COLUMNAS:"
Leer B
Escribir "INGRESE DATOS DE LA 1A MATRIZ"
Para i<-1 Hasta A Con Paso 1 Hacer
    Para j<-1 Hasta B Con Paso 1 Hacer
        Escribir "INGRESE DATO DE LA POSICION ",I," ",J
        Leer M[I,J]
        SUMA<-SUMA+M[I,J]
    FinPara
FinPara
Escribir ""
```

mostrarMatriz(M,A,B)

Escribir ""

Escribir "LA SUMA DE LA MATRIZ ES: ",SUMA

FinProceso

2. EJERCICIO SUMA DE LA MATRIZ CON DOS SUBPROCESO

SubProceso inicializarMatriz(matriza,columna,fila)

Escribir "Ingrese Datos de Matriz A ";

Para i<-1 Hasta columna Con Paso 1 Hacer

Para j<-1 Hasta fila Con Paso 1 Hacer

Escribir "INGRESE DATO DE LA POSICION ",i," ",j

leer matriza[i,j];

Finpara

FinPara

FinSubProceso

Subproceso mostrarMatriz(matriza,columna,fila)

Para i<-1 Hasta columna Con Paso 1 Hacer

Para j<-1 Hasta fila Con Paso 1 Hacer

res<-res+matriza[i,j]

FinPara

Escribir ""

FinPara

escribir "la suma de la matriz es:",res

FinSubProceso

Proceso sumamatrices

 Dimension matriz(10,10)

 Escribir "Ingrese Tamaño de Columna ";

 leer columna ;

 Escribir " Ingrese Tamaño de Fila ";

 leer fila ;

 inicializarMatriz(matriz,columna,fila)

 mostrarMatriz(matriz,columna,fila)

 Escribir ""

 Para i<-1 Hasta columna Con Paso 1 Hacer

 Para j<-1 Hasta fila Con Paso 1 Hacer

 Escribir Matriz[i,j], " ", Sin Saltar;

 FinPara

 Escribir ""

 FinPara

FinProceso

3. Con varios subprocesos

SubProceso printMatriz(matriz, n)

Para i<-1 Hasta n Con Paso 1 Hacer

Para j<-1 Hasta n Con Paso 1 Hacer

 Escribir Sin Saltar matriz[i,j], " "

Fin Para

 Escribir ""

 Escribir ""

Fin Para

FinSubProceso

SubProceso printTranspuesta(matriz, n)

Para i<-1 Hasta n Con Paso 1 Hacer

Para j<-i Hasta n Con Paso 1 Hacer

 aux <- matriz[i,j]

 matriz[i,j] <- matriz[j,i]

 matriz[j,i] <- aux

FinPara

Fin Para

Escribir "Matriz transpuesta"

Escribir ""

printMatriz(matriz, n)

FinSubProceso

SubProceso testSimetrica(matriz, n)

cent <- 1

para i <- 1 hasta n con paso 1 hacer

para j <- i hasta n con paso 1 hacer

si matriz[i,j] <> matriz[j,i] entonces

Escribir "La matriz NO es simétrica"

j <- n

i <- n

cent <- 10

FinSi

FinPara

FinPara

si cent = 1 Entonces

Escribir "La matriz ES simétrica"

FinSi

FinSubProceso

SubProceso cent <- verificaDiagUNOS(matriz, n)

Para i<-1 Hasta n Con Paso 1 Hacer //CICLO QUE RECORRE LA DIAGONAL para verificar que todos sean 1

```
cent <- 0
```

```
si matriz[i,i] <> 1 Entonces
```

```
cent <- 1
```

```
i <- n+1
```

```
FinSi
```

```
FinPara
```

```
FinSubProceso
```

```
SubProceso cent <- verificaDiagNOceros(matriz, n)
```

```
Para i<-1 Hasta n Con Paso 1 Hacer //CICLO QUE RECORRE LA DIAGONAL para verificar que  
todos sean distintos de 0
```

```
cent <- 0
```

```
si matriz[i,i] = 0 Entonces
```

```
cent <- 1
```

```
i <- n+1
```

```
FinSi
```

```
FinPara
```

```
FinSubProceso
```

```
SubProceso cent <- verificaTriangSup(matriz, n)
```

```
Para i<-1 Hasta n-1 Con Paso 1 Hacer // CICLO QUE RECORRE EL TRIANGULO SUPERIOR (sin la  
diagonal) para verificar que todos sean 0
```

```
Para j<-i+1 Hasta n Con Paso 1 Hacer
```

```
cent <- 0
```

```
si matriz[i,j] <> 0 Entonces
```

cent <- 1

i <- n+1

j <- n+1

FinSi

Fin Para

Fin Para

FinSubProceso

SubProceso cent <- verificaTriangInf(matriz, n)

Para i<-2 Hasta n Con Paso 1 Hacer // CICLO QUE RECORRE EL TRIANGULO INFERIOR (sin la diagonal) para verificar que todos sean 0

Para j<-1 Hasta i-1 Con Paso 1 Hacer

cent <- 0

si matriz[i,j] <> 0 Entonces

cent <- 1

i <- n+1

j <- n+1

FinSi

Fin Para

Fin Para

FinSubProceso

SubProceso testIdentidad(matriz, n)

cent <- verificaDiagUNOS(matriz, n)

Si cent <> 0 Entonces

 Escribir ">>> La matriz NO es identidad <<<"

Sino

 cent <- verificaTriangSup(matriz, n)

 Si cent <> 0 Entonces

 Escribir ">>> La matriz NO es identidad <<<"

 Sino

 cent <- verificaTriangInf(matriz, n)

 Si cent <> 0 Entonces

 Escribir ">>> La matriz NO es identidad <<<"

 Sino

 Escribir ">>> La matriz ES identidad <<<"

 Fin Si

 Fin Si

Fin Si

FinSubProceso

SubProceso testDiagonal(matriz, n)

 cent <- verificaDiagNOceros(matriz, n)

 Si cent <> 0 Entonces

 Escribir ">>> La matriz NO es Diagonal <<<"

 Sino

 cent <- verificaTriangSup(matriz, n)

 Si cent <> 0 Entonces

 Escribir ">>> La matriz NO es Diagonal <<<"

 Sino

```
cent <- verificaTriangInf(matriz, n)
```

```
Si cent <> 0 Entonces
```

```
    Escribir ">>> La matriz NO es Diagonal <<<"
```

```
Sino
```

```
    Escribir ">>> La matriz ES Diagonal <<<"
```

```
Fin Si
```

```
Fin Si
```

```
Fin Si
```

```
FinSubProceso
```

```
SubProceso testTriangSup(matriz, n)
```

```
cent <- verificaTriangInf(matriz, n)
```

```
Si cent <> 0 Entonces
```

```
    Escribir ">>> La matriz NO es Triangular Superior <<<"
```

```
sino
```

```
    Escribir ">>> La matriz ES Triangular Superior <<<"
```

```
finsi
```

```
FinSubProceso
```

```
SubProceso testTriangInf(matriz, n)
```

```
cent <- verificaTriangSup(matriz, n)
```

```
Si cent <> 0 Entonces
```

```
    Escribir ">>> La matriz NO es Triangular Inferior <<<"
```

sino

 Escribir ">>> La matriz ES Triangular Inferior <<<"

finsi

FinSubProceso

SubProceso menu()

 Escribir " MENU PRINCIPAL"

 Escribir ""

 Escribir "para verificar si la matriz es Simétrica digite 1"

 Escribir "para verificar si la matriz es Identidad digite 2"

 Escribir "para verificar si la matriz es Diagonal digite 3"

 Escribir "para verificar si la matriz es Triangular Superior digite 4"

 Escribir "para verificar si la matriz es Triangular Inferior digite 5"

 Escribir "para imprimir la matriz digite 6"

 Escribir "para imprimir la matriz Transpuesta digite 7"

FinSubProceso

Proceso matrizCuadrada

Escribir "ingrese el tamaño de la matriz cuadrada"

Leer n

Dimension matriz[n,n]

Para i<-1 Hasta n Con Paso 1 Hacer

Para j<-1 Hasta n Con Paso 1 Hacer

// Escribir "ingrese el dato de la fila ", i, " columna ", j

// Leer matriz[i,j]

matriz[i,j] <- azar (2)

Fin Para

Fin Para

// printMatriz(matriz, n)

resp <- 's'

Mientras resp = 's' | resp = 'S' Hacer

menu()

leer menuSelect

Segun menuSelect Hacer

1:

printMatriz(matriz, n)

Escribir ""

testSimetrica(matriz, n)

2:

printMatriz(matriz, n)

Escribir ""

testIdentidad(matriz, n)

3:

```
printMatriz(matriz, n)
Escribir ""
testDiagonal(matriz, n)
```

4:

```
printMatriz(matriz, n)
Escribir ""
testTriangSup(matriz, n)
```

5:

```
printMatriz(matriz, n)
Escribir ""
testTriangInf(matriz, n)
```

6:

```
Escribir ""
printMatriz(matriz, n)
```

7:

```
printMatriz(matriz, n)
Escribir ""
printTranspuesta(matriz, n)
```

De Otro Modo:

```
Escribir "acaba de marcar un número equivocado"
```

```
Fin Segun
```

```
Escribir "desea hacer otra verificación? s/n"
```

```
Leer resp
```

```
Borrar Pantalla
```

```
FinMientras
```

```
FinProceso
```