

Introducción a la Programación I – Práctico 8 (Matrices)

```
Program e1p8;  
{Carga e imprime los datos de una matriz}  
  
Const maxfilas=5;  
      maxcolumnas=5;  
  
Type TMat=array[1..maxfilas,1..maxcolumnas]of integer;  
  
Procedure CargarMatriz (var matrizCar:TMat);  
{Carga la matriz con numeros enteros ingresados por el usuario}  
var fila,columna,entero:integer;  
begin  
  for fila:=1 to maxfilas do  
    for columna:=1 to maxcolumnas do  
      begin  
        write('Ingrese un Numero ');  
        read(entero);  
        matrizCar[fila,columna]:=entero;  
      end;  
end;  
  
Procedure MostrarMatriz (matrizMos:TMat);  
{Muestra la matriz por pantalla}  
var fila,columna:integer;  
begin  
  for fila:=1 to maxfilas do  
    begin  
      writeln('');  
      writeln('Fila: ',fila);  
      for columna:=1 to maxcolumnas do  
        begin  
          write(matrizMos[fila,columna],', ');  
        end;  
    end;  
end;  
  
Var matrizenteros:TMat;  
Begin  
  CargarMatriz (matrizenteros);  
  MostrarMatriz (matrizenteros);  
end.  
  
/*****/  
  
Program e2p8;  
{Crea la matriz identidad, y la imprime por pantalla}  
  
Const maxfilas=5;
```

```
maxcolumnas=5;
```

```
Type TMat=array[1..maxfilas,1..maxcolumnas]of integer;
```

```
Procedure CargarMatriz (var matrizCar:TMat);  
{Carga la matriz con numeros enteros ingresados por el usuario}
```

```
var fila,columna:integer;
```

```
begin
```

```
  for fila:=1 to maxfilas do
```

```
    for columna:=1 to maxcolumnas do
```

```
      if (fila=columna) then
```

```
        matrizCar[fila,columna]:=1
```

```
      else
```

```
        matrizCar[fila,columna]:=0;
```

```
end;
```

```
Procedure MostrarMatriz (matrizMos:TMat);
```

```
{Muestra la matriz por pantalla}
```

```
var fila,columna:integer;
```

```
begin
```

```
  for fila:=1 to maxfilas do
```

```
    begin
```

```
      writeln('');
```

```
      writeln('Fila: ',fila);
```

```
      for columna:=1 to maxcolumnas do
```

```
        begin
```

```
          write(matrizMos[fila,columna],', ');
```

```
        end;
```

```
      end;
```

```
end;
```

```
Var matrizenteros:TMat;
```

```
Begin
```

```
  CargarMatriz (matrizenteros);
```

```
  MostrarMatriz (matrizenteros);
```

```
end.
```

```
/******
```

```
Program e3p8;
```

```
{Carga e imprime los datos de una matriz}
```

```
Const maxfilas=10;
```

```
  maxcolumnas=10;
```

```
  SINVALOR=0;
```

```
Type TMat=array[1..maxfilas,1..maxcolumnas]of integer;
```

```
Procedure CargarMatriz (var matrizCar:TMat);
```

```
{Carga la matriz con numeros enteros ingresados por el usuario}
```

```
var fila,columna,entero:integer;
```

```
begin
  for fila:=1 to maxfilas do
    for columna:=1 to maxcolumnas do
      begin
        write('Ingrese un Numero ');
        read(entero);
        matrizCar[fila,columna]:=entero;
      end;
    end;

Procedure SumarColumna (matrizSumC:TMat);
{Suma la columna de la matiz que el usuario ingrese}
var fila,columna,suma:integer;
begin
  columna:=SINVALOR;
  suma:=SINVALOR;
  write('Ingrese La Columna Que Desea Sumar: ');
  read(columna);
  if (columna>SINVALOR) and (columna<=maxcolumnas) then
    begin
      for fila:=1 to maxfilas do
        suma:=suma+matrizSumC[fila,columna];
        writeln('Suma De La Columna: ',suma);
      end;
    end;

Procedure SumarFila (matrizSumF:TMat);
{Suma la fila de la matiz que el usuario ingrese}
var fila,columna,suma:integer;
begin
  fila:=SINVALOR;
  suma:=SINVALOR;
  write('Ingrese La Fila Que Desea Sumar: ');
  read(fila);
  if (fila>SINVALOR) and (fila<=maxcolumnas) then
    begin
      for columna:=1 to maxcolumnas do
        suma:=suma+matrizSumF[fila,columna];
        writeln('Suma De La Fila: ',suma);
      end;
    end;

Procedure PromedioFila (matrizPromF:TMat);
{Saca el Promedio de la fila de la matiz que el usuario ingrese}
var fila,columna,suma:integer;promedio:real;
begin
  fila:=SINVALOR;
  suma:=SINVALOR;
  write('Ingrese La Fila Que Desea Sacar El Promedio: ');
  read(fila);
```

```
if (fila>SINVALOR) and (fila<=maxfilas) then  
  begin  
    for columna:=1 to maxcolumnas do  
      suma:=suma+matrizPromF[filas,columna];  
      promedio:=(suma / maxcolumnas);  
      writeln('Promedio De La Fila: ',(promedio):2:2);  
    end;{:2:2, es para que lo imprima de la forma 2.50}  
end;
```

```
Procedure PromedioColumna (matrizPromC:TMat);  
{Saca el Promedio de la columna de la matriz que el usuario ingrese}
```

```
var fila,columna,suma:integer;promedio:real;  
begin  
  columna:=SINVALOR;  
  suma:=SINVALOR;  
  write('Ingrese La Columna Que Desea Sacar El Promedio: ');  
  read(columna);  
  if (columna>SINVALOR) and (columna<=maxcolumnas) then  
    begin  
      for fila:=1 to maxfilas do  
        suma:=suma+matrizPromC[filas,columna];  
        promedio:=(suma / maxfilas);  
        writeln('Promedio De La Columna: ',(promedio):2:2);  
      end;  
    end;
```

```
Procedure MostrarMatriz (matrizMos:TMat);  
{Muestra la matriz por pantalla}
```

```
var fila,columna:integer;  
begin  
  for fila:=1 to maxfilas do  
    begin  
      writeln('');  
      writeln('Fila: ',fila);  
      for columna:=1 to maxcolumnas do  
        begin  
          write(matrizMos[filas,columna],', ');  
        end;  
      end;  
    end;  
end;
```

```
Procedure Menu (matrizMenu:TMat);  
{Dependiendo de la opcion elegida, realiza distintas acciones}
```

```
var opcion:integer;  
begin  
  opcion:=-1;  
  while (opcion<>SINVALOR) do  
    begin  
      writeln('Para Sumar Una Columna Ingrese 1');  
      writeln('Para Sumar Una Fila Ingrese 2');  
    end;
```

```
writeln('Para Calcular El Promedio De Una Fila Ingrese 3');
writeln('Para Calcular El Promedio De Una Columna Ingrese 4');
writeln('Para Imprimir La Matriz Ingrese 5');
writeln('Para Salir Ingrese 0');
read(opcion);
if opcion=1 then
    SumarColumna(matrizMenu);
if opcion=2 then
    SumarFila(matrizMenu);
if opcion=3 then
    PromedioFila(matrizMenu);
if opcion=4 then
    PromedioColumna(matrizMenu);
if opcion=5 then
    MostrarMatriz(matrizMenu);
end;
end;

Var matrizenteros:TMat;
Begin
    CargarMatriz(matrizenteros);
    Menu(matrizenteros);
end.

/*****/

Program e6p8;
{Carga una matriz y verifica si una matriz es palindroma}

Const maxfilas=10;
    maxcolumnas=10;

Type TMat=array[1..maxfilas,1..maxcolumnas]of integer;

Procedure CargarMatriz(var matrizCar:TMat);
{Carga la matriz con numeros enteros ingresados por el usuario}
var fila,columna,entero:integer;
begin
    for fila:=1 to maxfilas do
        for columna:=1 to maxcolumnas do
            begin
                write('Ingrese un Numero ');
                read(entero);
                matrizCar[fila,columna]:=entero;
            end;
        end;
    end;

Function FilasCapicuas(matrizCapicF:TMat):boolean;
{Devuelve true si la matriz es capicua en todas sus filas, si no devuelve false}
```

```
var fila, columna, ultimo:integer; capicuafila:boolean;
begin
  fila:=1;
  capicuafila:=true;
  while (capicuafila) and (fila<maxfilas) do
    begin
      columna:=1;
      ultimo:=maxcolumnas;
      if capicuafila then
        begin
          while (columna<ultimo) and
(matrizCapicF[fila, columna]=matrizCapicF[fila, ultimo]) do
            begin
              columna:=columna+1;
              ultimo:=ultimo-1;
            end;
          if (matrizCapicF[fila, columna]<>matrizCapicF[fila, ultimo]) then
            capicuafila:=false;
          end;
          fila:=fila+1;
        end;
      FilasCapicuas:=capicuafila;
    end;

Function ColumnasCapicuas (matrizCapicC:TMat):boolean;
  {Devuelve true si la matriz es capicua en todas sus columnas, si no
  devuelve false}
var fila, columna, ultimo:integer; capicuacolumna:boolean;
begin
  columna:=1;
  capicuacolumna:=true;
  while (capicuacolumna) and (columna<maxcolumnas) do
    begin
      fila:=1;
      ultimo:=maxfilas;
      if capicuacolumna then
        begin
          while (fila<ultimo) and
(matrizCapicC[fila, columna]=matrizCapicC[ultimo, columna]) do
            begin
              fila:=fila+1;
              ultimo:=ultimo-1;
            end;
          if (matrizCapicC[fila, columna]<>matrizCapicC[ultimo, columna]) then
            capicuacolumna:=false;
          end;
          columna:=columna+1;
        end;
      ColumnasCapicuas:=capicuacolumna;
    end;

```

end;

```
Function VerificaCapicua (matrizVer:TMat):boolean;  
{Devuelve true si la matriz es palindroma, si no devuelve false}  
begin  
  if FilasCapicuas (matrizVer) and ColumnasCapicuas (matrizVer) then  
    VerificaCapicua:=true  
  else  
    VerificaCapicua:=false;  
end;
```

```
Var matrizenteros:TMat;  
Begin  
  CargarMatriz (matrizenteros);  
  if VerificaCapicua (matrizenteros) then  
    writeln('La Matriz Es Palindroma')  
  else  
    writeln('La Matriz No Es Palindroma');  
end.
```

/*****/

```
Program e7p8;  
{ordena por seleccion cada fila de la matriz, de menor a mayor, y luego lo  
imprime por pantalla}
```

```
Const MAXFILAS=3;  
      MAXCOLUMNAS=3;
```

```
Type TMat=array [1..MAXFILAS,1..MAXCOLUMNAS] of integer;
```

```
Procedure CargarMatriz (var matrizCar:TMat);  
{Carga la matriz con numeros enteros ingresados por el usuario}
```

```
var fila,columna,entero:integer;  
begin  
  for fila:=1 to MAXFILAS do  
    for columna:=1 to MAXCOLUMNAS do  
      begin  
        write('Ingrese un Numero ');  
        read(entero);  
        matrizCar[fila,columna]:=entero;  
      end;  
end;
```

```
Function Posmin (matrizPos:TMat;liminfPos, filaPos:integer):integer;
```

```
var posicionmenor,posicion:integer;  
begin  
  for posicion:=liminfPos to MAXCOLUMNAS do  
    if (matrizPos[filaPos,liminfPos])>=(matrizPos[filaPos,posicion]) then  
      begin
```

```
    matrizPos[filapos, liminfPos] := matrizPos[filapos, posicion];
    posicionmenor := posicion;
  end;
  Posmin := posicionmenor;
end;

Procedure IntercambiaElementos (var
matrizInt: TMat; PosminInt, liminfInt, filaInt: integer);
var ent: integer;
begin
  ent := matrizInt[filaInt, liminfInt];
  matrizInt[filaInt, liminfInt] := matrizInt[filaInt, PosminInt];
  matrizInt[filaInt, PosminInt] := ent;
end;

Procedure OrdenaPorSeleccion (var matrizOrd: TMat);
var Liminf, fila: integer;
begin
  for fila := 1 to MAXFILAS do
    for Liminf := 1 to (MAXCOLUMNAS-1) do

      IntercambiaElementos (matrizOrd, Posmin (matrizOrd, Liminf, fila), Liminf, fila);
    end;

Procedure MostrarMatriz (matrizMos: TMat);
  {Muestra la matriz por pantalla}
var fila, columna: integer;
begin
  for fila := 1 to MAXFILAS do
    begin
      writeln('');
      writeln('Fila: ', fila);
      for columna := 1 to MAXCOLUMNAS do
        begin
          write (matrizMos[fila, columna], ', ');
        end;
      end;
    end;
end;

var matrizenteros: TMat;
begin
  CargarMatriz (matrizenteros);
  OrdenaPorSeleccion (matrizenteros);
  MostrarMatriz (matrizenteros);
end.
```