Arrays

lecture

* Two- Dimensional Arrays:

the simplest Form of multidimensio
array is Two dimensional array. a two
dimensional array is a list of one dimens
arrays.

to declare a two-dimensional array of
size x, y you would write as follows

```
var
  array name: array [1..X, 1..Y] of elemen
                                    type
```

ex:

	col0	col1	col2	col3
row0				
row1	a11	a12	a13	
row2	a21	a22	a23	
	a31	a32	a33	

يكون العنصر عندنا بالشكل التالي.

a[0][0] a[0][1] a[0][2] a[0][3]

a[1][0] a[1][1] a[1][2] a[1][3]

a[2][0] a[2][1] a[2][2] a[2][3]

re!

if the number of rows ~~is~~ are equal to the number of columns the array is called ~~square array~~ (square array) then,

we have

1- main Diagonal : القطر الرئيسي.

set of element that include the First element in the first row to the last element in the last column. the condition ~~of the~~ to satisfy the main Diagonal is (I=J)

2- Secondary diagonal :- القطر الثانوي

include the last element of the first row to the first element in the last row.

the condition is ~~(I+J=J+1)~~

$j = n - i + 1$ / where n is no. of columns or rows in the array.

برنامج لطباعة عناصر القطر الرئيسي في مصفوفة ثنائية 3x3

```
Program array;
Var
  A: array[1..3, 1..3] of integer;
  i,j : integer;
begin
   For i:= 1 to 3 do
   For j:= 1 to 3 do
   begin
   read ( A[i,j]);
   end;
   For i:= 1 to 3 do
   begin
   For j:= 1 to 3 do
   if i=j then
   writeln (A[i,j] : 3);
   end;
   readln;
end.
```

3) upper triangle : المثلث العلوي
set of element that stay upper of the main Diagonal.
the condition to satisfy it is (J > I)

4) lower triangle : المثلث السفلي
Set of element that stay at the lower of the ~~secondary~~ main Diagonal.
to satisfy it (J < I).

H.W.

١- لنفس عناصر الصفوف السابقة
اطبع عناصر القطر الثانوي.

٢- اطبع عناصر الصف الثالث في مصفوفة ثنائية
البعد 5*5 .

ex1

```
var
  a: array[0..3, 0..3] of integer;
     i, j: integer;
begin
     for i := 0 to 3 do
     for j := 0 to 3 do

     a[i,j] := i * j;

...
end;
```

ملاحظة:

to read an array of two D..

لقراءة المصفوفة ذات البعدين نحتاج الى حلقتين الاولى للبعد الاول، والثانية للبعد الثاني. هذا اذا كنا نريد ادخال قيم الى المصفوفة عن طريق جمل الادخال Read, Readln.

```
or i:= 1 to 3 do
  begin
    For j:= 1 to 3 do
    X[I,J]:= Sum + (a[i,j]*(b[i,j]));
    End;
```

هنا يتم ضرب العمود الأولى في الثانية (a في b)
ويتم تخزين الناتج في العمود الثالث X.

```
For i:=1 to 3 do
  begin
    for j:= 1 to 3 do
    writeln (X[1,j]);
  end.
```

هنا يتم طباعة العمود الثالث X.

* برنامج لضرب مصفوفتين مربعتين.

Program example;

```
var
i,j,sum: integer;
X, a, b: array[1..3, 1..3] of integer;
Begin
  For i:=1 to 3 do
  Begin
    For j:=1 to 3 do
    Read ( a[i,j]);
  end;
  For i:=1 to 3 do
  Begin
    for j:=1 to 3 do
    Read (b[i,j]);
  End;
  sum:=0;
  For i:=1 to 3 do
  Begin
    For j:=1 to 3 do
    X[i,j]:=sum+((a[i,j])*(b[i,j]
  end;
  For i:=1 to 3 do
  begin
    For j:=1 to 3 do
    writeln ( X[i,j]:2);
  End;
End.
```

ترك مسافة بين X (رقمين) عند الطباعة
2 space

ملاحظة: المصفوفة المربعة هي المصفوفة ذات البعدين.

شرح البرنامج السابق.

First block

```
For i:= 1 to 3 do
 Begin
   For j:=1 to 3 do
    Read (a[i,j]);
 End;
```

هنا سيتم قراءة المصفوفة (a) كاملة (البعد الأول والثاني)

Second block

```
For i:= 1 to 3 do
 Begin
   For j:= 1 to 3 do
   Read (b[i,j]);
 End;
```

هنا سيتم قراءة المصفوفة b كاملة (البعد الأول والثاني)

```
sum:=0;
```

عبارة عن مخزن سيتم فيه جمع قيمة ضرب المصفوفة الأولى بالثانية وسيتم تخزينها في مصفوفة ثالثة X.