جامعة حماة
كلية العلوم في مصياف / السنة الأولى

المادة: لغات البرمجة

المحاضرة الرابعة: المتجهات والمصفوفات في لغة باسكال

Array Structures in Pascal

العام الدراسي ٢٠١٩- ٢٠٢٠

استخدام الأحرف في كتابة برنامج باسكال

- مترجم لغة باسكال Pascal Compiler (بعكس اللغات الأخرى) يغضّ الطرف عن حالة الأحرف (عالية أو منخفضة). لذلك فإن التعريفات التالية Myname ، MyName، myname، myName و MYNAME كلها متساوية. و بشكل عام يعدّ هذا أمرا ايجابيا، ففي اللغات الحسّاسة لحالة الأحرف قد تحدث العديد من الأخطاء اللغوية بسبب الإهمال في مراعاة حالة الأحرف.
- **ملاحظة**: ربما الحالة الاستثنائية الوحيدة لقاعدة حساسية الأحرف في باسكال هي :إجراء Register في حزمة المكونات، لابد لها أن تبدأ بحرف R العالي، وذلك للمحافظة على التوافقية مع.C++ Builder
- إلا أنه توجد بعض السلبيات.

✓ **أولا**: يجب أن تكون هذه التعريفات متساوية بالفعل، لذا يجب تجنب استعمالها كعناصر مختلفة.

✓ **ثانيا**: يجب أن نكون حذرين قدر الإمكان عند استخدامنا للأحرف العالية (لتحسين قراءة البرنامج).

- توحيد استخدام حالة الأحرف ليس ملزما من قبل المترجم ولكنها عادة حسنة يحبّذ إتباعها.
- الأسلوب المتبّع هو تكبير الحرف الأول فقط من كل معرّف identifier وعندما يكون المعرّف مركّبا من عدة كلمات (لا يمكننا حشر فراغ في المعرّف)، فإن كل أول حرف من كل كلمة يجب أن يكون عاليا:

MyLongIdentifier
MyVeryLongAndAlmostStupidIdentifier

استخدام الأحرف في كتابة برنامج باسكال

- هناك حالات أخرى لا يأبه لها المترجم كالفراغات والأسطر الفارغة والمسافات (tabs) التي نقوم بوضعها في البرنامج. كل هذه الحالات مجتمعة تسمّى بالفراغ الأبيض white space الفراغات البيضاء تستخدم فقط لتحسين قراءة البرنامج؛ و لا تؤثر على عملية الترجمة .
- لغة باسكال تسمح لنا بكتابة تعليمة واحدة موزعة على عدة أسطر، فالتعليمة الطويلة يمكن تجزئتها لتكون في سطرين أو أكثر. السلبية الوحيدة لإمكانية أن تكون التعليمات في أكثر من سطر هي أنه عليك أن تتذكّر بأن تضع فاصلة منقوطة آخر كل تعليمة، أو بدقة أكثر، أن تفصل بين التعليمة والتي تليها. لاحظ أن القيد الوحيد هنا أن الجملة النصية الواحدة لا يمكن مدّها لعدّة أسطر.
- لا توجد قواعد ثابتة لاستخدام الفراغات و التعليمات متعددة الأسطر، إنما توجد بعض الأعراف التي تجعل برنامجك يبدو أفضل عندما تقوم بطباعته على الورق. غير هذا فإن الأسطر الطويلة قد يتم تجزئتها من أي موضع حتى من منتصف الكلمة عند طباعة البرنامج .
- عندما يكون للوظيفة أو الإجراء عدة محددات Parameters ، فإن العادة المتّبعة هي وضع هذه المحدّدات في سطر آخر.
- نستطيع ترك سطر بكامله أبيضا (فارغا) قبل وضع أي تعليق أو ملاحظة، أو لتقسيم جزء كبير من التعليمات إلى أجزاء أصغر. هذه الفكرة البسيطة بإمكانها تحسين قراءة البرنامج، سواء على الشاشة أو عند طباعتها.
- استخدم الفراغات لفصل محددات استدعاء وظيفة function call ، وربما حتى فراغ قبل فتح الأقواس، أيضا حافظ على فراغات لفصل رموز العمليات في التعابير البرمجية.

تعليمات لغة باسكال والكلمات المفتاحية و الكلمات المحجوزة

- تمكننا لغة باسكال من تحديد أسماء بعض المعرّفات (المتغيرات والثوابت) التي سنستخدمها لاحقاً في جسم البرنامج ضمن التعليمات أو المعادلات لأنها تصبح جزء منها، تقدم باسكال مجموعة من التعليمات والتعبيرات.
- **الكلمات المفتاحية:** هي كل المعرّفات المحجوزة من قبل (اوبجكت) باسكال، والتي لها معاني في لغة باسكال. دليل لغة دلفي (Help) يميّز بين الكلمات المحجوزة والتوجيهات كالتالي:
- الكلمات المحجوزة لا يمكن استخدامها كمعرّفات، بينما التوجيهات لا يجب استخدامها لنفس الغرض، حتى لو قبلها المترجم Compiler. عند كتابة البرنامج عليك تجنّب استخدام أية كلمة محجوزة كمعرّف.

في الجداول التالية يمكننا رؤية قائمة كاملة بالمعرّفات التي لها دورا خاصا في لغة اوبجكت باسكال (في دلفي ٤)، بما في ذلك الكلمات والكلمات المحجوزة الأخرى.

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

الكلمة الرئيسة	الدور
absolute	directive (variables) موجّه (متغير)
abstract	directive (method) موجّه (مسار)
and	operator (boolean) محامل (بولي)
array	type نوع
as	operator (RTTI) محامل
asm	statement تعليمة
assembler	backward compatibility (asm) توافقية مع السابق
at	statement (exceptions) تعليمة (استثناءات)
automated	access specifier (class) محدد دخول (طبقة)
begin	block marker علامة حيّز
case	statement تعليمة
cdecl	function calling convention قاعدة استدعاء وظيفة
class	type نوع
const	declaration or directive (parameters) تعريف أو توجيه (محددات)
constructor	special method مسار خاص
contains	operator (set) عامل (فئة)
default	directive (property) توجيه (سمة)
destructor	special method مسار خاص
dispid	dispinterface specifier محدد واجهة اطلاق
dispinterface	type نوع
div	operator عامل
do	statement تعليمة
downto	statement (for) (for) تعليمة
dynamic	directive (method) توجيه (مسار)
else	statement (if or case) (case او if) تعليمة
end	block marker تعليم حيّز

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

except	statement (exceptions) تعليمة (استثناءات)
export	backward compatibility (class) توافقية مع السابق
exports	declaration تعريف
external	directive (functions) توجيه
far	backward compatibility (class) توافقية مع السابق
file	type نوع
finalization	unit structure بنية وحدة
finally	statement (exceptions) تعليمة
for	statement تعليمة
forward	function directive توجيه وظيفة
function	declaration تعريف
goto	statement تعليمة
if	statement تعليمة
implementation	unit structure بنية وحدة
implements	directive (property) توجيه (مسار)
in	operator (set) - project strucure عامل (فئة) - بنية مشروع
index	directive (dipinterface) توجيه
inherited	statement تعليمة
initialization	unit structure بنية وحدة
inline	backward compatibility (see asm) توافقية مع السابقة
interface	type نوع
is	operator (RTTI) عامل
label	declaration تعريف
library	program structure بنية برنامج
message	directive (method) توجيه (مسار)
mod	operator (math) عامل (رياضي)
name	directive (function) توجيه (وظيفة)
near	backward compatibility (class) توافقية مع السابق

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

nil	value قيمة
nodefault	directive (property) توجيه (مسار)
not	operator (boolean) عامل (بولي)
object	backward compatibility (class) توافقية مع السابق
of	statement (case) تعليمة
on	statement (exceptions) تعليمة
or	operator (boolean) معامل (بولي)
out	directive (parameters) توجيه (محدّدات)
overload	function directive توجيه وظيفة
override	function directive توجيه وظيفة
package	program structure (package) بنية برنامج (حزمة)
packed	directive (record)توجيه (تسجيلة)
pascal	function calling convention طريقة استدعاء وظيفة
private	access specifier (class) محيّن لوصول (طبقة)
procedure	declaration تعريف
program	program structure بنية برنامج
property	declaration تعريف
protected	access specifier (class) محيّن لوصول (طبقة)
public	access specifier (class) محيّن لوصول (طبقة)
published	access specifier (class) محيّن لوصول (طبقة)
raise	statement (exceptions) تعليمة (اعتراضات)
read	property specifier محيّن سمة
readonly	dispatch interface specifier محيّن واجهة ارسال
record	type نوع
register	function calling convention طريقة لاستدعاء وظيفة
reintroduce	function directive توجيه وظيفة
repeat	statement تعليمة

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

requires	program structure (package) بنية برنامج (حزمة)
resident	directive (functions) توجيه (وظيفة)
resourcestring	type نوع
safecall	function calling convention طريقة لاستدعاء وظيفة
set	type نوع
shl	operator (math) عامل (رياضة)
shr	operator (math) عامل (رياضة)
stdcall	function calling convention طريقة لاستدعاء وظيفة
stored	directive (property) توجيه (سمة)
string	type نوع
then	statement (if)
threadvar	declaration تعريف
to	statement (for) تعليمة
try	statement (exceptions) تعليمة (استثناءات)
type	declaration تعريف
unit	unit structure بنية وحدة
until	statement تعليمة
uses	unit structure بنية وحدة
var	declaration تعريف
virtual	directive (method) توجيه (مسار)
while	statement تعليمة
with	statement تعليمة
write	property specifier محيّن سمة
writeonly	dispatch interface specifier محيّن واجهة ارسال
xor	operator (boolean) عامل (بولي)

بنية المتجهة Array Structure

المتجهة: هي بنية معطيات نظامية متجانسة تكون جميع عناصرها من نوع قاعدي واحد بسيط قياسي أو بدائي (Integer, Real, Char, Boolean) وتتميز بإمكانية الوصول المباشر إلى أي عنصر داخلها دون المرور بالعناصر الأخرى و ذلك بمساعدة (i) دليل المتجهة (الفهرس Index) الذي **يحدد موقع العنصر** الذي نرغب بتنفيذ العمليات عليه ضمن المتجهة [i] x، و يتم التعريف عنها بواسطة التعليمة **Type** على الشكل التالي في بداية البرنامج:

Type T = **array** [i] of **To**;

بناءً على ما سبق نرى أن تعريف المتجهة (T) يتم بدلالة نوعين مصممين هما:

- نوع القاعدة (To) الذي يحدد نوع عناصر المتجهة (Integer, Real, Char, Boolean).
- النوع المعرّف لدليل المتجهة [i].

بنية المتجهة Array Structure

١. **الخطوة الأولى:** تعريف المتجهة (تحديد اسمها و حجمها و نوع البيانات بداخلها)

```
Type Row = array [ 1 .. 80 ] of integer ;
Type c = [ 80 .. 100 ] of real ;
Type Alfa = array [ 'a' .. 'z' ] of char;
```

١- تعريف المتجهة

٢. **الخطوة الثانية:** يجب حجز متحولات من النوع المعرّف في قسم التصريح Var و كذلك يتم التعريف عن الدليل Index الذي يعتبر المفتاح الأساسي للوصول المباشر إلى أي عنصر من عناصر المتجهة، وتعتبر المتجهات من أكثر بنى المعطيات استخداماً في البرمجة.

```
Var   x : Row ;
      y : c ;
      z : Alfa ;
      i : integer ;
```

٢- تعريف المتغيرات

مثال: أوجد مجموع عناصر المتجهة التالية SUM

```
Program SUM;
Type MAT = array [1..n] of integer;
Var
   M: MAT;
   i, n, Sum: integer;
Begin
   write ('Input n =');
   readln (n);
    Sum:= 0;
   for i:=1 to n do
   begin
        write('Input M [', i , '] =');
        readln(M[ i ]);
        Sum:= Sum +M [ i ];
   end;
   write ('Sum= ', Sum);
   readln();
End.
```

تعريف المتجهة

تعريف المتغيرات

إدخال عدد عناصر المتجهة

إدخال عناصر المتجهة

طباعة مجموع عناصر المتجهة

مثال: أوجد (Sum مجموع) و (AVG متوسط عناصر المتجهة التالية)

```
Program AVG ;                                   تعريف المتجهة
Type MAT = array [1..n] of integer;
Var
    M: MAT;                                     تعريف المتغيرات
    i, n, Sum : integer;
    Avg: real;
Begin
    Sum:= 0;
    write('Input n ='); readln (n);             إدخال عدد عناصر المتجهة
    for i:=1 to n do
    begin                                       إدخال عناصر المتجهة
        write('Input M [', i , '] =');
        readln(M[ i ]);
        Sum:= Sum + M [ i ];
    end;
    Avg := Sum / n;
    writeln('Sum= ', Sum);                      طباعة مجموع
    writeln('Avg= ', Avg);                      عناصر المتجهة
                                                ومتوسطها
    readln();
End.
```

مثال: أوجد العنصر الأصغر MIN في المتجهة التالية

```
Program MIN;
Type MAT = array [1..n] of integer;
Var
   M: MAT;
   i, n, min : integer;
Begin
   write('Input n =');
   readln(n);
   for i:=1 to n do
   begin
        write('Input M [', i , '] =');
        readln(M[ i ]);
   end;
   min:=M[ 1 ];
   for i :=2 to n do
    if (M [ i ] < min) then
    min := M [ i ];
   write('Min = ', min);
   readln();
End.
```

تعريف المتجهة

تعريف المتغيرات

إدخال عدد عناصر المتجهة

إدخال عناصر المتجهة

إيجاد العنصر الأصغر

طباعة العنصر الأصغر

مثال: أطبع عناصر المتجهة التالية والعنصر الأكبر MAX

```
Program MAX;
Type MAT = array [1..n] of integer;
Var
    M: MAT;
    i, n, max : integer;
Begin
    write ('Input n =');
    readln (n);
    for i:=1 to n do
    begin
        write('Input M [', i , '] =');
        readln(M[ i ]);
    end;
    max :=M[ 1 ];
    for i :=2 to n do
    if (M [ i ] > max) then
    max := M [ i ];
    for i :=1 to n do
    write(M [ i ]) ;
    writeln();
    write('Max = ', max );
    readln();
End.
```

تعريف المتجهة

تعريف المتغيرات

إدخال عناصر المتجهة

إيجاد العنصر الأكبر

طباعة عناصر المتجهة

طباعة العنصر الأكبر

وظيفة Homework

١. اكتب برنامجاً يقوم بإدخال درجات الحرارة في مدينة حمص لمدة أسبوع، ثم يظهر متوسط درجات الحرارة لهذا الأسبوع و درجة الحرارة العظمى و ذلك بمساعدة المتجهات ؟

٢. اكتب برنامجاً يقوم بإدخال عشرة أرقام صحيحة عشوائياً، ثم يرتبها تصاعدياً بمساعدة المتجهات ؟

٣. اكتب برنامجاً يقوم بإدخال أسماء عشرة طلاب، ثم يرتبها أبجدياً بشكل تصاعدي بمساعدة المتجهات ؟

متجهة المتجهات و المصفوفة
Array of Array and Matrix Structure

- **المصفوفة**: هي بنية معطيات نظامية متجانسة تكون جميع عناصرها من نوع قاعدي واحد **مرّكب** أي يمكن أن يكون نوع بنيوي معرّف عليه في بداية البرنامج أو له نوع متجهي " متجهة متجهات " وتتميز بإمكانية الوصول المباشر إلى أي عنصر فيها دون المرور بالعناصر الأخرى و ذلك بمساعدة مجموعة الأدلة (الفهارس i , j) للمصفوفات Index's التي تحدد موقع العنصر الذي نرغب بتنفيذ العمليات عليه ضمن المصفوفة.. x [i] [j] [k]، أو ... x [i , j] ، و يتم التعريف عنها ضمن قسم تعريف المتغيرات Var على الشكل التالي في بداية البرنامج: Var

Matrix : array [i] of To;

تعريف المصفوفة ذات البعد واحد

Mat : array [i , j] of To;

تعريف المصفوفة ذات البعدين – طريقة (١)

أو

Mat : array [i] of array [j] of To;

تعريف المصفوفة ذات البعدين – طريقة (٢)

مثال: برنامج يقرأ ١٠ أحرف ، ثم يقوم بطباعتها بالعكس

- التعريف باسم البرنامج
- التعريف بالمتغيرات
- بداية البرنامج
- إدخال عناصر المصفوفة
- طباعة عناصر المصفوفة عكسياً
- نهاية البرنامج

```
Program ex4 (input,output);
Var
    i: integer;
    x: array [1..10] of char;
Begin
    for i:= 1 to 10 do
    read(x[i]);

    For i:=10 down to 1 do
    write(x[i]);
End.
```

مثال: أوجد العنصر الأصغر MIN في المصفوفة التالية

```
Program MIN;
VAR
   M: array [1..n] of integer;
   i, n, min : integer;
```

تعريف المصفوفة والمتغيرات

```
Begin
   write (‘Input n =');
   readln (n);
```

إدخال عدد عناصر المصفوفة

```
   for i:=1 to n do
   begin
         write(‘Input M [‘, i , ‘] =‘);
         readln(M[ i ]);
   end;
```

إدخال عناصر المصفوفة

```
   min:=M[ 1 ];
   for i :=2 to n do
    if (M [ i ] < min) then
    min := M [ i ];
```

إيجاد العنصر الأصغر

```
   writeln(‘MIN = ‘, min);
   readln();
End.
```

طباعة العنصر الأصغر

مثال: أوجد العنصر الأكبر MAX في المصفوفة التالية و أطبع عناصرها

```
Program MAX;
Var
    M: array [1..n] of integer;
    i, n, max : integer;
```

تعريف المصفوفة والمتغيرات

```
Begin
    write (‘Input n =');
    readln (n);
```

إدخال عدد عناصر المصفوفة

```
    for i:=1 to n do
    begin
        write(‘Input M [‘, i , ‘] =‘);
        readln(M[ i ]);
    end;
```

إدخال عناصر المصفوفة

```
    max :=M[ 1 ];
    for i :=2 to n do
     if (M [ i ] > max) then
     max := M [ i ];
```

إيجاد العنصر الأكبر

```
    for i :=1 to n do
     write(M [ i ]) ;
    writeln();
```

طباعة عناصر المصفوفة

```
    writeln(‘Max = ‘, max );
    readln();
```

طباعة العنصر الأكبر

```
End.
```

مثال: رتّب المصفوفة التالية تصاعدياً

```
Program Sort 1;
Var
   M: array [1..n] of integer;
     i, j, n, min ,temp: integer;

  Begin
  write (‘Input n =');
  readln (n);
  for i:=1 to n do
  begin
        write(‘Input M [‘, i , ‘] =‘);
        readln(M[ i ]);
  end;
  for j:=1 to n do
  begin
     min:=M[ j ];
     for i:= j+1 to n do
       if (M [ i ] < min) then
       begin
            temp := M [ i ];
            M[ i ]:= min;
            min := temp;
       end;
     M[ j ] := min;
   end;
  for j:=1 to n do
  write(M [ j ]);
End.
```

تعريف المصفوفة والمتغيرات

إدخال عدد عناصر المصفوفة

إدخال عناصر المصفوفة

إيجاد العنصر الأصغر

ترتيب عناصر المصفوفة تصاعديا

طباعة عناصر المصفوفة

مثال: رتّب المصفوفة التالية تنازلياً

```
Program Sort 2;
Var
   M: array [1..n] of integer;
     i, j, n, max ,temp: integer;
Begin
   write('Input n =');
   readln(n);
   for i:=1 to n do
   begin
         write('Input M [', i , '] =');
         readln(M[ i ]);
   end;
   for j:=1 to n do
   begin
      max :=M[ j ];
      for i:= j+1 to n do
            if (M [ i ] > max) then
            begin
            temp := M [ i ];
             M [ i ]:= max ;
             max := temp;
            end;
   end;
   for j:=1 to n do
   write(M [ j ]);
End.
```

تعريف المصفوفة والمتغيرات

إدخال عدد عناصر المصفوفة

إدخال عناصر المصفوفة

إيجاد العنصر الأكبر

ترتيب عناصر المصفوفة تنازلياً

طباعة عناصر المصفوفة

المصفوفة ذات البعدين
2 X Array (Matrix)

- **المصفوفة ذات البعدين**: هي بنية معطيات نظامية متجانسة لها بعدان (n,m) وتكون جميع عناصرها من نوع قاعدي واحد بسيط أو مرّكب، أي يمكن أن يكون من نوع بنيوي معرّف عليه في بداية البرنامج أو له نوع متجهي " متجهة متجهات " وتتميز بإمكانية الوصول المباشر إلى أي عنصر فيها دون المرور بالعناصر الأخرى وذلك بمساعدة مجموعة الأدلة (i , j) Index's التي تحدد موقع العنصر الذي نرغب بتنفيذ العمليات عليه ضمن المصفوفة [i , j] x، ويمكن التعريف عنها بعد تعليمة Varعلى الشكل التالي في بداية البرنامج:

```
Var
Matrix: array [ i , j ] of To;
```

تعريف المصفوفة ذات البعدين

- نستخدم في المصفوفات ذات البعدين حلقتين متداخلتين ليتم قراءة البعد الأول i والبعد الثاني j .

مثال: برنامج لضرب مصفوفتين A و B
لهما نفس البعدين n و m و طباعة
عناصر المصفوفة الجديدة X

```
Program example (input,output);
Var                                   تعريف المصفوفات
    i, j , n, m : integer ;           ذات البعدين
    x,a,b:array [ 1..n , 1..m ] of integer ;
Begin
    readln( n ) ;   readln( m ) ;     إدخال أبعاد المصفوفات
    for i:= 1 to n do                 إدخال عناصر المصفوفة A
     begin
    for j:=1 to m do
          readln( a[ i , j ] ) ;
    end ;
    for i:= 1 to n do                 إدخال عناصر المصفوفة B
    begin
       for j:=1 to m do
            readln( b[ i, j ] ) ;
    end ;
```

```
                                      ضرب عناصر المصفوفتين A و B و إسنادهم إلى المصفوفة X
for i:=1 to n do
    for j:= 1 to m do
x [ i, j ] := ( a[ i, j ] ) * ( b[ i, j ] ) ;
```

```
for i:= 1 to n do
begin
        for j:= 1 to m do
        writeln( x [ i, j ] ) ;       طباعة عناصر المصفوفة X
writeln();
end ;
End.
```

مثال: برنامج لطباعة مصفوفة عناصرها ناتج مجموع العنصرين المتقابلين في مصفوفتين مربعتين

```
Program example (input,output);
Var
    i,j :integer;
    x,a,b:array [1..5,1..5] of integer;
Begin
    for i:= 1 to 5 do
    begin
        for j:=1 to 5 do
            readln( a[ i, j ] );
    end;
    for i:= 1 to 5 do
    begin
            for j:=1 to 5 do
             readln( b[ i, j ] ) ;
    end;
    for i:=1 to 5 do
    begin
        for j:= 1 to 5 do
    x[ i, j ] := ( a[ i, j ] ) + ( b[ i, j ] ) ;
    end;
    for i:= 1 to 5 do
    begin
            for j:= 1 to 5 do
            write( x [ i, j ] ) ;
    writeln();
    end;
End.
```

تعريف المصفوفات ذات البعدين

إدخال عناصر المصفوفة A

إدخال عناصر المصفوفة B

جمع عناصر المصفوفتين و إسنادهم إلى A و B المصفوفة X

طباعة عناصر المصفوفة X

متجهة المتجهات و المصفوفة
Array of Array and Matrix Structure

٢- لدينا مكتبة تحوي مجموعة أقسام، وكل قسم يحتوي مجموعة كتب، وكل كتاب مجموعة صفحات، وكل صفحة مجموعة سطور، وكل سطر مجموعة محارف.

```
Type Line = array [ 1 .. 80 ] of char ;
     page = array [ 1 .. 25 ] of Line ;
     Book = array [ 1 .. 500 ] of page ;
     Section = array [ 1 .. 1000 ] of Book ;
     Library = array [ 1 .. 10 ] of Section ;
Var
x : array [1..n] of Library ;
i, j, k, l, m : integer ;
```

- حيث تم حجز مكان في ذاكرة الحاسب للمتحول x من النوع المعرّف في قسم التصريح Library و كذلك يجب التعريف عن الأدلة Index's.

وظيفة Homework

١- حدّد مكان العنصر الأكبر (إحداثياته) في مصفوفة أعداد صحيحة ذات بعدين (n x m) ؟

٢- أوجد متوسط الهطولات المطرية في خمسة مدن سورية تم أخذ عيناتها على مدار شهر في فصل الشتاء مستخدماً المصفوفات ذات البعدين ؟

٣- أوجد معدّل درجات كل طالب، إذا علمت أن عدد الخريجين ١٥ طالب و عدد المواد ٣٠ مادة، ثم اكتب اسم و علامة صاحب أعلى معدّل بين الخريجين مستخدماً المصفوفات ذات البعدين ؟