Loops and duplicates

الحلقات والتكرارات في لغة باسكال

التكرار يستخدم متى ما اردنا اعادة تطبيق ايعاز معين اكثر من مره

مثال على ذلك اذا اردنا طباعة جملة hello على شاشه التنفيذ 5 مرات سوف نحتاج الى كتابة الايعاز writeln خمسة مرات , لكن باستخدام ايعاز التكرار يمكن كتابة الامر مره واحده فقط و سوف يتم طباعة الجمله بالعدد المحدد

هنالك 3 انواع من ايعازات التكرار و هي
For , While , Repeat

ايعاز For

في البدايه و قبل كتابة ايعاز ال for يجب تعرف المتغير حسب التكرار المطلوب مثلا اذا كان ارقام صحيحه يجب ان يعرف integer اذا كان احرف يجب ان يعرف char و هكذا

تكتب جملة ال for بهذا الشكل اثناء كتابة البرنامج

For ….. To …. Do

Example :- write a program in pascal to do the summation of the even numbers between 10 to 30 ?

اكتب برنامج لايجاد مجموع الارقام الزوجيه المحدده بين الرقمين 10 الى 30

```
program forsumation;
var
i,sum:integer;
begin
sum:=0;
for i :=10 to 30 do
if ( i mod 2=0) then
sum:= sum+i;
writeln('sum=',sum);
readln
end.
```

شاشة التنفيذ

هنا البرنامج سيقوم بجمع الاعداد الزوجيه فقط اي

10+12+14+16 +..... + 28+30

ممكن تغيير خطوه واحده و جعل البرنامج يجمع الاعداد الفرديه فقط

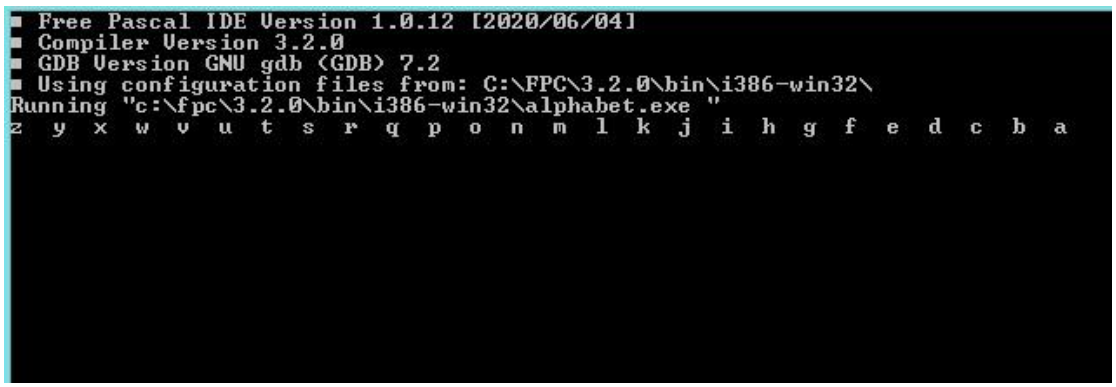
If (I mod 2<>0) then

Example :- write a program in pascal for printing the alphabet in descending order

اكتب برنامج لطباعة الحروف الابجديه بشكل تنازلي

```
program alphabet;
var
i:char;
begin
for i:='z' dwonto 'a' do
write (i,'   ');
readln
end.
```

شاشة التنفيذ



** للطباعه تم وضع الايعاز write حيث سيتم الطباعه في نفس السطر و ايضا تم وضع بين علامة الاقتباس (فراغ) هذا لكي يظهر فراغ بين حرف و الاخر و لا تظهر الاحرف متصله

** في حال وضع الايعاز writeln لا يتم وضع (فراغ) بين علامتي الاقتباس كون كل حرف سيتم طباعته في سطر لوحده

ملاحظة :- في حال رغبنا بإظهار الاحرف بشكل تصاعدي نقوم بتبديل ايعاز downto الى ايعاز to و كتابة الايعاز بهذا الشكل

```
for i:='a' to 'z' do
```

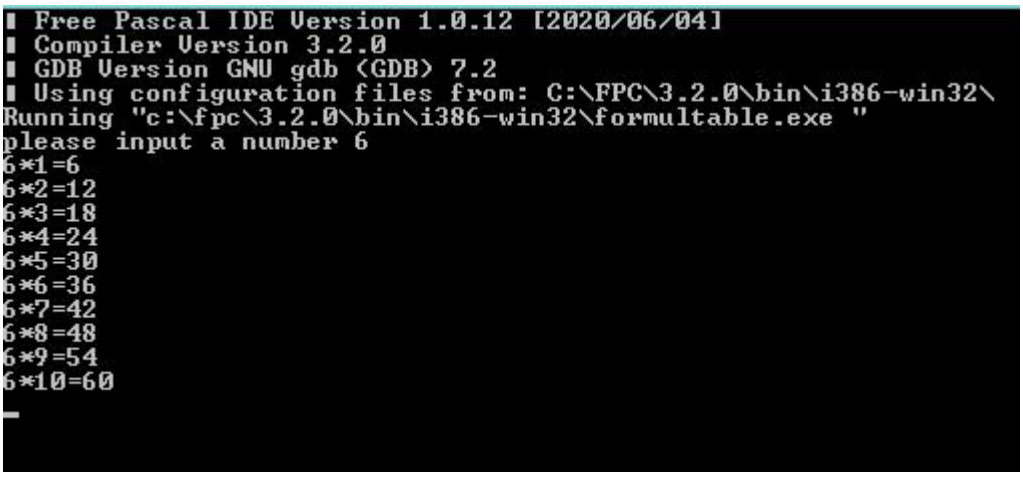
H.W 1

write a program to find the multiplication of a number that to user enter using for loop

اكتب برنامج لايجاد جدول الضرب لرقم يقوم بادخاله المستخدم

```
program formultable;
var
x,i: integer;
begin
write ('please input a number' );
readln(x);
for i:=1 to 10 do
writeln (x , ' * ' , i , ' = ' , x * i );
readln
end.
```

شاشة التنفيذ



مضروب الرقم 6

H.W 2

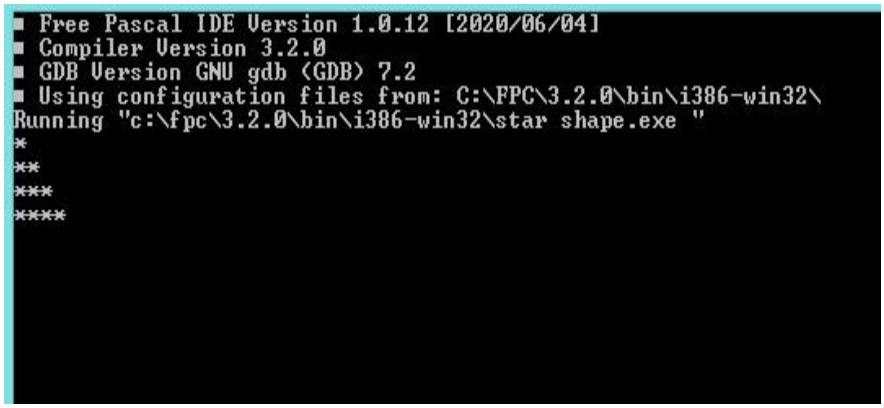
Write a program in Pascal to do the shape below using For Loop

```
*
**
***
****
```

اكتب برنامج لرسم الشكل ادناه باستخدام التكرار For

```
program starshape;
var
i,j:integer;
begin
for i:=1 to 4 do
begin
for j:=1 to i do
write('*');
writeln;
end;
readln
end.
```

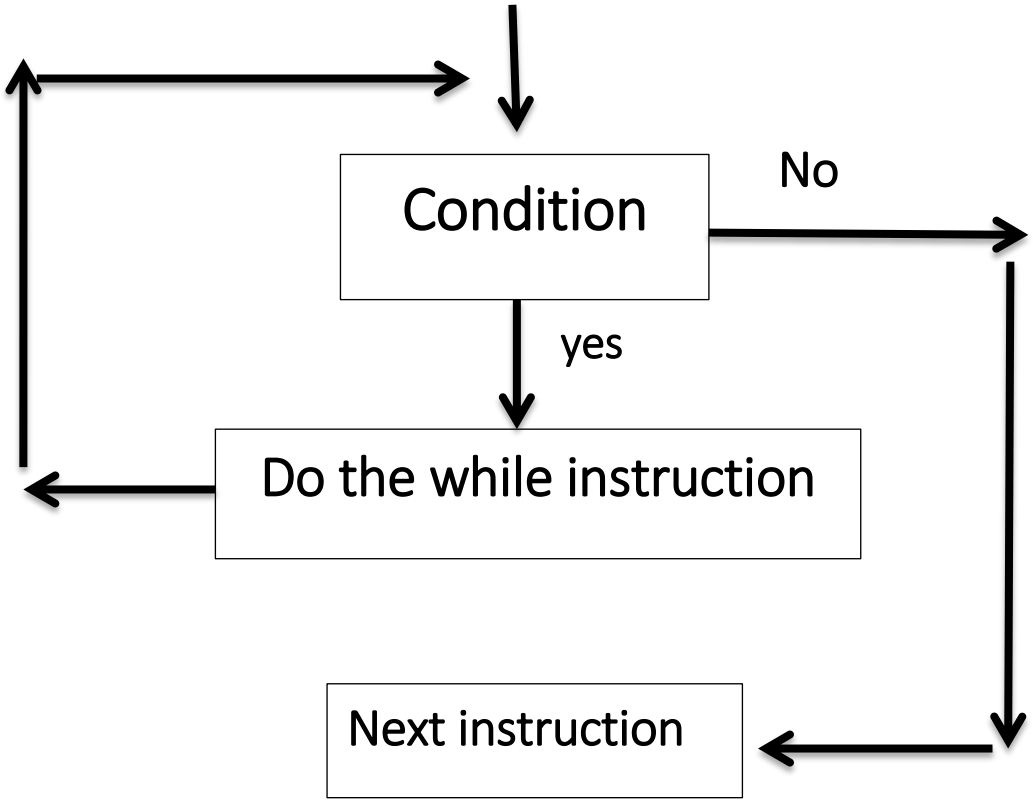
شاشة التنفيذ



ايعاز While

في البدايه يتم التحقق من الشرط هل هو متحقق ام لا

ان كان الشرط يتحقق فان التكرار سوف ياخذ القيمة الاولى و ينفذ الاوامر الموجوده داخل الايعاز حتى يصل الى end; و يتحقق هم تم الشرط ام لا اذا كان الجواب نعم فيعود لتكرار تنفيذ الايعاز مره ثانيه , الى ان يصل الى عدم تحقق الشرط فيخرج من دائرة التكرار و ينفذ الاوامر التي تليه



تكتب جملة while بهذا الشكل

```
While condition do
begin
:
:
End;
```

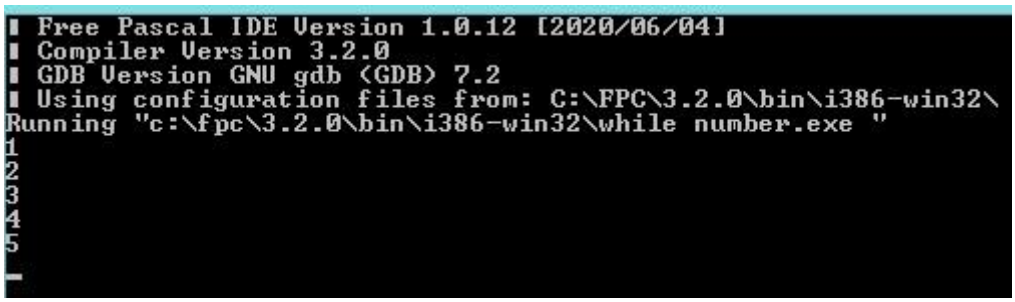
**ملاحظة تحوي جملة while على begin و end;

Ex : write a program that print the numbers from 1 to 5 using while loop

اكتب برنامج لطباعة الارقام من 1 الى 5 باستخدام التكرار while

```
Program number;
Var
i, :integer;
Begin
i:= 0;
While (i<5) do
Begin
i:=i+1;
Writeln(i);
End;
Readln
End.
```

شاشة التنفيذ



****تم طباعه الارقام من 1 الى الرقم 5 كل رقم في سطر**

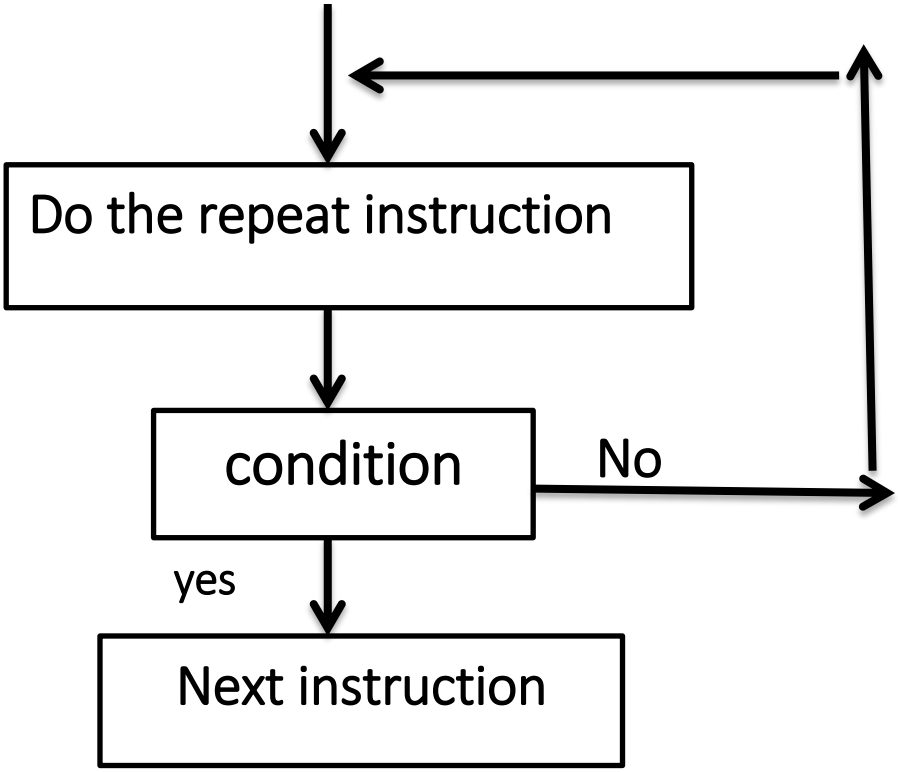
**يجب اعطاء قيمة للعداد قبل ايعاز ال while و في حال عدم اعطاء قيمة سيقوم البرنامج مباشرة باعطاء قيمة افتراضيه و هي 0

** في حال عدم كتابة الجمله ; i:=i+1 فان الناتج سيكون عباره عن قيمة المتغير i و سيبقى يكرر القيمة الى مالانهايه (جرب بنفسك)

ايعاز Repeat

يقوم البرنامج بالدخول الى حلقة التكرار repeat و ينفذ جميع الاوامر الموجوده بداخله لغاية الوصول الى جملة Until فيتوقف لسأل هل تحقق الشرط ام لا في حال تحقق الشرط فيخرج من الجمله التكراريه

في حال عدم تحقق الشرط فيعود الى جملة Repeat



تكتب جملة ال Repeat بهذا الشكل

```
Repeat
:
:
Until condition
```

**** ملاحظة**

جملة ال Repeat لا تحتاج الى

Begin و end

EX:- write a program to find the square of the numbers from 1 to 10 using Repeat loop

اكتب برنامج لطباعة مربع العدد من 1 الى 10 باستخدام تكرار Repeat

```
Program repeatsquare;
Var
i:integer;
begin
i:=1;
Repeat
Writeln (i, 'square=' , i*i);
i:=i+1;
Until i=11;
Readln
End.
```

شاشة التنفيذ

** سيقوم البرنامج بالتكرار من اول رقم تم اعطاءه في العداد i وهو العدد 1 لغايه العدد 10 رغم العدد 11 المُعطى في الشرط عند الايعاز Untel عندما يتحقق هذا الشرط i=11 سوف يخرج البرنامج من دائرة التكرار .

** في حال لدينا اكثر من شرط في السؤال يتم وضع الشرط بين الاقواس و يفصل ما بين الشرط اما OR او AND

Or : يجب ان تكون حاله واحده او الحالتين صحيحه

And: ان تكون الحالتين صحيحه

Ex: write A program to print a certain digit of numbers and the program stops if 10 numbers are entered or if the number 0 is entered using **Repeat** loop

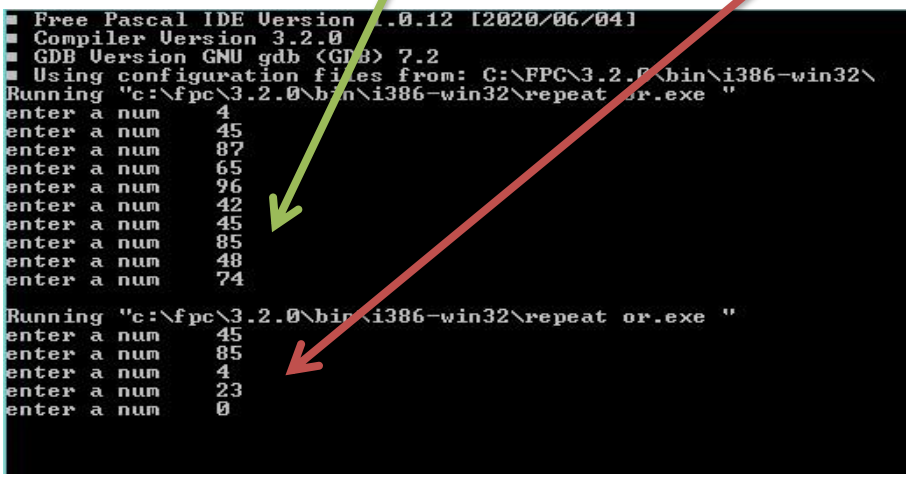
برنامج لطباعة عدد من الارقام ويتوقف البرنامج في حال تم ادخال 10 اعداد او في حال تم ادخال العدد 0 باستخدام التكرار Repeat

```
program Loops;
var
i,s: Integer;
begin
i:=0;
repeat
i:=i+1;
write('enter a num');
readln(s);
until (i=10) or (s=0);
readln
end.
```

يستمر بالتنفيذ و يخرج من الشاشه عند ادخال 10 اعداد

يستمر بالتنفيذ و يخرج من الشاشه عند ادخال العدد 0

شاشة التنفيذ

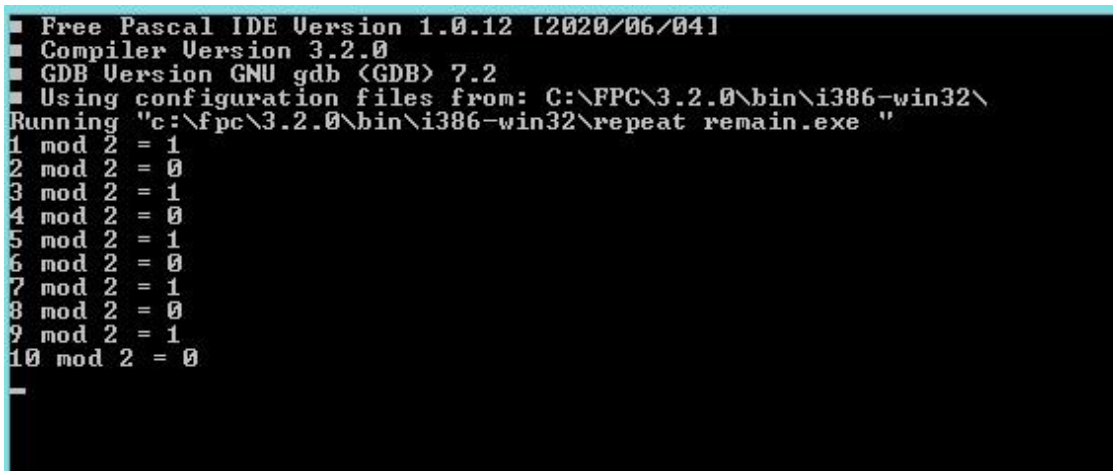


ex :- write a program to find the remainder of the division of numbers from 1 to 10 using Repeat loop

اكتب برنامج لايجاد باقي القسمة للاعداد المحصوره بين 1 و 10 باستخدام تكرار repeat

```
Program remain ;
Var
i:integer;
begin
i:=1;
Repeat
Writeln (i, ' mod 2 = ' , i mod 2);
i:=i+1;
Until i=11;
readln
end.
```

شاشة التنفيذ



ARRARY

المصفوفات

المصفوفه : عباره عن متغير له اكثر من قيمة واحده و تكون على شكل اعمده و صفوف و كل خليه تحوي على الرقم الخاص بها للتفريق بين القيم
الفائدة من المصوفوفات هي بدل ان نقوم بتعريف عدد من المتغيرات نقوم بتعرف متغير واحد فقط و هذا المتغير سيكون له عدد من القيم , اتت فكرة الجداول و المصفوفات لتسهيل عملية البرمجة

المتغير له قيمة واحدة
المصفوفه لها متغير له عدة قيم

اولا:- المصفوفة احادية الابعاد 1Dimensions :-

هي مصفوفة تحوي اما على عمود واحد او على صف واحد فقط و داخل هذا الصف او العمود تحوي عدد من القيم

طريقة كتابة المصفوفة الاحادية داخل البرنامج

```
X : array [ 1..10] of integer
```

المتغير الذي نرمزه للمصفوفة

طول المصفوفة

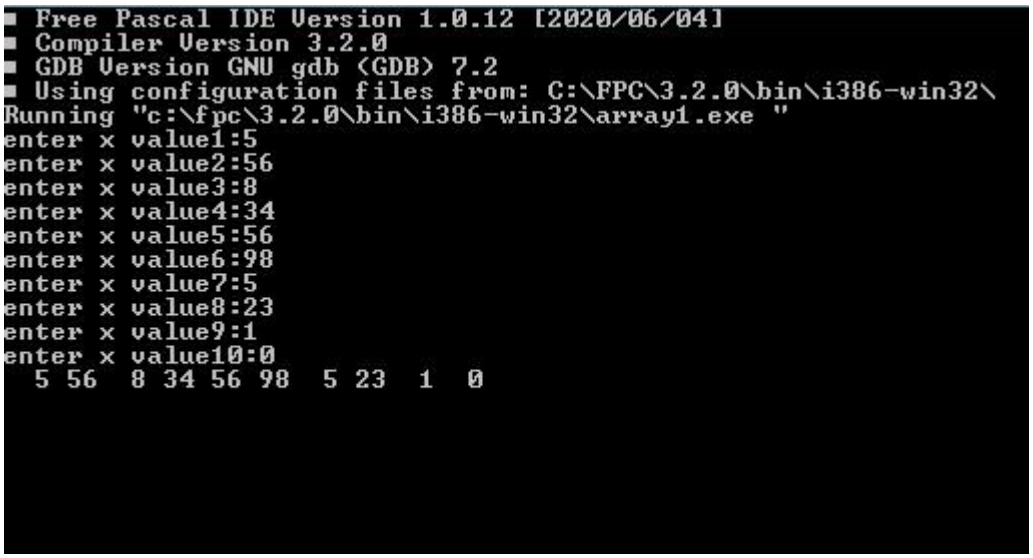
تعريف نوع المصفوفه ممكن تكون integerاو realاو charالى اخره من انواع المتغيرات

EX :- write a program to print a one-dimensional array of 10 numbers

اكتب برنامج لطباعة مصفوفة احادية الابعاد مكونه من 10 ارقام

```
program array1D;
var
x : array[ 1 . . 10 ] of integer;
i : integer;
begin
for i:= 1  to  10  do
begin
write ( ' enter x value ' , i  , ' : ' );
Readln ( x [ i ] );
end;
for i:=  1  to  10  do
write  (x [i] : 3);
readln
end.
```

شاشة التنفيذ



ثانيا :- المصفوفة ثنائية البعد 2 Dimensions:-

عباره عن متغير له اكثر من قيمة واحده و متكون من اعمدة و صفوف و الفهرس الخاص به يحوي على رقمين رقم للصف و رقم للعمود

تُرمّز الاعمده و الصفوف بهذا الشكل

	1	2	3	4
1	[1,1]	[1,2]	[1,3]	[1,4]
2	[2,1]	[2,2]	[2,3]	[2,4]
3	[3,1]	[3,2]	[3,3]	[3,4]

و تكتب المصفوفة ثنائية البعد بهذه الطريقة

```
T : array [ 1..5 , 1..5] of  integar ;
```

المتغير الذي نرمزه للمصفوفة

عدد الصفوف

عدد الاعمدة

تعريف نوع المصفوفه ممكن تكون integerاو realاو charالى اخره من انواع المتغيرات

22	[1, 1]
56	[1, 2]
8	[1, 3]
0	[2,1]
23	[2, 2]
56	[2, 3]

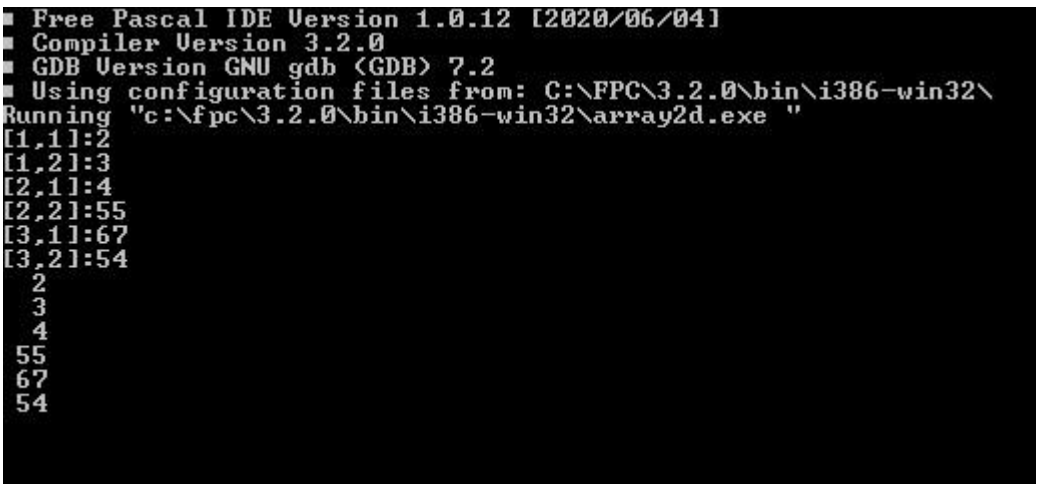
و عند تنفيذ البرنامج يظهر اما على شكل عمود واحد او سطر واحد لكن كل قيمة لديها موقع معين من الاعمدة و الصفوف

EX : write a program to print a two-dimensional array of 3 row and 2 column

اكتب برنامج لطباعة مصفوفه ثنائية الابعاد تحوي على ثلاث صفوف و عمودين

```
program  array2D ;
var
t : array [ 1 .. 3 , 1..2 ]  of  integer ;
i , j : integer ;
begin
for  i  := 1  to  3  do
for   j  :=1  to  2  do
begin
write  ( '[' , i , ',' , j , ']:' ) ;
readln ( t [ i , j ] ) ;
end ;
for  i:=  1  to  3  do
for  j:=  1  to  2  do
write  ( t [ i , j ] :3 ) ;
readln
end.
```

شاشة التنفيذ



Files

الملفات

الملف :- عبارة عن مكان يبحث فيه البرنامج عما يحتاجه وفق اسلوب منطقي و بشكل مستقل و تضع هذه الملفات ضمن القرص الصلب في الحاسبة
نستطيع تخزين انواع متعددة من المعلومات ضمن الملفات قد تكون (سلسلة string) او قيم عددية

الملفات النصيه :-
الملف في الواقع عباره عن مفهوم منطقي لا يدل على نوع المعلومات الموجوده ضمن هذا الملف لاننا كما ذكرنا سابقا نستطع تخزين انواع متعدده من المعلومات ضمن الملفات قد تكون سلاسل رمزيه او قيم عدديه لذلك يطلق على الملف النصي (text file) بملف ASCII حيث يحتوي هذا الملف على معلومات مشفره ممثله بشفرة ASCII و يمكننا قراءة الملفات النصيه بواسطة محرر نصوص text editor

الايعازات read , readln , write , writeln
تستخدم في عمليات الادخال و الاخراج التي تطبق على الملفات النصيه

بعض الايعازات المهمة المستخدمة في برمجة الملفات :-

Assign:- يعمل على ربط اسم الملف بنوعه
Rewrite:- يعمل على انشاء ملف جديد
Close :- يعمل على غلق الملف (الحفظ قبل الغلق)

EX :-write a program in pascal to create a new text file and then write the number (7870) inside the file and when it end type 'Done' ?

اكتب برنامج بلغة باسكال لانشاء ملف نصي جديد و قم بكتابة العدد 7870 في داخله و عند الانتهاء اطبع العباره Done

```
program filetest ;
const
filename = ' f : / firstclass ' ;
var
samplefile : text ;
begin
assign (samplefile , filename) ;
rewrite  (samplefile);
writeln ( samplefile , 7870) ;
writeln (' done ') ;
readln
end .
```

في حال كان البرنامج بهذا الشكل سوف نرى ان الملف موجود على القرص الصلب F و لكن بحجم 0byte و محتواه فارغ اذا اين ذهبت القيمة العدديه 7870 المكتوبه في البرنامج ؟؟

في الحقيقه هذه القيمة ضاعت بسبب عدم التخزين قبل الانتهاء من البرنامج و بهذا علينا استخدام الايعاز close لغرض حفظ محتوى الملف و من ثم غلق الملف
و بهذا سيكتب البرنامج بهذا الشكل

```
program filetest ;
const
filename = ' f : / firstclass ' ;
var
samplefile : text ;
begin
assign (samplefile , filename ) ;
rewrite ( samplefile ) ;
Writeln (samplefile , 7870 ) ;
Close (samplefile) ;
writeln(' done ' ) ;
readln
end .
```

ملاحظه :- يتم تعريف مكان حفظ الملف ضمن تعريف الثوابت