



البرمجة بلغة باسكال

مفاهيم برمجية

يحتوي هذا الملف سلسلة محاضرات تم اعطاؤها في

مركز الشهيد باسل الأسد للتدريب على الحاسوب

أمجد طه

1434/02/01

الانتقال من الخوارزمية الى البرنامج :

1- القراءة

التعبير عن القراءة عند كتابة خوارزمية :

اقرأ A

في لغة باسكال :

Readln (A); Read (A);

2- الكتابة

الخوارزميات : اكتب A , " the value is "

:"

3- الاسناد :

الخوارزميات A:=B3 +C

في لغة باسكال :

A:= B * B * B + C

4- التعليمات الشرطية

أ- الشرط

الخوارزميات

إذا (A<0) نفذ

<مجموعة تعليمات >

في لغة باسكال :

If (A<0)then

Begin

.....

End

ب- الاختيار

الخوارزميات

إذا ($A < 0$) نفذ

<مجموعة تعليمات >

والا

<مجموعة تعليمات >

لغة باسكال

If ($A < 0$) then

Begin

.....

End

Else

Begin

.....

End

5- التكرار

الخوارزميات
تكرار مجموعة من التعليمات طالما شرط محقق
طالما (i أصغر أو يساوي n) كرر

<مجموعة تعليمات >

لغة باسكال :

While (i<= n) do

Begin

.....

End

انواع المتغيرات

العمليات الحسابية في لغة باسكال :

1- العمليات الحسابية الأربعة

+ الجمع

- الطرح

* الضرب

/ القسمة

أمثلة

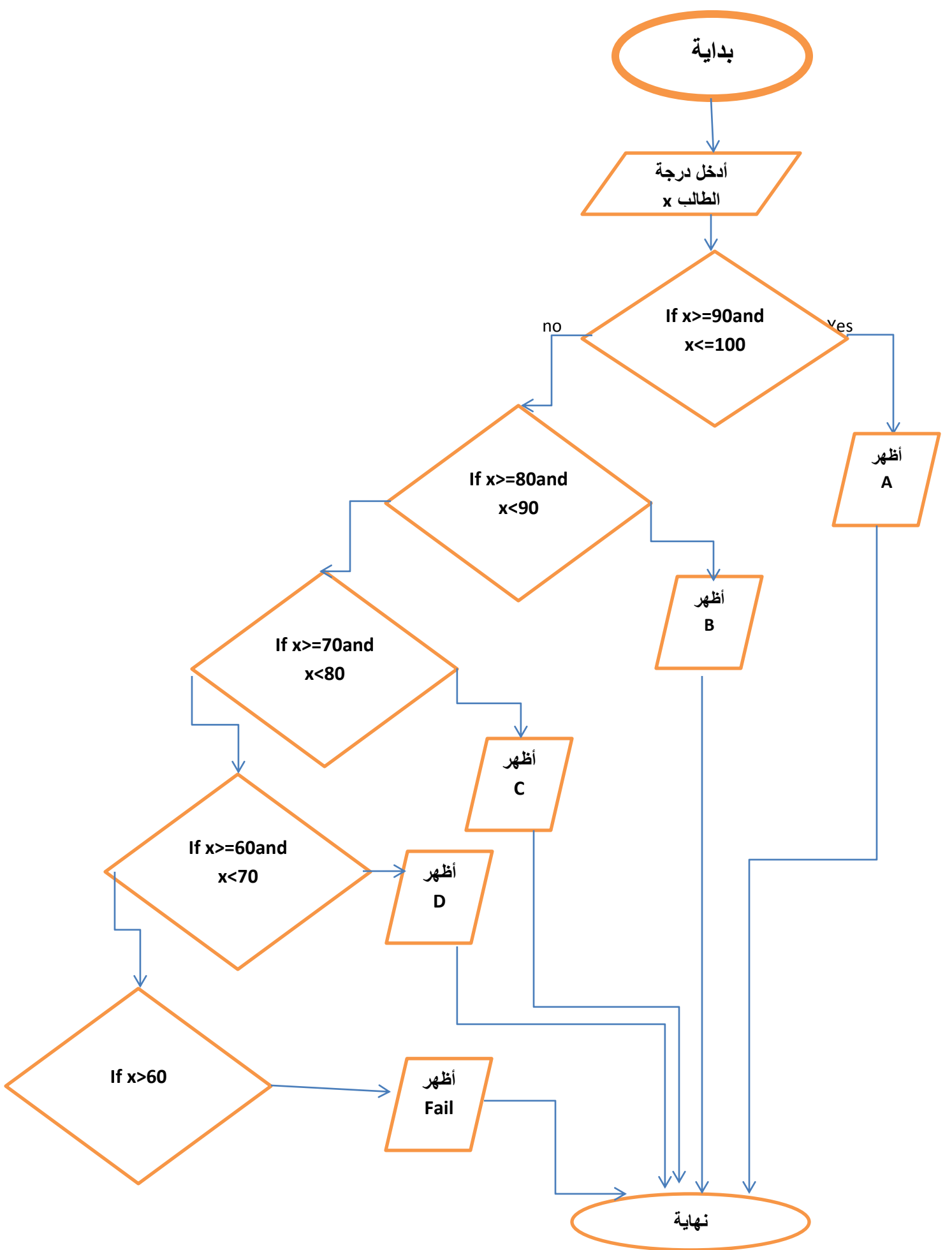
C := a + b;

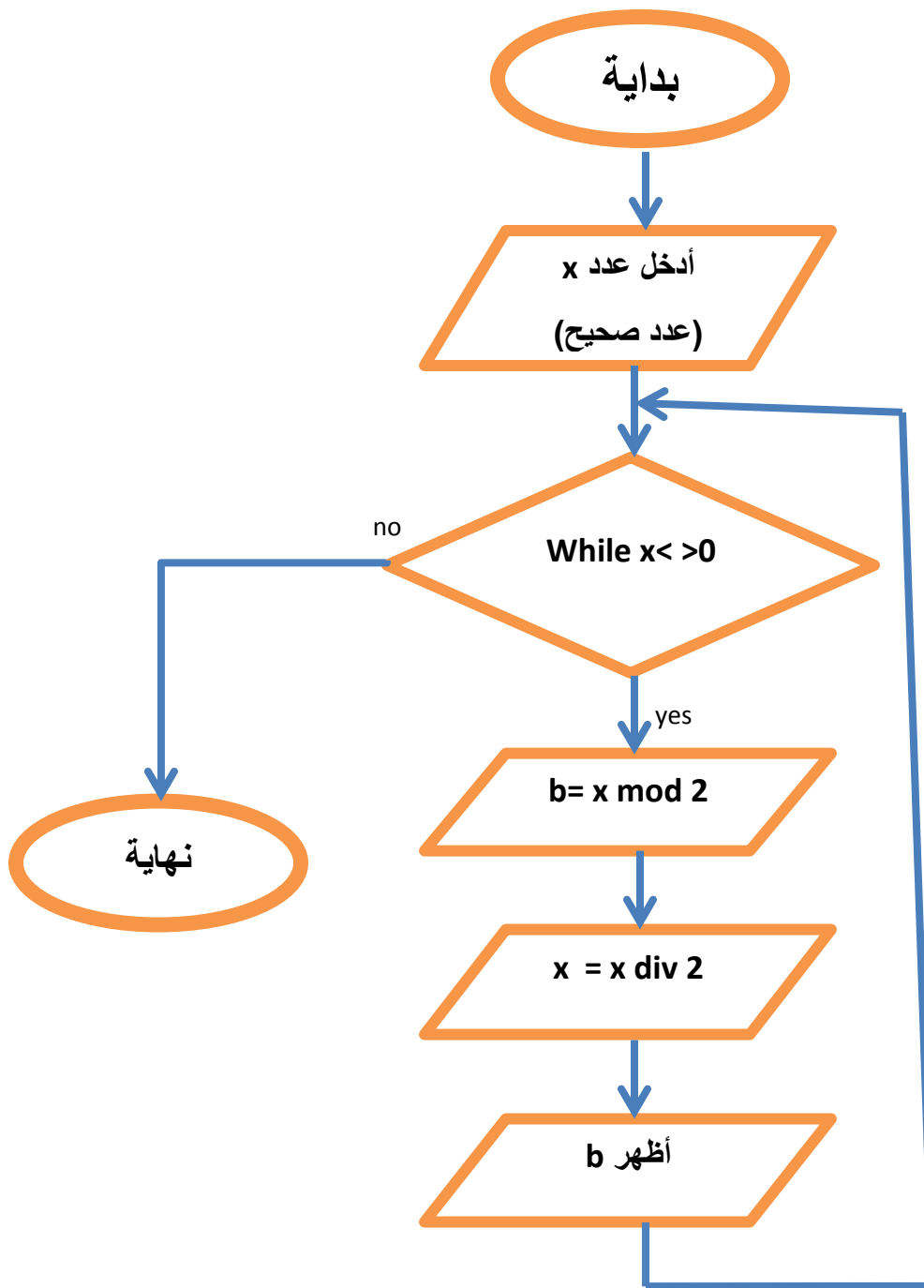
S := a - b;

E := a * b;

D := a/b;

الرمز	العملية	نوع المتغير
+	الجمع	Integer
-	الطرح	
*	الضرب	
div	القسمة	
mod	باقي القسمة	
+	الجمع	Real
-	الطرح	
*	الضرب	
/	القسمة	





تطبيق للمخطط السابق (برنامج لتحويل عدد صحيح الى عدد ثنائي)

```
program dectobinary;  
var x,b:integer;  
begin  
  writeln('enter decimal number');  
  readln(x);  
  while(x<>0) do  
    begin  
      b:= x mod 2;  
      x:= x div 2;  
      writeln(b);  
    end  
  end.  
end.
```


برنامج لحساب القاسم المشترك الأكبر لعددين

```
program findgcd;
var a,b :integer;
gcd:integer;
begin
writeln('enter two numbers');
readln(a,b);
while (a<>b)do
begin
if(a>b) then
begin
a:=a-b;
end
else
b:=b-a;
end;
gcd:=a;
writeln('grand common divider =',gcd);
end.
```

برنامج لحساب العدد الأكبر و الأصغر و المتوسط الحسابي لمجموعة من الأعداد

```
program minmax;
var n,i : integer;
x,xmax,xmin,xmean:real;
begin
readln(n);
readln(x);
xmin:=x;xmax:=x;xmean:=x;
i:=1;
while(i<n) do
begin
readln(x);
if(x<xmin) then
xmin:=x;
if(x>xmax)then
xmax:=x;
xmean:=xmean+x;
i:=i+1;
end;

writeln('Max=',xmax);
writeln('Min=',xmin);
writeln('Mean=',xmean/n);
end.
```

استخدام المميز Using the discriminant

من القانون العام لحل المعادلة $ax^2 + bx + c = 0$ تكون الصورة العامة لجذري المعادلة كالآتي:

$$x = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \text{ و } x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

يسمى $\Delta = b^2 - 4ac$ المميز، قد يكون الناتج عدداً موجباً أو صفراً أو عدداً سالباً لأنه يميز لنا نوع جذري المعادلة من حيث كونهما: عددين حقيقيين مختلفين، إذا كان المميز موجباً أو عددين حقيقيين متساويين، إذا كان المميز يساوي صفراً أو لا حلول حقيقية، إذا كان المميز سالباً ويتضح ذلك من الأمثلة الآتية:

مثال (١)

أوجد نوع جذري المعادلة

$$2x^2 + 2x - 3 = 0$$

الحل

$$a = 2, b = 2, c = -3$$

المميز

$$b^2 - 4ac = 2^2 - 4 \times 2 \times (-3) = 4 + 24 = 28$$

وحيث إنه عدد موجب

إذن: الجذران عدداً حقيقيان مختلفان

يمكن التحقق من ذلك بحل المعادلة

$$2x^2 + 2x - 3 = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x = \frac{-2 \pm \sqrt{28}}{2 \times 2}$$

$$x = \frac{-2 + \sqrt{28}}{4} = \frac{-2 + 5.29}{4} = \frac{3.29}{4} = 0.82$$

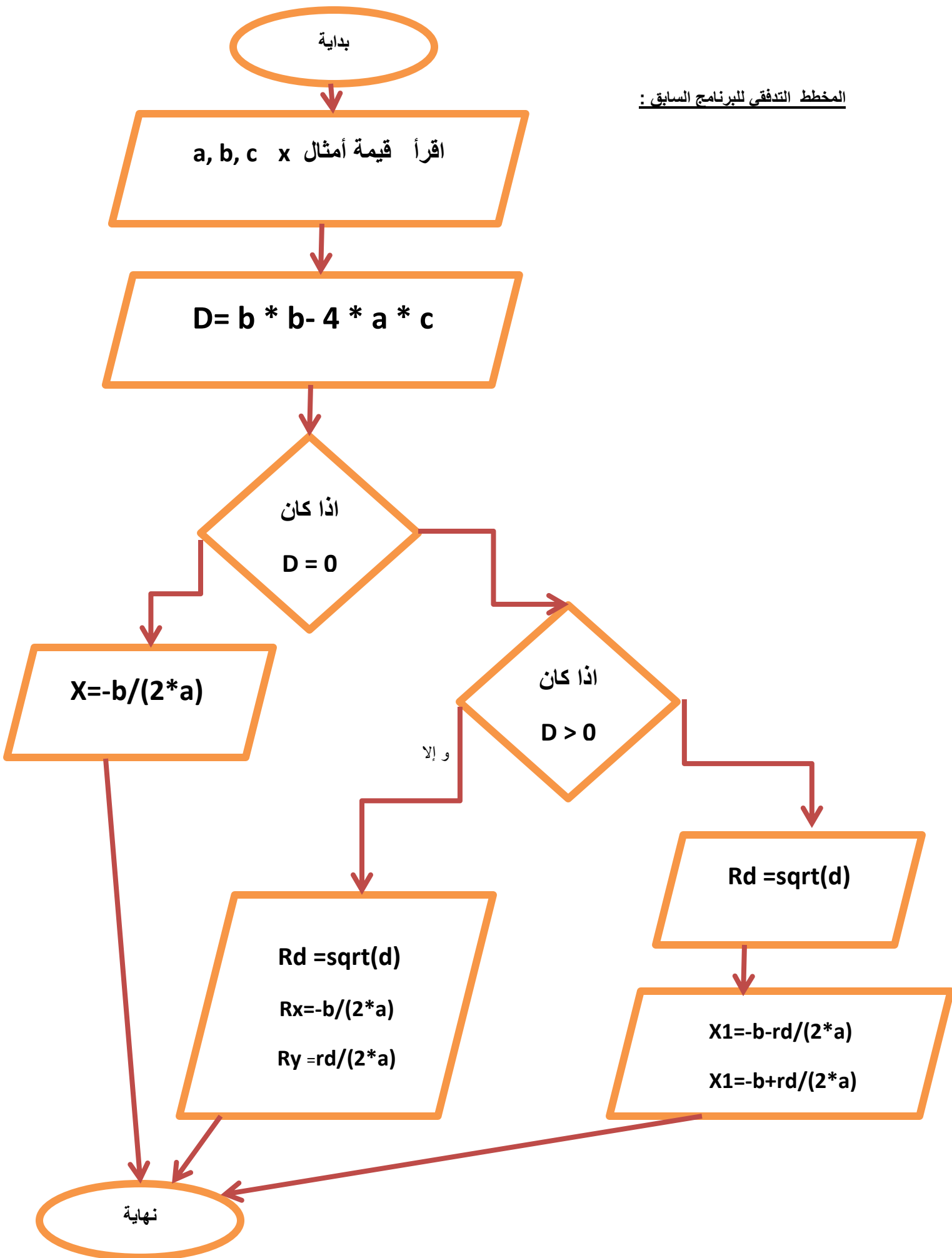
$$x = \frac{-2 - \sqrt{28}}{4} = \frac{-2 - 5.29}{4} = \frac{7.29}{4} = -1.82$$

ومن الواضح أن الجذرين عبارة عن عددين حقيقيين مختلفين.

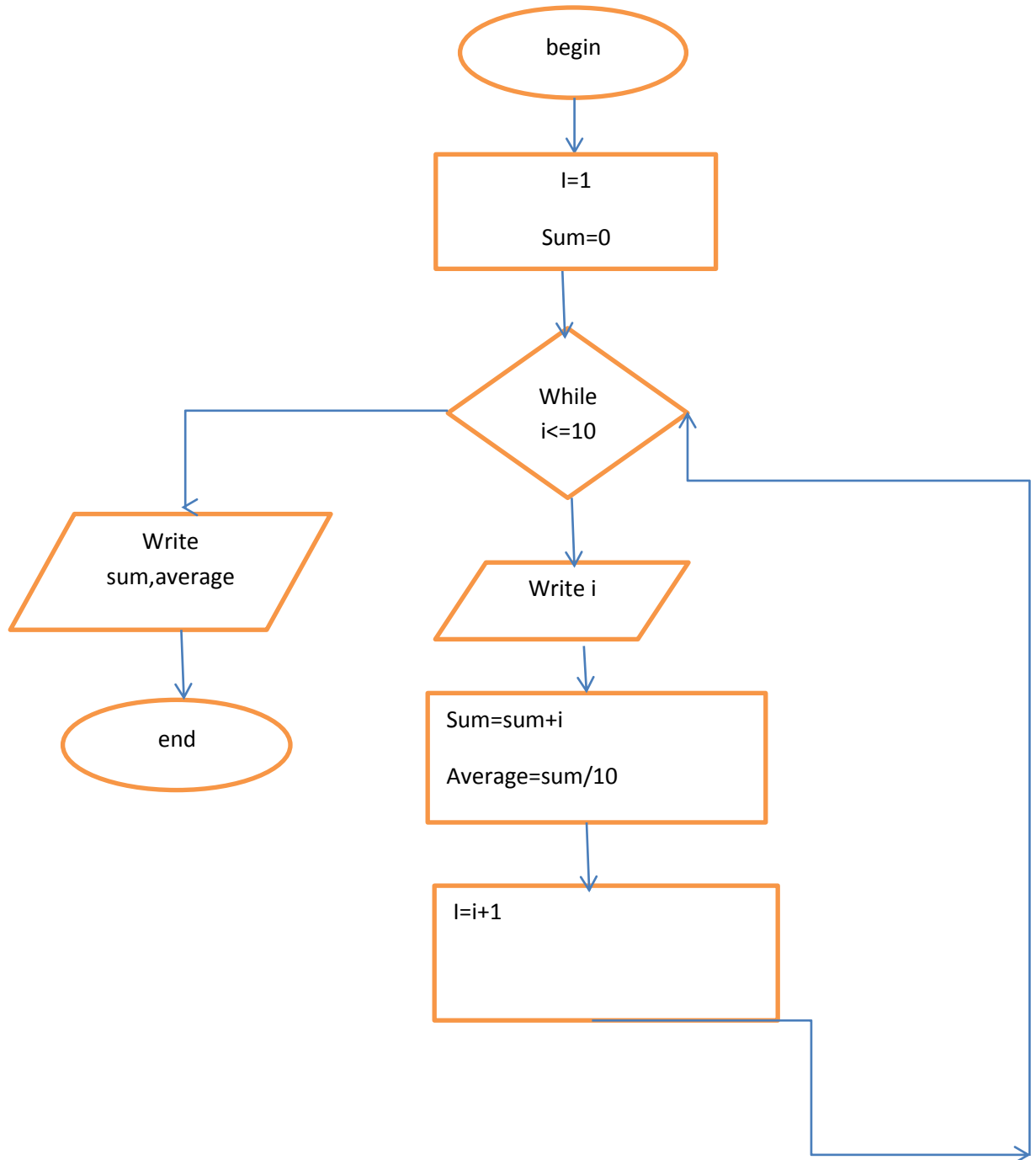
برنامج لحل المعادلات من الدرجة الثانية

```
program equation;
var a,b,c :real;
d,rd,rx,ry,x,x1,x2:real;
begin
writeln('equation of type ax2+bx+c Enter values for a b c');
readln(a,b,c);
if(a=0) then
begin
writeln('equation is not second degree');
if(b<>0)then
writeln('solution=',-c/b)
else
writeln('Equation is triaval or not correct');
end
else
begin
d:=b*b-4*a*c;
if(d=0)then
begin
x:=-b/(2*a);
writeln('identical roots');
end
else
if(d>0) then
```

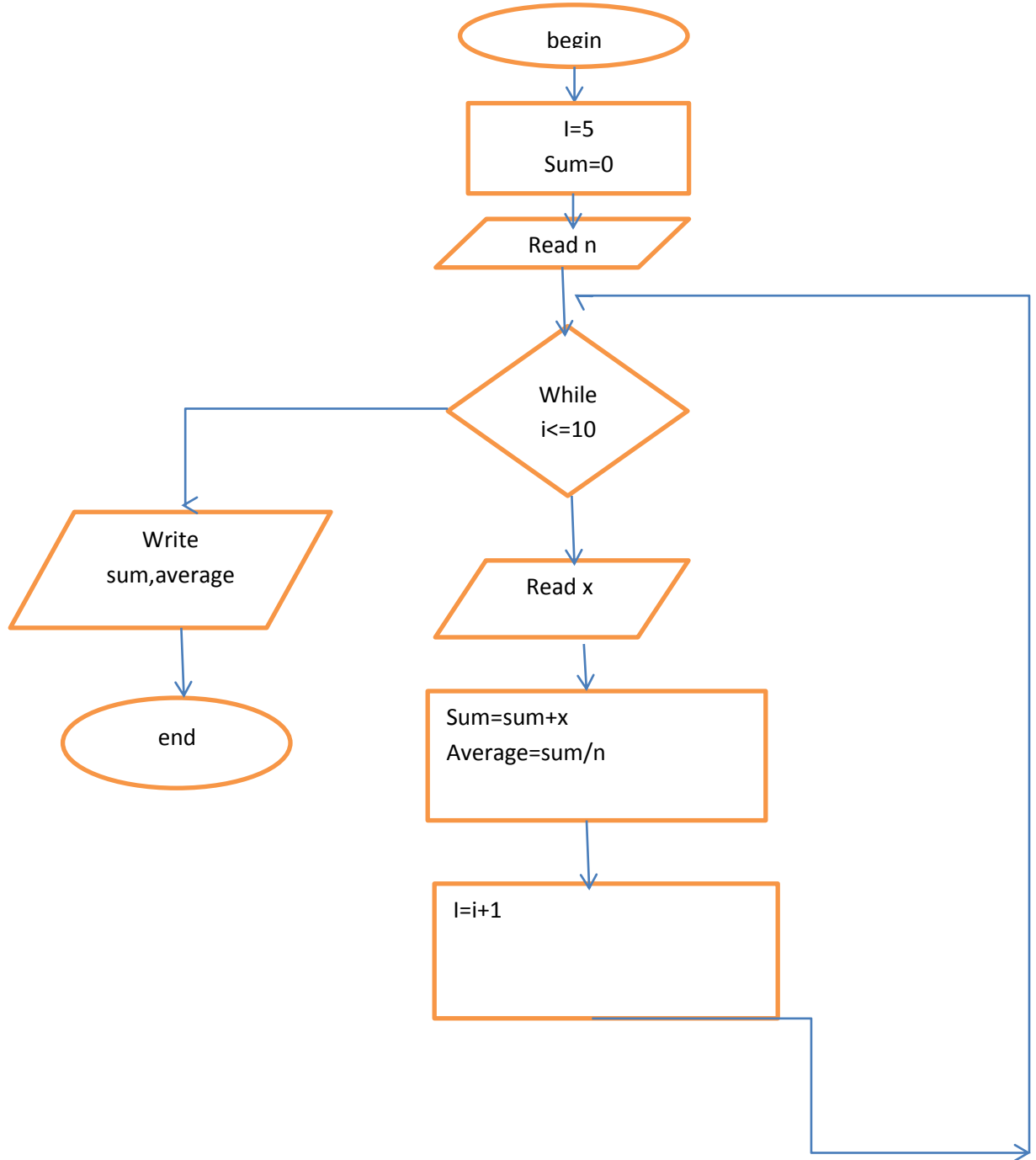
```
begin
rd:=sqrt(d);
x1:=(-b-rd)/(2*a);
x2:=(-b+rd)/(2*a);
writeln('two real roots');
writeln('x1=',x1,'x2=',x2);
end
else
begin
rd:=sqrt(abs(d));
rx:=-b/(2*a);
ry:=rd/(2*a);
writeln('complex roots:');
writeln('x1=',rx,'+i',ry);
writeln('x2=',rx,'-i',ry);
end;
end;
end.
```



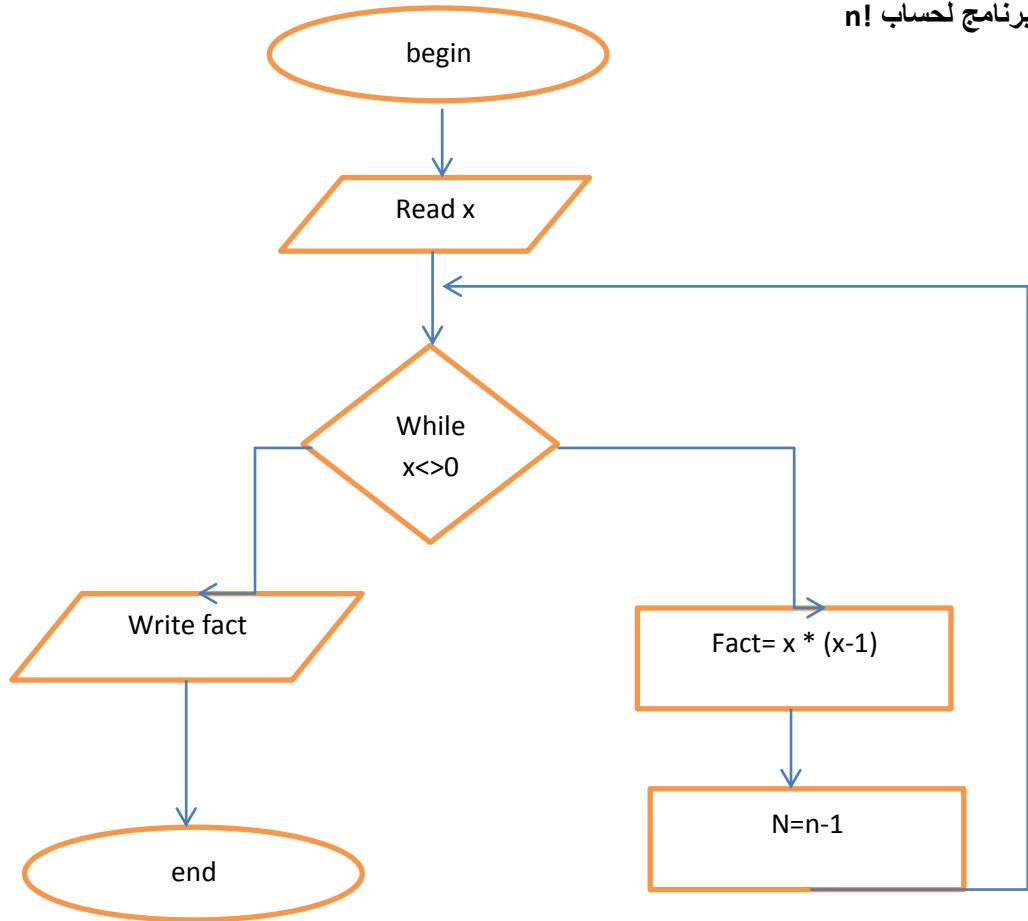
اكتب برنامج بلغة باسكال يقوم بطباعة الأعداد من 1 الى 10 ثم طباعة المجموع و المتوسط الحسابي لهذه الأعداد:



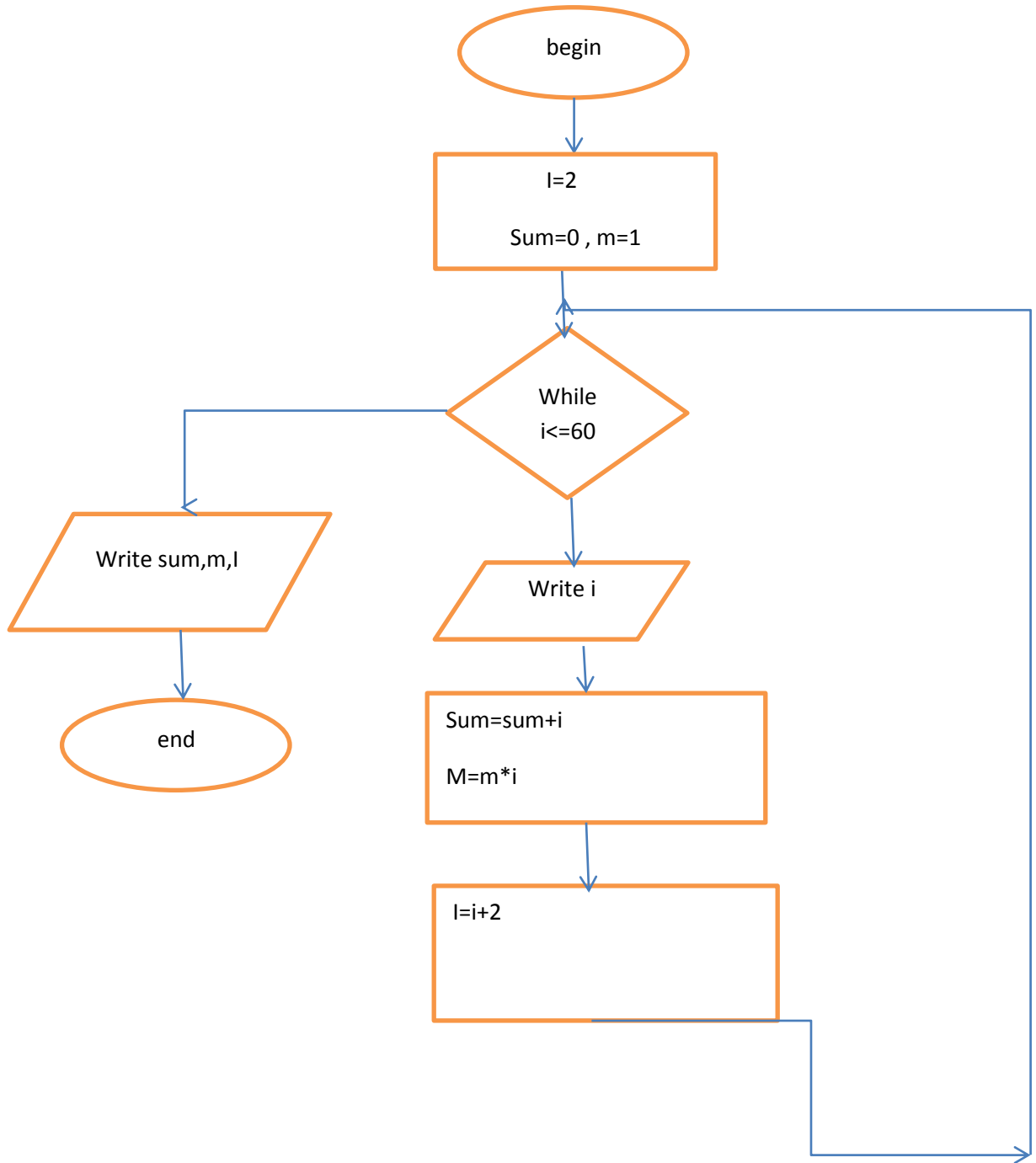
اكتب برنامج بلغة باسكال يقوم بقراءة مجموعة من الأعداد من لوحة المفاتيح إيجاد المجموع و المتوسط الحسابي لهذه الأعداد:



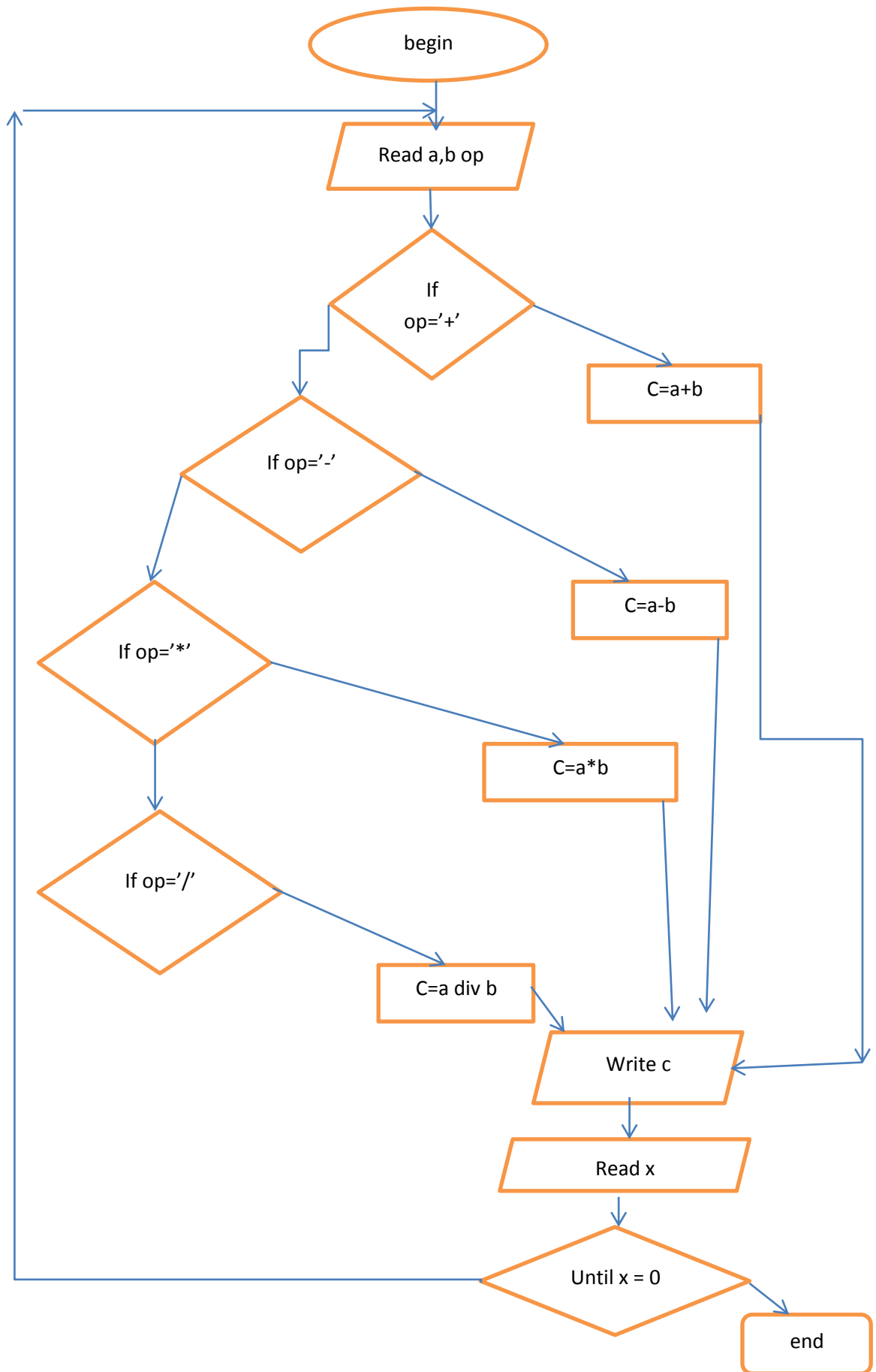
برنامج لحساب n!



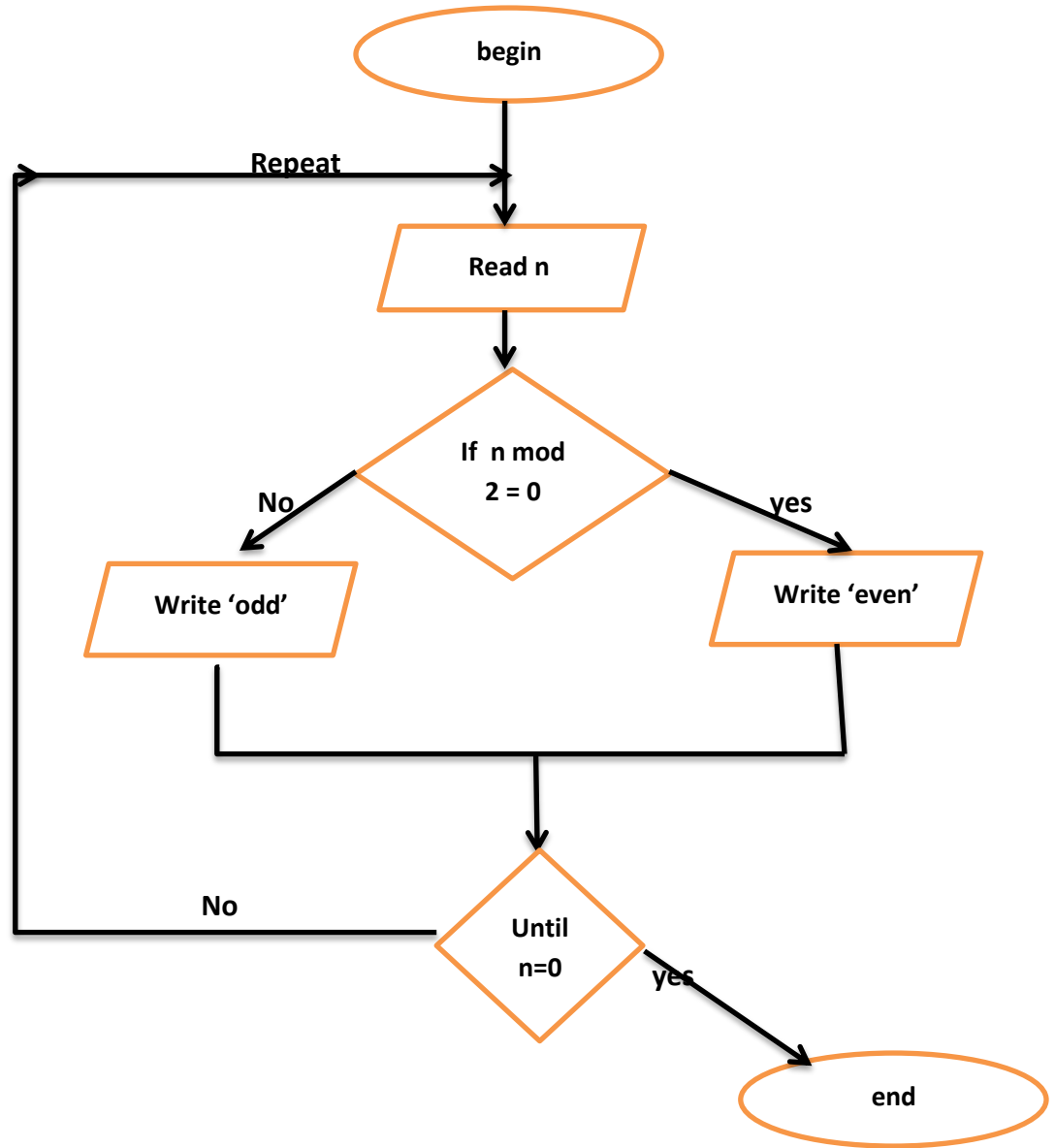
اكتب برنامج بلغة باسكال يقوم بحساب مجموع و جداء الأعداد الزوجية من 2 حتى 60:



Repeatuntil



اكتب برنامج بلغة باسكال يحدد اذا كان العدد الصحيح المدخل من لوحة المفاتيح زوجي ام فردي



المصفوفات :

تعاملنا في الدروس السابقة مع التعليمات الشرطية و التكرار و لم يكن من الممكن التحكم بتخزين المتغيرات او استعادتها او معرفة اماكن توضعها في الذاكرة .

مفهوم المصفوفة : هي بنية معطيات مؤلفة من تجميع عناصر من نفس النمط يمكن الوصول الى أي عنصر من خلال دليل أو عنوان لهذا العنصر $a[i]$

A[1]	a[2]	a[3]	a[4]
i=1	i=2	i=3	i=4

تم حجز اربع مواقع في الذاكرة للمصفوفة السابقة حيث يمكن كتابة و قراءة المتغيرات الى تلك المواقع على فرض قمنا باستخدام تعليمة $read(a[i]);$ التي تسمح للمستخدم بإدخال البيانات من لوحة المفاتيح

و أدخلنا الأرقام التالية على التالي 50 40 30 20

فسوف تخزن بالشكل التالي

A[1]	a[2]	a[3]	a[4]
50	40	30	20
i=1	i=2	i=3	i=4

و اذا قمنا بطباعة (اظهار على الشاشة) ما هو موجود في الذاكرة باستخدام تعليمة $write();$

فمثلاً: $Write(a[2]);$

النتيجة ستكون 40

يتم التعريف عن المصفوفة

$a:\text{array}[1..4]\text{of integer};$

حيث ان القيمة الابتدائية للمصفوفة هي 1 و النهائية 4

كلمة of ضرورية و بما ان المصفوفة تجميع للبيانات من نوع واحد يمكن التصريح عن جميع انواع المتغيرات

$a:\text{array}[1..4]\text{of string};$

$a:\text{array}[1..4]\text{of char};$

$a:\text{array}[1..4]\text{of boolean};$

$a:\text{array}[1..4]\text{of real};$

برنامج بسيط يستخدم مصفوفة احادية (اي شعاع واحد) و قراءة عناصر هذا الشعاع من المستخدم

```
program array2;
var i:integer;
a:array[1..4]of integer;
begin
  for i:=1 to 4 do
    readln(a[i]);
  for i:=1 to 4 do
    writeln('a['',i,'']=',a[i]);
end.
```

تعليلة الانتقال المطلق (غير المشروط) Goto

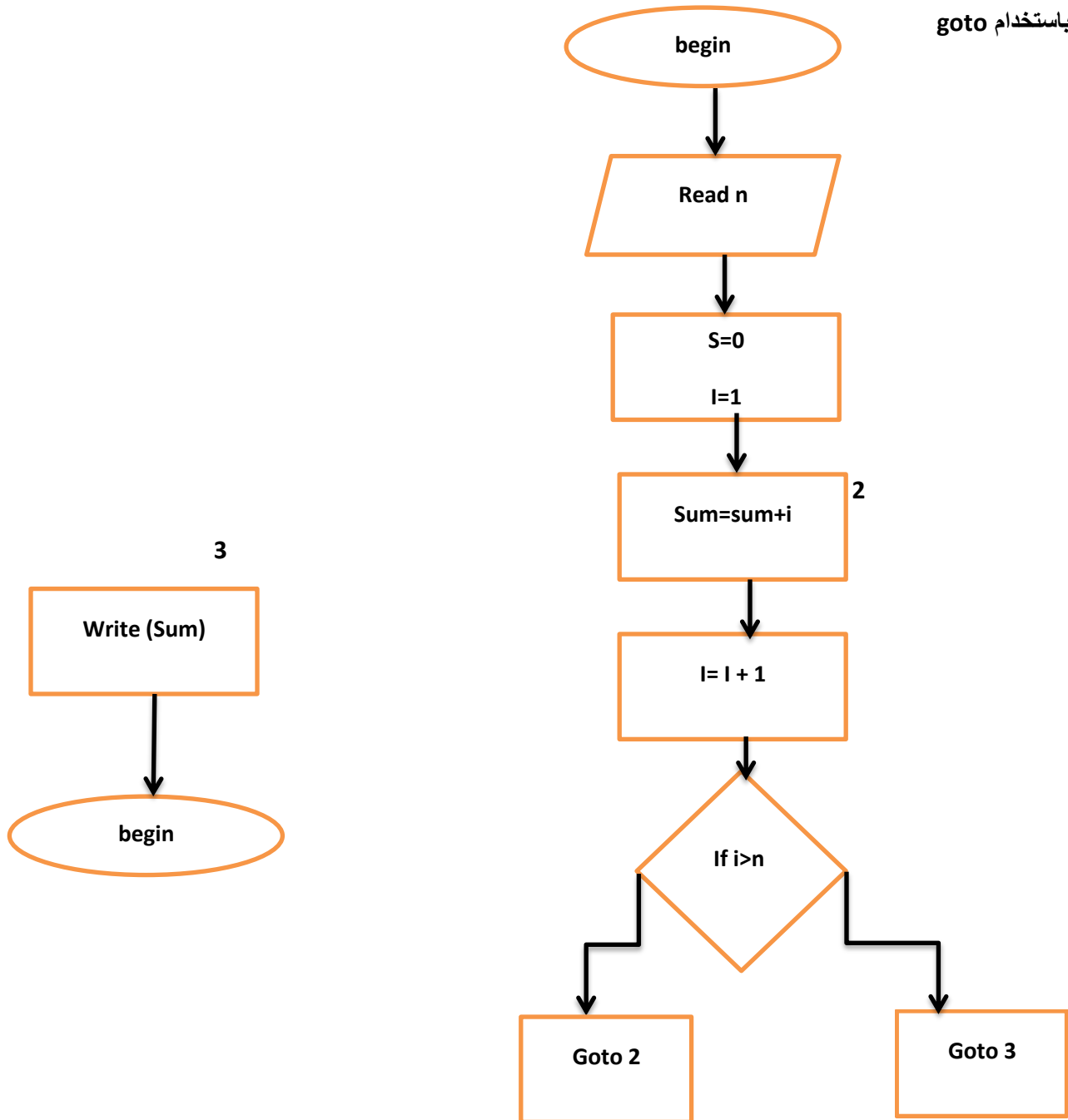
تقوم هذه التعليلة بتكرار مجموعة من التعليلات والجمال المحصورة بين قيمة مميزة التي يراد الانتقال إليها وبين تعليلة الانتقال عدد غير منته .

الصيغة العامة : Goto variable/number ;

حيث يمكن أن يكون العنوان رقم سطر أو متحول (عددي – حرفي) يتم التعريف عن هذا العنوان في قسم التعريف بكلمة label ; متحول أو رقم label

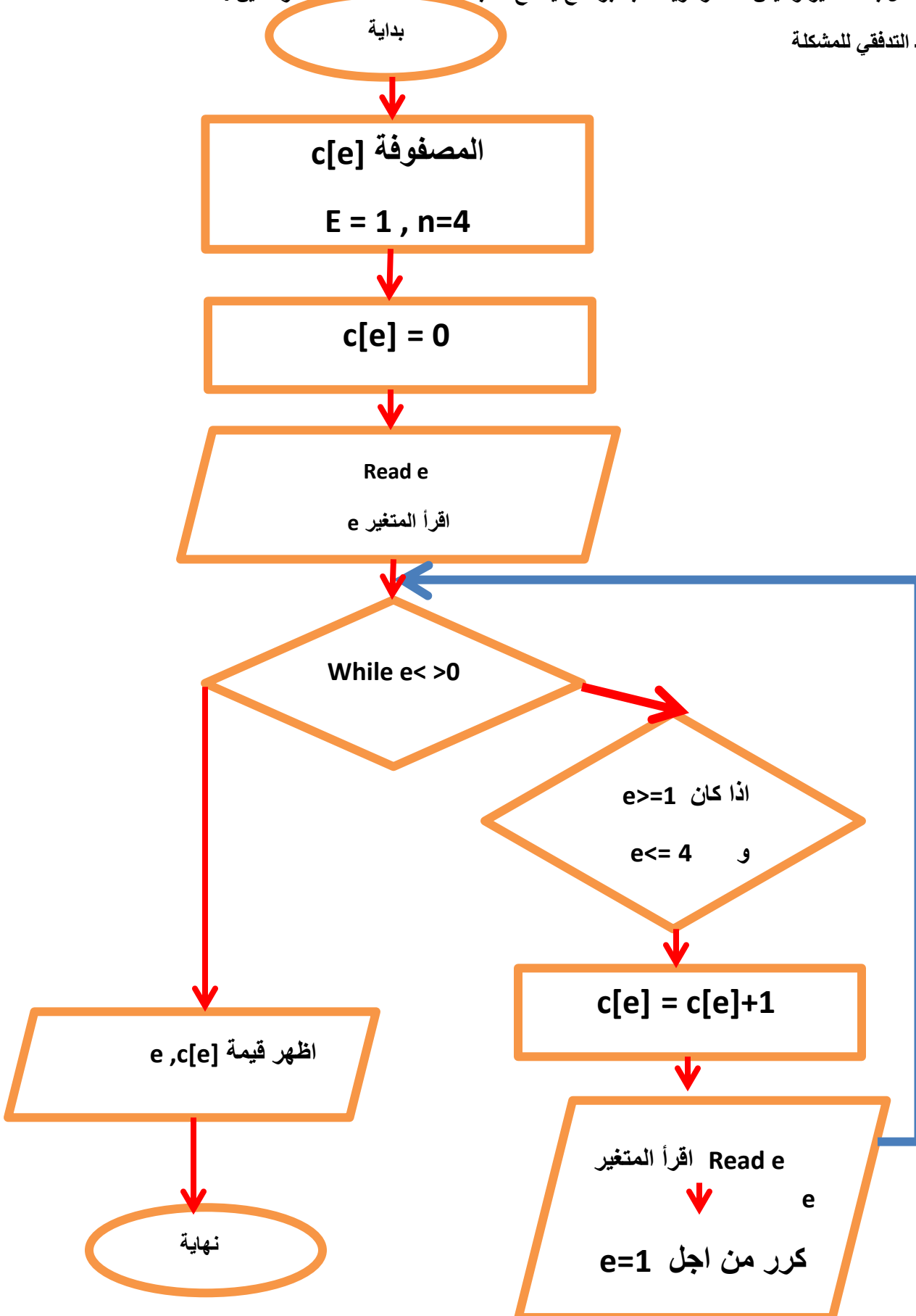
مثال : اكتب برنامجاً بلغة باسكال يقوم بطباعة مجموع الأعداد من 1 وحتى N .

باستخدام goto



لدينا اربع مرشحين لانتخابات تصل نتائج الاقتراع على شكل دخل عددي كل عدد على سطر يمثل رقم المرشح المنتخب
ننهي الدخل بعدد مميز و ليكن الصفر نريد كتابة برنامج يسمح لنا بحساب الاصوات لكل من المرشحين .

المخطط التدفقي للمشكلة



البرنامج :

```
program election;
var i:integer;
c:array[1..4]of integer;
begin
for i:=1 to 4 do
  c[i]:=0;
  readln(i);
  while(i<>0) do
  begin
if (1<=i) and (i<=4) then
  c[i]:=c[i]+1;
  readln(i);
  end;
for i:=1 to 4 do
  writeln('C',i,'=',c[i]);
end.
```

مثث باسكال

1	0	0	0	0	
1	1	0	0	0	
1	2	1	0	0	
1	3	3	1	0	
	1	4	6	4	1

$p[l,z]$

مصفوفة ثنائية $5 * 5$

[1,1]	[1,2]	[1,3]	[1,4]	[1,5]
[2,1]	[2,2]	[2,3]	[2,4]	[2,5]
[3,1]	[3,2]	[3,3]	[3,4]	[3,5]
[4,1]	[4,2]	[4,3]	[4,4]	[4,5]
[5,1]	[5,2]	[5,3]	[5,4]	[5,5]

الخطوة الأولى نجعل قيمة المصفوفة 0

$A[l,z] := 0;$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

0	0	0	0	0
0	0	0	0	0

$A[i,1] = 1$ الخطوة الثانية

1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0

الخطوة 3

for i:= 2to 5do

for j:= 2to 5do

1	0	0	0	0
1	0	0	0	0
1	0	0	0	0

1	0	0	0	0
1	0	0	0	0

الخطوة 4

$$a[i,j] := a[i-1,j] + a[i-1,j-1];$$

1	0	0	0	0
1	1	0	0	0
1	2	1	0	0
1	3	3	1	0
1	4	6	4	1

