Chapter Three
DO Construct In FORTRAN 90

Introduction المقدمـــــة

في كل الأمثلة السابقة كانت تنفذ أي عبارة مرة واحدة ولكن في كثير من الأحيان نحتاج الى تكرار مجموعة من العبارات لعدد من المرات ولتحقيق هذا الغرض نستخدم تركيبة الدوران أو التكرار DO (DO Construct) وتسمح تركيبة DO بتكرار أي جملة أو مجموعة جمل عدداً معيناً من المرات أو تترك عدد تكرار الجمل مفتوحاً كي تنهيه جملة ظرفية (EXIT , CYCLE) (سنأتي على ذكرها) ٠ وتستعمل تركيبة DO لإيجاد قيم المتواليات والمصفوفات وقيم عمليات الجمع والضرب المتكررة وأي عملية تحتاج إلى عدد من التكرارات .

هناك نوعين من تركيبة التكرار DO Construct في لغة Fortran 90 هي:

1٠ تركيبة الدوران المحددة Loop Control With a DO Construct

هذا النوع من تركيبة الدوران تكون محددة التكرار أي عدد مرات التكرار تكون محددة مسبقاً ٠

والشكل العام لتركيبة DO المحددة كما يلي :

```
DO counter = initial, final, increment
   ...
   ...   } block of statements
   ...
END DO
```

حيث :

counter : متغير عددي (يفضل متغير صحيح للحلقات الدورية) يمثل عداد الحلقة التكرارية ويسمى المؤشر أيضاً

initial : القيمة الابتدائية للعداد وهي أما أن تكون ثابت أو متغير عددي معرف أو تعبير رياضي ٠

final : القيمة النهائية للعداد وهي أما أن تكون ثابت أو متغير عددي معرف أو تعبير رياضي ٠

increment : مقدار الزيادة للعداد وهي أما أن تكون ثابت أو متغير عددي معرف أو تعبير رياضي وهي تضاف في كل مرة للقيمة الابتدائية للعداد (***initial***) حتى يصل إلى القيمة النهائية للعداد (***final***) ٠

مثال:

```
DO Counter = 1, 10 , 2
   PRINT *, Counter
END DO
```

أي أن قيمة عداد الحلقة الابتدائية هو 1 والقيمة النهائية 10 وقيمة الزيادة المنتظمة هي 2 ٠ وفي هذا المثال تتكرر جملة PRINT *, Counter خمس مرات ويطبع الأرقام الفردية من 1 إلى 10 ٠

مثال:

```
K = 10
DO Counter = K, K*K , 2
  PRINT *, Counter
END DO
```

وهذا المثال يوضح أن القيمة الابتدائية والقيمة النهائية ومقدار الزيادة ممكن أن تكون ثابت أو متغير أو تعبير رياضي كما تم ذكره في التعريف .

مثال:

```
DO Counter = 100, 1 , -1
  PRINT *, Counter
END DO
```

أن هذا الجزء من البرنامج يطبع الأرقام من 100 إلى 1 ويوضح أيضاً أن تسلسل قيمة عداد الحلقة لا يشترط أن يكون متصاعداً فإذا كان مقدار الزيادة سالباً فأن القيم التي يأخذها عداد الحلقة تكون متناقصة .

ملاحـــظات مهمة لاستخدام تركيبة DO المحددة :

١ - إذا لم يذكر ***increment*** (مقدار الزيادة) يفهم انه يساوي 1 .

مثال:

```
DO I = 1, N
```

يعني هذا المثال أن الدليل I يأخذ القيمة الابتدائية 1 والقيمة النهائية N بزيادة مقدارها 1 في كل تكرار .

٢ - لا يجوز أن يكون مقدار الزيادة (***increment***) يساوي صفر .

٣ - لا يجوز أعادة تعريف أي من متغيرات تركيبة التكرار داخل حلقة التكرار ومثال ذلك:

```
DO  I = 1 , 20
  I = I+1          <-------- لا يجوز
  X(I) = Y(I) + Z(I)
END DO
```

٤ - عداد الحلقة ويسمى أحياناً متغير الدوران هو متغير عددي (صحيح أو حقيقي) ولا ينصح باستخدام الأعداد الحقيقية لمتغير الدوران .

٥ - تركيبة التكرار لا تنفذ في الحالات التالية :

A - إذا كان مقدار الزيادة أكبر من صفر والقيمة الابتدائية أكبر من القيمة النهائية أي ***initial*** > ***final*** , ***inc.*** > 0)

مثل :

```
DO  I = 20 , 10 , 2
```

B - إذا كان (***initial*** < ***final*** , ***inc.*** < 0) مثل :

```
Do  I = -5 , 10 , -1
```

وفي حالة عدم تنفيذ تركيبة التكرار DO فإن أول عبارة تنفذ بعد ذلك هي أول عبارة تنفيذية تلي العبارة END DO

٦ - عدد مرات تكرار تركيبة التكرار يمكن إيجادها من العلاقة الآتية :

$$NO. = INT\left(\frac{final-initial+increment}{increment}\right)$$

مثال : أكتب برنامج لإيجاد قيمة المفكوك N ؟

الجواب :

```
Program Factorial
Implicit none
Integer :: N, I, Fact
Read*,N
 Fact =1
Do I = 1, N
Fact=Fact * I
End Do
Print*, N , "!=", Fact
End Program Factorial
```

حلقات التكرار المتداخلة Nested Do Loops

قد تحتوي أحدى حلقات التكرار على حلقة تكرار أخرى أو أكثر وتسمى الحلقات التكرارية في هذه الحالة بالحلقات المتداخلة والقواعد المستخدمة للحلقات المتداخلة هي نفس القواعد المستخدمة لحلقة تكرار واحدة، ومع ذلك يمكن مراعاة النقاط الآتية :

١ - يجب أن يختلف عداد الحلقات الداخلية عن عداد الحلقات الخارجية،

٢ - يجب أن تكون الحلقة التكرارية موجودة بكاملها داخل مدى الحلقة التكرارية الخارجية أي غير متقاطعة،

```
DO  I=
    -----
DO  J=
    -----
    -----
 DO  K=
    -----
    -----
 END DO
    -----
END DO
    -----
END DO
```

```
DO  I=
    -----
DO  J=
    -----
    -----
END DO
    -----
 DO  K=
    -----
    -----
END DO
    -----
END DO
```

```
DO  I=
DO  J=
-----
-----
END DO
END DO
```

حلقات متداخلة مسموح بها

حلقات تكرارية متداخلة غير مسموح بها لأن لها نفس العداد I

```
DO  I=
-----
DO  I=
-----
END DO
END DO
```

مثال : أكتب برنامجاً لإيجاد مجموع مفكوك الأرقام الزوجية المحصورة بين 1 ، 19 ؟

Sum = 2! + 4! + 6! + 8! + ….. + 18!

الجواب :

```
PROGRAM SUM_FACTORIAL
 IMPLICIT NONE
INTEGER :: I, J, SUM , FACT
 SUM = 0
 DO I = 2, 18, 2
  FACT = 1
   DO  J = 1, I
  FACT = FACT* J
 END DO
  SUM = SUM + FACT
END DO
   PRINT *, 'SUM =', SUM
END PROGRAM SUM_FACTORIAL
```

2. تركيبة الدوران غير المحددة (الدوران المفتوح) Loop Control With a DO Construct

هذا النوع من تركيبة الدوران يكون فيه التكرار غير محدد أي عدد مرات الدوران فيه مفتوحاً بحيث تتحكم فيه جمل ظرفية من داخله. والشكل العام لتركيبة DO غير المحددة كما يلي :

```
DO
    ...
    block of statements
    IF (logical_expression)  EXIT
    ...
    IF (logical_expression)  CYCLE
    ...
    block of statements
END DO
```

حيث :

EXIT : جملة ظرفية أذا حدث أن نفذت فإن الدوران ينتهي وينتقل الكمبيوتر إلى تنفيذ البرنامج بعد ترك التكرار (أي تنفيذ العبارات بعد جملة END DO) .

CYCLE :جملة ظرفية أذا حدث أن نفذت داخل التكرار فإنها تقفز عن مجموعة الجمل اللاحقة لها وتبدأ دورة جديدة

ملاحـــظة : أن هاتين الجملتين (EXIT , CYCLE) ظهرت في لغة FORTRAN 90 (أي لم تكن موجودة في نسخ فورتران القديمة) وهي تغني عن بعض استخدامات GOTO في نسخ فورتران القديمة .

وكما ذكرنا سابقاً فإنه يجب التخلص من GOTO قدر المستطاع .

مثال: أكتب برنامج لطباعة قوى العدد 3 أي .... , 3^3 , 3^2 , 3^1 , 3^0 ويتوقف الدوران عندما تتجاوز قيمة 3^n مقدار 2000 .

الجواب:

```
PROGRAM SOME_POWER_OF_3
  IMPLICIT NONE
  INTEGER :: POWER_OF_3
POWER_OF_3 =1              ! THE ZERO POWER_OF_3
DO
   PRINT *, POWER_OF_3
   POWER_OF_3 = 3 * POWER_OF_3
   IF (POWER_OF_3 >= 2000 ) EXIT
END DO
END PROGRAM SOME_POWER_OF_3
```

في هذا المثال نلاحظ أولاً خلو تركيبة التكرار من جملة تحديد عدد مرات الدوران (Loop Control) في بدايته . وهذا يعني أن عدد مرات الدوران يكون مفتوحاً .

لاحظ الجملة الظرفية IF (POWER_OF_3 >= 2000) EXIT

حيث يتم تنفيذ الجملة EXIT عند تحقق الشرط (التعبير المنطقي POWER_OF_3 >= 2000) وبالطبع فإنه إذا نفذت جملة EXIT ، فإن التنفيذ ينتقل الى الجملة التي تلي جملة END DO وهي نهاية البرنامج .

ومن الأمثلة العملية على الدوران المفتوح هو أن تتحكم أحدى معطيات الإدخال في إنهاء الدوران . وغالباً ما تنتهي سلسلة من المعطيات الداخلة برقم مقصود كما في المثال التالي :

```
PROGRAM AVERAGE
  IMPLICIT NONE
  INTEGER :: NUMBER , NUMBER_OF_NUMBERS , SUM
SUM =0
NUMBER_OF_NUMBERS = 0
DO
   READ *, NUMBER
   IF (NUMBER < 0 ) EXIT
   PRINT *, "Input data :", NUMBER
   SUM = SUM + NUMBER
   NUMBER_OF_NUMBERS = NUMBER_OF_NUMBERS + 1
END DO
PRINT *, " The average of the numbers is " , REAL(SUM) / NUMBER_OF_NUMBERS
END PROGRAM AVERAGE
```

يقوم هذا البرنامج بإدخال أرقام صحيحة موجبة ثم يقوم بجمعها وإيجاد معدلها . أن أي رقم سالب في سلسلة الأرقام المدخلة يعني نهاية البرنامج وهذا تتحكم به الجملة الشرطية IF (NUMBER < 0) EXIT

ولاستعراض جملة CYCLE نأخذ المثال التالي :

```
PROGRAM ODD_NUMBERS
  IMPLICIT NONE
  INTEGER :: NUMBER , NUMBER_OF_ODD_NUMBERS
NUMBER_OF_ODD_NUMBERS = 0
DO
   READ *, NUMBER
   PRINT *, "Input data :", NUMBER
   IF (NUMBER < 0 ) THEN
   EXIT
   ELSE IF (MODULO(NUMBER, 2) == 0) THEN
CYCLE
ELSE
   NUMBER_OF_ODD_NUMBERS = NUMBER_OF_ODD_NUMBERS + 1
END IF
END DO
PRINT *, " The number of odd numbers is" , NUMBER_OF_ODD_NUMBERS
END PROGRAM AVERAGE
```

وفي هذا البرنامج نقوم بإدخال أرقام معينة وتستخدم هنا تركيبة IF- THEN الشرطية فعند إدخال أول رقم سالب فسوف تنتهي الحلقة التكرارية من خلال العبارة

```
IF (NUMBER < 0 ) THEN
   EXIT
```

وأذا لم يتحقق هذا الشرط فيتم اختبار الرقم المدخل لمعرفة ما أذا كان هذا الرقم فردياً وتتم عملية الاختبار هذه بواسطة الدالة المكتبية

MODULO(NUMBER, 2) وتكون قيمة الدالة المكتبية هي قيمة باقي قسمة NUMBER / 2

وبالطبع تكون صفراً للأعداد الزوجية ، لاحظ أنه عند انطباق هذا الشرط يقوم البرنامج بتنفيذ جملة CYCLE التي تنقل تركيبة الدوران إلى دورة جديدة ،

أســـــــــــــئلة محلولة :

س 1/ أكتب برنامجاً لإيجاد قيمة Y من العلاقة الآتية : $$Y=\sum_{a=1}^{5}\sum_{x=3}^{12}\frac{X^a\,(X-a)}{a(X+a)}$$

الجواب:

```
PROGRAM VALUE_OF_Y
IMPLICIT NONE
REAL :: A , X , Y
Y = 0.0
DO A = 1., 5.
 DO  X = 3., 12.
 Y = Y + X**A*(X-A) /(A*(X+A))
END DO
END DO
PRINT *, "Y = " ,Y
 END PROGRAM VALUE_OF_Y
```

س 2/ أكتب برنامجاً لقراءة درجات مئة طالب S1, S2, S3, S4, S5, ...,Si, ….,S100

ثم حساب :

أ- الدرجة المتوسطة $AVG = (\sum_{i=1}^{100} Si)/100$

ب- الفرق بين درجة كل طالب والدرجة المتوسطة ، $Di = Si - AVG$

$i = 1, 2, 3, 4, \ldots, 100$

الجواب:

```
PROGRAM DEGREES_OF_STUDENTS
IMPLICIT NONE
REAL :: SUM = 0.0 , S(100) , DEGREE(100) ,  AVERAGE
INTEGER :: I
DO I = 1, 100
  READ *, S(I)
  SUM = SUM + S(I)
END DO
 AVERAGE = SUM / 100.
PRINT  "(T5,A, F8.3)", "AVERAGE =", AVERAGE
 DO  I = 1, 100
DEGREE(I) = S(I) – AVERAGE
   PRINT *,  DEGREE(I)
 END DO
END PROGRAM DEGREES_OF_STUDENTS
```

ملاحظـــــــــــة : من الأخطاء الشائعة التي يقع فيها بعض الطلبة في البرنامج السابق وضع العبارة التي تعطى SUM قيمته الابتدائية داخل حلقة التكرار DO ومثال على ذلك:

```
DO I = 1, 100
READ *, S(I)
SUM = 0.0
SUM = SUM + S(I)
END DO
```

وهذا خطأ لأنه لا يؤدي إلى جمع قيم (S(I كلها بإضافتها واحدة بعد الأخرى إلى SUM حيث أنه بعد إضافة أي واحدة من قيم (S(I إلى SUM وقبل إضافة (S(I التي تليها نعيد قيمة SUM إلى قيمتها الابتدائية SUM = 0.0 مرة أخرى حين نزيد قيمة I ونبدأ في تنفيذ الخطوات تحت عبارة DO مرة جديد ولذلك فيجب أن توضع العبارة SUM = 0.0 قبل عبارة DO حتى لا تنفذ هذه العبارة ألا مرة واحدة فقط.

HOME WORK :

س1 / أوجد قيمة J عند تنفيذ البرنامج الآتي:
(وضح ذلك على مخطط للأعمدة)

```
PROGRAM VALUE_OF_J
IMPLICIT NONE
INTEGER :: M = 2 , N = 3 , J = 0 , I, L , K
DO  I = 1, 2
 DO  K = 2, 3
M = M + 3
END DO
 DO  L = 1, 3
  N = N + 4
END DO
 J = J + M + N
END DO
 PRINT   " (5X,A, I6) " , "J=", J
 END  PROGRAM VALUE_OF_J
```

س2 / ما هي النتيجة المتوقعة للبرنامج التالي ، تتبع خطوات إيجادها بجدول.

```
PROGRAM VALUE_OF_SUM
IMPLICIT NONE
REAL :: SUM = 0.0
INTEGER :: I, J
DO  I = 1, 5
DO  J = 1, I
 SUM = SUM + 1.5
END DO
END DO
PRINT "(2X, A , F8.2)", "SUM=", SUM
END PROGRAM VALUE_OF_SUM
```

المتواليات (المتسلسلات) Series

المتوالية : هي عبارة عن سلسلة من الحدود المتوالية والتي أما أن تزداد زيادة منتظمة أو تتناقص تناقص منتظم .

مثال ذلك :

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{30}$$

أشارات متشابهة

الحد (١) الحد (٢) الحد (٣) الحد (٤)

$$S = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

أشارات مختلفة

والمتوالية : تكون على نوعين هما :

١ - المتوالية المحددة : وهي المتوالية التي تكون منتهية بحد معين . ومثال ذلك :

$$Y = \frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \frac{1}{8} + \dots + \frac{1}{20}$$

ملاحظات :

A) نتعرف على المتوالية بأنها محددة أما من خلال شكل المتوالية أو من خلال منطوق السؤال بأن يذكر عبارة (لغاية 10 حدود أو لغاية 20 حد وهكذا) .

B) يفضل استخدام عبارة DO المحددة في المتواليات المحددة .

٢ - المتوالية غير المحددة : وهي تلك المتوالية التي تكون منتهية بعدة نقاط تدل على أنها غير منتهية الحدود .

ومثال ذلك : $S = 1 + \frac{1}{3} - \frac{1}{5} + \frac{1}{7} - \frac{1}{9} + \dots$

ملاحظات :

A) نتعرف على المتوالية بأنها غير محددة أما من خلال شكل المتوالية أو من خلال منطوق السؤال بأن يذكر عبارة (جد قيمة المتوالية لجميع الحدود التي تزيد فيها قيمة الحد عن $5*10^{-4}$ أو عبارة جد قيمة المتوالية أدناه لغاية قيمة الحد أقل من $5*10^{-4}$) .

B) تحل المتوالية غير المحددة بأستخدام عبارة DO غير المحددة (الدوران المفتوح) .

أســـــــــــــئلة محلولة على المتواليات المحددة :

س1 / أكتب برنامجاً لحساب قيمة SUM من مجموع حدود المتسلسلة التالية :

$$SUM = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \ldots + \frac{1}{30}$$

الحل :

```
PROGRAM VALUE_OF_SUM
IMPLICIT NONE
REAL :: SUM = 0.0
INTEGER :: I
DO I = 1, 30
  SUM = SUM + 1 / I
END DO
PRINT *, 'SUM =' , SUM
END PROGRAM VALUE_OF_SUM
```

س2 / أكتب برنامجاً لإيجاد قيمة SUM من المتوالية أدناه و لغاية ٢٥ حداً ؟

$$SUM = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots$$

الحل :

```
PROGRAM VALUE_OF_SUM
IMPLICIT NONE
REAL :: SUM = 0.0
INTEGER :: I
DO  I = 1, 25
SUM = SUM + (-1)**(I+1) *  1. / (2* I – 1)
END DO
PRINT *, 'SUM =' , SUM
END PROGRAM VALUE_OF_SUM
```

س3 / أكتب برنامجاً لإيجاد قيمة p التقريبية من المتوالية أدناه و لغاية ٢٠ حداً ؟

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \ldots$$

الحل :

```
PROGRAM VALUE_OF_APPROXIMATE_PI
IMPLICIT NONE
REAL :: SUM = 0.0 , PI
INTEGER :: I
DO  I = 1, 20
SUM = SUM + 1. / I**2
END DO
PI = SQRT (6*SUM)
PRINT *, 'PI=' , PI
END PROGRAM VALUE_OF_ APPROXIMATE_PI
```

س4 / أكتب برنامجاً لإيجاد قيمة المتوالية Z أدناه و لغاية ١٠ حدود ؟

$$Z = 1 - \frac{1}{3!} + \frac{1}{5!} - \frac{1}{7!} + \frac{1}{9!} - \dots$$

الحل :

```
PROGRAM VALUE_OF_Z
IMPLICIT NONE
REAL :: Z = 0.0
INTEGER :: I , J , FACT
DO  I = 1, 10
FACT = 1
DO  J = 1, 2*I-1
  FACT = FACT * J
END DO
Z = Z + (-1)**(I+1) *  1. / FACT
END DO
PRINT *, 'Z=' , Z
END PROGRAM VALUE_OF_Z
```

س5 / أكتب برنامجاً يوجد قيمة Y من المتوالية أدناه و لغاية ١٠ حدود ؟

$$4 + Y^2 = 2X + \frac{4X}{45} - \frac{6X}{42} + \frac{8X}{39} - \frac{10X}{36} + \dots$$

الحل:

```
PROGRAM VALUE_OF_Y
IMPLICIT NONE
REAL :: X , Y , SUM
INTEGER :: I
READ *, X
SUM = 2 * X
DO  I = 1, 9
SUM = SUM + (2*I+2) * X / (48 – 3*I)
END DO
Y = SQRT (SUM - 4)
PRINT *, 'Y=' , Y
END PROGRAM VALUE_OF_Y
```

HOME WORK : أكتب برنامجاً لأيجاد قيمة Y بموجب الشروط التالية :

أ – في حالة $-2 \le x < 4$ يستخدم المتوالية أدناه لعشرة حدود $\frac{Y}{3} = X - \frac{X^2}{3!} + \frac{X^4}{5!} - \frac{X^6}{7!} + \frac{X^8}{9!} - \dots$

ب - للحالات الأخرى $Y^2 = |X^2 - 8X|$

يقوم البرنامج بطباعة قيمة X في السطر الثاني والعمود الرابع وقيمة Y في السطر الثالث والعمود السادس وبعناوين وصيغ مناسبة .

أســـــــــــــئلة محلولة على المتواليات غير المحددة :

س1 / أكتب برنامجاً لإيجاد قيمة SUM لغاية قيمة الحد أقل من 0.01 ثم أطبع عدد تلك الحدود ؟

$$SUM = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots$$

الحل :

```
PROGRAM VALUE_OF_SUM
IMPLICIT NONE
REAL :: SUM = 0.0 , TERM
INTEGER :: I = 1
DO
TERM =-1**(I+1)*( 1. / (2*I-1))
IF ( ABS(TERM) < 1.E-2) EXIT
SUM = SUM +  TERM
I = I + 1
 END DO
  PRINT * , "SUM =", SUM , "  NO. OF TERMS = ", I-1
 END PROGRAM VALUE_OF_SUM
```

س2 / أكتب برنامجاً لإيجاد قيمة Z من المتوالية أدناه لغاية قيمة الحد أقل من 10^{-4} ؟

$$Z = 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \ldots$$

الحل :

```
PROGRAM VALUE_OF_Z
IMPLICIT NONE
REAL :: Z = 0.0 , TERM
INTEGER :: I = 1 , J , FACT
DO
 FACT = 1
DO  J = 1, I
 FACT = FACT * J
END DO
TERM =1. / FACT
IF ( ABS(TERM) < 1.E-4) EXIT
Z = Z +  TERM
I = I + 1
  END DO
 PRINT * , "Z = ", Z
 END PROGRAM VALUE_OF_Z
```

س3 / أكتب برنامجاً لأيجاد قيمة p التقريبية من المتوالية أدناه لجميع الحدود التي تزيد فيها قيمة الحد عن $1*10^{-5}$ وطبع عدد تلك الحدود ؟

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \ldots.$$

الحل :

```
PROGRAM VALUE_OF_APPROXIMATE_PI
IMPLICIT NONE
REAL :: SUM = 0.0 , TERM ,  PI
INTEGER :: I = 1
DO
  TERM =1. / I**2
IF (ABS(TERM) <= 1.E-5) EXIT
SUM = SUM +  TERM
I = I + 1
 END DO
20  PI = SQRT (6*SUM)
PRINT * , "PI =", PI , "  NO. OF TERMS = ", I-1
END PROGRAM VALUE_OF_APPROXIMATE_PI
```

س4 / أكتب برنامجاً يسأل عن قيمة X ثم يقوم بحساب وطبع قيمة Y لجميع الحدود التي تزيد القيمة المطلقة للحد عن $1*10^{-4}$ ؟

$$Y = \frac{\pi}{2} - \frac{3\sin X}{2!} + \frac{5\sin X}{4!} - \frac{7\sin X}{6!} + \frac{9\sin X}{8!} - \ldots$$

الحل :

```
PROGRAM VALUE_OF_Y
IMPLICIT NONE
REAL :: X , Y , FACT , TERM
INTEGER :: I = 1 , J
READ *, X
Y = 3.14159 / 2.
DO
   FACT = 1.0
DO  J = 1, 2*I
  FACT = FACT * J
END DO
TERM =-1**I*(2*I+1)*SIN(X) / FACT
IF ( ABS(TERM) <= 1.E-4) EXIT
Y = Y +  TERM
I = I + 1
    END DO
 PRINT * , "Y = " , Y
 END PROGRAM VALUE_OF_Y
```