جمل التحكم والشرط والتكرار

PROGRAM CONTROL, CONDITION AND LOOP STATEMENTS

1- الجمل الشرطية :-

تقسم جمل الشرط الى نوعين وهي :-

1- جملة اذا الشرطية (if – statement)

2- جملة التوزيع (switch – case)

1- جملة اذا الشرطية (if – statement) :-

وتستخدم للتحكم بطريقة سير البرنامج اعتمادا على تحقق الشرط **(condition)** ، ولها ثلاث صيغ :-

1- تستخدم if لوحدها في حالة استخدام شرط واحد وبالشكل التالي :-

```
if  condition
statement
end
ex:-
x=input('enter the number =')
if  x  > 0
disp(' x is positive number ')
end
```

2- الجملة (if / else) :- يستخدم الامر else كجزء من جملة الشرط if ، فاذا تحقق شرط جملة if يترتب عليه تحقيق النتائج التالية لجملة if (الى ما قبل جملة else مباشرة) والا تحقق النتائج ما بعد else ، ويأخذ الصورة التالية :-

```
if condition
statement(s)
else
statement(s)
end
ex:-
x=input('enter the number =')
if  x  > 0
disp(' x is positive number ')
else
disp('x is negative number ')
end
```

3- الجملة (if / elseif) :- الغرض من هذه الصورة من صور جملة الشرط if هو اختبار مدى تحقق اكثر من شرط ويأخذ الصيغة التالية :-

```
if condition 1
statement 1
elseif condition 2
statement 2
elseif condition (n-1)
statement (n-1)
else
statement (n)
end
```

مثال / اكتب برنامج بلغة MATLAB لإدخال عدد ، يبين اذا كان العدد موجب ام سالب او يساوي صفر ؟

```
x = input ('enter the number =')
if   x > 0
disp (' this number is positive ')
elseif  x < 0
disp ('this number is negative ')
else
disp('this number is equal 0 ')
end
```

ملاحظة :- يجب ان تحتوي الجمل الشرطية على احد ادوات المقارنة او الادوات المنطقية (Logical or Relational Operators) الموضحة في الجدول التالي :-

Logic Operators المعاملات المنطقية	Operation الوظيفة
<	اقل من
<=	اقل من او يساوي
>	اكبر من
>=	اكبر من او يساوي
==	يساوي
~=	لا يساوي
&	And

Short-circuit And للقيم العددية فقط (scalars)	&&
Or	|
Short-circuit or للقيم العددية فقط (scalars)	||
Not	~

2- جملة التوزيع (switch – case)

تستخدم للتحكم بطريقة سير البرنامج اعتمادا على قيمة المتغير المعطى لها والصيغة العامة لها :-

```
switch  variable
case value 1
Statement 1
case value2
Statement 2
case value n
Statement n
otherwise
Statement
end
```

يبدأ البرنامج اولا بقراءة قيمة المتغير (variable) ثم يبدأ بمقارنتها مع القيم (value) بعد عبارة case اذا كانت قيمة المتغير variable تنطبق مع اي من القيم value يتم تنفيذ الجملة statement بعدها وهكذا ، واذا لا تنطبق اي من القيم value مع قيمة المتغير variable سوف ينتقل الى العبارة otherwise ويتم تنفيذ الجملة بعدها .

مثال / اكتب برنامج بلغة MATLAB لحساب قيمة الاطوال بوحدات السنتمتر .

```
clc ; clear ; close all ;
disp (' convert L to centimeters ');
L=input (' Enter the length :')
units= input (' Enter the unit of L :','s');
switch units
        case {'inch','in'}
y=L*2.54;
        case {'feet','ft'}
y=L*2.54*12;
        case {'meter','m'}
y=L*100;
```

```
        case {'millimeter','mm'}
y=L/10;
otherwise
        disp('unknown unit');y=nan
end
display( [ num2str(y) ,' cm ' ] ) ;
```

```
>>file_name
convert L to centemeters
Enter the length : 5
Enter the unit of L : m
500 cm
```

تمرين للطالب نفذ البرنامج لأطوال ووحدات اخرى

3- حلقات التكرار (loops statements)

وهي مجموعة من الجمل تستخدم لتكرار تنفيذ مجموعة من الاوامر لعدد محدد من المرات ، يحتوي برنامج MATLAB على نوعين من حلقات التكرار هما :-

1- حلقات for (for … loops)

2- حلقات while (while … condition)

يستخدم برنامج MATLAB اوامر للتحكم في عملية التكرار هما :-

1- امر التوقف break او return

2- امر الاستمرار continue .

1- حلقات for :-

وتستخدم لتنفيذ مجموعة من الاوامر مرات متعددة ، ويتم التحكم بعدد مرات تنفيذ الاوامر باستخدام عداد تحدد له قيمتي البداية والنهاية وكذلك مقدار الزيادة ، والصيغة العامة لها :-

```
for  i = n : k : m
statement(s)
end
```

حيث ان :-

i :- قيمة العداد .

n :- قيمة البداية للعداد .

k :- مقدار الزيادة للعداد .

m :- قيمة نهاية العداد .

المثال التالي يجمع الاعداد من 1 الى 10

Ex:-

```
S=0 ;
for i = 1 : 10
s=s+i ;
end
disp (s)
```

☒ ملاحظة :- تهمل مقدار الزيادة k اذا كانت مقدار زيادة العداد هي 1 ، لان القيمة الافتراضية لبرنامج MATLAB هي 1 .

مثال1 / اكتب برنامج بلغة MATLABلحساب المعادلات التالية: -

1) $S = \sum_{n=1}^{5000} n$

```
s= 0 ;
for  n = 1 : 5000
s = s + n ;
end
disp ( s)
```

sol.

```
>>file_name
12502500
```

2) $S = \sum_{n=1}^{5000} n^2$

```
s = 0 ;
for  n  =  1  :  5000
s  =  s  +  n  ^  2  ;
end
disp ( s )
```

sol.

```
>>file_name
41.679
```

3) $S = \sum_{n=1}^{5000} \frac{1}{n^2}$

```
s = 0 ;
for n = 1 : 5000
s = s + 1 / n^2 ;
end
disp ( s )
```

sol.

```
>>file_name
1.6447
```

مثال2 / اكتب برنامج بلغة MATLAB لإيجاد مفكوك العدد n! .

```
% Program Calculate Factorial Number
n = input ( ' the number of factorial = ' ) ;
fact = 1 ;
for i = 1 : n
fact = fact * i ;
disp ( [ i   fact  ])
end
>>file_name
  the number of factorial = 8
   1     1
   2     2
   3     6
   4    24
   5   120
   6   720
   7  5040
   8 40320
```

2- حلقات while (while … condition)

تستخدم لتنفيذ امر او مجموعة الاوامر مرات متعددة مادام شرط ما متحقق والصيغة العامة لها :-

```
while  condition
statement(s)
end
```

مثال / البرنامج التالي يجمع الاعداد من 1 الى 10 وبزيادة مقدارها 0.5 (مجموع الاعداد 1 , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 . . . 10) باستخدام while .

```
s = 0 ;
i = 1 ;
while  i  <=  10
s = s + i ;
i = i +  0.5 ;
end
display ( s )
```

sol.

```
>>file_name
s =
104.500
```

تمارين

س1) اكتب برنامج بلغة MATLAB لحساب المتسلسلات التالية :-

1- $1^2 + 2^2 + 3^2 + \ldots + 1000^2$

2- $1 - 1/3 + 1/5 - 1/7 + 1/9 - \ldots.$

3- $1/(1^2 \cdot 3^2) + 1/(3^2 \cdot 5^2) + 1/(5^2 \cdot 7^2) + \ldots$

1) $1^2 + 2^2 + 3^2 + \ldots + 1000^2$

```
s = 0 ;
for  k = 1 : 1000
s = s + k ^ 2 ;
end
display ( s )
```

نتائج تنفيذ البرنامج

```
>>file_name
s = 338350
```

2) $1 - 1/3 + 1/5 - 1/7 + 1/9 - \ldots.$

```
s1 = -1 ;
s2 = 0 ;
n = input ( ' input the number = ')
for  m = 1 : 2 : n
s1 = s1 * -1 ;
s2 = s2 + s1 / m ;
end
display ( s2 )
```

نتائج تنفيذ البرنامج

```
>>file_name
```

input the number = 4

s2 = 0.6667

3) $1/(1^2 . 3^2) + 1/(3^2 . 5^2) + 1/(5^2 . 7^2) + \ ...$

```
s = 0 ;
m = input ( ' input the number = ') ;
for i = 3 : 2 : m
s = s + 1 /  ( ( i – 2 ) ^ 2 * i ^ 2 ) ;
end
disp ( [ m   s ] )
```

نتائج تنفيذ البرنامج

```
>>file_name
input the number = 4
4.0000   0.1111
```

Q2) a) Write MATLAB program to calculate x ,

$$a\,x^2 + b\,x + c = 0$$

$$when\ a = 0\ , x = -\frac{c}{b}$$

Use the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ when $a \neq 0$

and a = 2 , b = -10 , c = 12

<u>sol.</u>

```
% This M-file solves the quadratic equation using the quadratic formula
a = 2 ;
b = -10 ;
c = 12 ;
% Different cases for a = 0 and otherwise :
if  a == 0    %  a x^2 + b x + c = 0
x = - c / b   %   b x + c = 0
```

else

x(1) = (- b + sqrt (b ^ 2 – 4 * a * c)) / (2 * a) ;

x(2) = (- b - sqrt (b ^ 2 – 4 * a * c)) / (2 * a) ;

end

display (x)

>> file_name

x =

    3   2

Q2)b) The steady-state current I flowing in a circuit that contains a resistance R=5,capacitance C =10, and inductance L=4 in series is given by

$$I = \frac{E}{\sqrt{R^2 + (2\pi\omega L - \frac{1}{2\pi\omega C})^2}}$$

where E =2 and ω=2 are the input voltage and angular frequency respectively. Compute the value of I. (Answer: 0.0396)

```
R = 5;
C = 10;
L = 4;
E = 2;
w = 2;
I = E / sqrt(R ^ 2 + (2 * pi * w * L - 1 / (2 * pi * w * C)) ^ 2)
```

4) جمل القفز Jamping statements

1) الخروج من جمل التكرار Breaking out of loops

ان تكرار while يستمر في العمل حتى يصبح الشرط صحيحا او خاطئا ، وان تكرار for يستمر في العمل حتى ينتهي عدد المرات التي تحددها له ، ولكن قد تحتاج في بعض الاحيان الى الخروج المبكر من التكرار قبل انتهائه ، وفي هذه الحالة يمكننا استخدام الامر **break** او **return** ، ويستخدم بالشكل التالي :-

```
for / while
statement(s)
if condition
break
end
…
end
```

ex:-

```
clc ; clear ; close all ;
for m = 1 : 10
if ( m ^2 ) >= 36
break ;
end
display ( m ) ;
end
```

sol.

```
>>file_name
m =
1
m =
2
m =
3
m =
4
m =
5
```

مثال / اطبع مربع الاعداد من 1 – 10 لغاية القيمة 36

وفي حالة استخدام while loop

```
clc ; clear ; close all ;
m = 1 ;
while m < 10
 if ( m ^ 2 ) >= 36
break ;
end
display ( m ) ;
m = m + 1 ;
end
```

2) جملة continue

يستخدم الامر continue في داخل الحلقة التكرارية for loop او الحلقة التكرارية المشروطة while loop لانهاء التكرار الحالي ، وتجاوز تنفيذ باقي الاوامر (الجمل البرمجية) الموجودة بداية من الامر continue ووصولا لنهاية الحلقة المستخدمة باستخدام جملة end ، ويستخدم بالشكل التالي :-

```
for / while
statement(s)
if condition
continue
end
end
```

مثال/ اطبع القيم الزوجية من 1 – 10

ex:-

```
clc ; clear ; close all ;
for n = 1 : 10
if rem ( n , 2 ) ~= 0
continue ;
end
display ( n ) ;
end
```

sol.

```
>>file_name
 n =
2
n =
4
n =
6
n=
8
n =
10
```

5) صناديق try – catch

يستخدم برنامج MATLAB طريقة الاستثناء exception في معالجة الاخطاء التي من الممكن ان يقع فيها المستخدم اثناء سير البرنامج باستخدام صناديق try – catch والتي تستخدم الصورة التالية :-

```
try
commands1
catch
commands2
end
```

في هذه الصورة يتم وضع جميع الاوامر commands1 التي يوجد احتمال حدوث خطأ فيها في جملة try فاذا لم يحدث خطأ ينتقل مباشرة الى جملة end (ينفذ الاوامر commands1 بصورة طبيعية) ، ولكن اذا حدث خطأ في تنفيذ الاوامر commands1 فيتم الانتقال الى جملة catch لتنفيذ الاوامر commands2 والتي تكتب من قبل المستخدم كرسالة توضح وجود خطأ ليتجنب حدوث هذا الخطأ في المرات القادمة .

Ex:-

```
clc ; clear ; close all ;
a = input ('Enter first matrix :');
b = input (Enter second matrix :');
try
c = a * b ;
catch
c = NaN ;
disp ( ' wrong dimensions for matrices multiply ');
end
display ( c );
```

sol.

```
>>file_name
Enter first matrix : [ 1 3 ; 2 4 ]
Enter second matrix : [ 5 7 ; 6 8 ]
c =
23  31
34  46
```

نلاحظ ان البرنامج ينتج عملية ضرب المصفوفتين بدون عرض اخطاء وذلك لتحقق شرط ضرب هاتين المصفوفتين وهو تساوي عدد اعمدة المصفوفة الاولى a مع عدد صفوف المصفوفة الثانية b .

نفذ البرنامج مع ادخال عناصر المصفوفتين بالقيم التالية :-

```
>>file_name
```

```
Enter first matrix : 2 * ones ( 2 , 4 )
Enter second matrix : magic ( 3 )
wrong dimentions for matrices multiply
c =
NaN
```

لاحظ ان البرنامج نفذ الاوامر التي تتضمنها جملة catch مما يدل على حدوث خطأ في الاوامر التي تتضمنها جملة try .