الكورس الثاني

هناك الكثير من الدوال الخاصة في لغة (MATLAB) وسوف نوجز منها مايهمنا في اطار دراستنا لهذه اللغة.

1-(' ')disp:يقوم هذا الايعاز بطبع الناتج بدون اسم.

(Ex1)
```
>> A=5
   A=
      5
>> disp (A)
      5
```

(Ex2)
```
>> B=[1 2 3 4 5]
B=
    1 2 3 4 5
>> disp (B)
  1 2 3 4 5
```

(Ex3)
```
>>C=[1 2 3; 4 5 6; 7 8 9]
C=
1   2   3
4   5   6
7   8   9
>> disp (C)
1   2   3
4   5   6
7   8   9
```

2-(' ') Fprintf : يقوم هذا الايعاز بنفس الايعاز السابق اعلاه اضافة الى قيامه بطبع اي تعبير, ويعرف ايضا بايعاز الطباعة.

```
(Ex1) >> fprintf('Matrix=M')
            Matrix=M
(Ex2) >> Fprintf('this is matlab course')
                  'this is matlab course'
```

3-Pi: يعطي هذا الايعاز قيمة النسبة الثابتة π

(Ex)
```
 >>pi
ans=
3.1416
```

4-format (long or short) : يقوم هذا الايعاز باخذ التقريب المناسب للعدد وبصورة مختصرة اوبصورة مطولة.

(Ex1) format short
```
 >>pi
ans=
3.1416
```

(Ex1) format long
```
>>pi
ans=
3.14159265358979
```

5-( )abs: تحسب هذه الدالة القيمة المطلقة لاي قيمة.

(Ex1)
>>abs(-4)
ans=
4

(Ex2)
>>abs=([-2 -55 -1 -3.4])
ans=
2.0000 55.0000 1.0000 3.40000

(Ex3) >>abs=([-0.1 -0.2;-0. 3 -0.4; -0.5 -0.6])
ans=
0.1000 0.2000
0.3000 0.4000
0.5000 0.6000

6-( )conj: تقوم هذه الدالة بحساب العدد المرافق للعدد المركب.

(Ex1)
>>conj (3+9i)
ans=
3.0000-9.0000i

(Ex2)
>>conj =([3+9i 2-3i;1+7i 5-i])
ans=
$3.0000-9.0000i$ $2.0000-3.0000i$
$1.0000-7.0000i$ $5.0000+1.0000i$

7-( )max
: يقوم هذا الايعاز بحساب اكبرعدد في المتجه الصفي او العمودي.

8-( )min :يقوم هذا الايعاز بحساب االصغرعدد في المتجه الصفي او العمودي.

(Ex)
>>a= [1 12 3 9 2 17]
ans=
1 12 3 9 2 17
>> max(a)
ans=
17
>> min(a)
ans=
1

9-( )mean: يقوم هذا الايعاز بحساب معدل الارقام في المتجه الصفي او العمودي.

(Ex)
>>mean= (a)
ans=
7.3333

10-( )sort:يقوم هذا الايعازبترتيب عناصر المتجه الصفي او العمودي تصاعديا.

(Ex)

```
>>sort= (a)
   ans=
    1  2  3  9  12  17
```

11-( )sum:تقوم هذه الدالة بحساب مجموع عناصر المتجه الصفي او العمودي.

(Ex)

```
>>sum= (a)
   ans=
44
```

12-( )prod: تقوم هذه الدالة بحساب حاصل ضرب عناصر المتجه الصفي او العمودي.

(Ex)

```
>>prod= (a)
    ans=
11016
```

13-( )factor :تقوم هذه الدالة بتحليل اي عدد الى عوامله الاولية.

(Ex)

```
>> factor = (164)
ans= 2  2  41
```

14-primes(A,B): عند اختيارنا لعدد ما فان هذه الدالة تقوم بتعويض الاعداد الاولية التي تكون اقل اويساوي من هذا العدد المختار.

(Ex)

```
>> primes = (9)
ans=
2  3  5  7
```

(Ex)

```
>> primes = (13)
ans=
2  3  5  7 11  13
```

15-gcd(A,B): تقوم هذه الدالة بايجاد العامل المشترك الاكبر بين العددين A,B.

(Ex)

```
>> gcd = (8,12)
    ans=
       4
```

(Ex)

```
>> gcd= (24,18)
ans=
       6
```

16-lcm(A,B): تقوم هذه الدالة بحساب المضاعف المشترك الاصغر للعددين A,B.

(Ex)

```
>> lcm = (4,2)
    ans=
       4
```

(Ex)

```
 >> lcm = (12,18)
ans=
       36
```

17-( )rat: تقوم هذه الدالة بتحويل الكسر العشري الى كسر طبيعي.

(Ex1)
```
>> rat = (1.2)
    ans=
      1+1/(5)
```
(Ex2)
```
>> rat = (1.5)
    ans=
      2+1/(-2)
```
(Ex3)
```
>> rat = (pi)
ans=
 3+1/(7+1/ (16))
```

18-nchoosek (n, k):

(Ex1)
```
>> nchoosek = (8,4)
ans=
     70
```
(Ex2)
```
>> nchoosek = (6,2)
ans=
     15
```
(Ex3)
```
>> nchoosek  (2,4)
???Error using==>
nchoosek   k must be
Integer between 0 and N
```

19-Syms: وهو ايعاز يستعمل لتعريف المتغيرات في لغة (matlab).

(Ex1)
```
 >> Syms x
>>
```
(Ex2)
```
>> Syms x  y  z  w
>>
```
(Ex3)
```
 >> Syms ('x'), Syms ('y'), Syms ('z')
  >>
```

20-Diff: يقوم هذا الايعازبحساب المشتقة للدوال.

(Ex1)
```
 >> Syms x
>>f=x^3-3*x^2+10*x-22
>>diff(f)
ans =
3*x^2-6*x+10
```

(Ex2)

```
>> Syms x
>>A=[2*x,x^2;x+1,x-1]
>>diff(A)
ans =
2    2*x
1     1
```

(Ex3)

```
>> Syms x
>>a=[2*x,x^2;x+1,x-1]
>>diff(A)
ans =
        2      2*x
      2*x      1
>>diff(a,2)
ans =
     0    2
     2    0
>>diff(a,3)
ans =
     0    0
     0    0
```

21-( )int : يقوم هذا الايعاز بحساب التكامل لاي دالة.

(Ex1)

```
>> Syms t
>>g= sin(t)
>>int(g)
ans =
-cos(t)
```

(Ex2)

```
>> Syms x
>>y=[x^2,1/(1+x);1-x,1]
>>int(y)
ans=
[x-1/2*x^2,x]
```

22-(, ,)int :ايجاد التكامل المحدد لاي دالة.

(Ex1)

```
>> Syms t
>>h= exp (t)
>>int(h, 0, 1)
ans =
      exp(1)-1
```

(Ex2)

```
>> Syms x
>>w=[exp(x), x; x^2, 2*x]
>>int(w , 1 ,2)
ans=
exp(2)-exp(1)    3/2
      7/3          3
```

23-( )limit : ايجاد غاية الدالة مقتربة من الصفر(0→x) .

(Ex1)

```
>>Syms t
>>f= 1/ (1+t);
 >> limit (f)
ans =1
```

(Ex2)

```
>> Syms x
 >>z=[exp(x), x/2 : x, 2/(x^2+1)]
>> limit (z)
```

ans=

$$\begin{matrix} 1 & 0 \\ 0 & 2 \end{matrix}$$

23-(,)limit : ايجاد غاية الدالة مقتربة من النقطة a. (a →x) .

(Ex1)

```
>>Syms t
>>f= 1/ (1+2);
 >> limit (f,2)
ans =
    1/3
```

(Ex2)

```
>> Syms x
 >>w=[exp(x), x/2 : x, 2/(x^2+1)]
>> limit (z,1)
ans=
```

$$\begin{matrix} \exp(1) & 1/2 \\ 1 & 1 \end{matrix}$$

25-(, ,)symsum: ايجاد الجمع المنتهي لمجموعة من الحدود (التكامل العددي Σ).

(Ex1)

```
>>Syms x
>> symsum (x, 1 , 9)
ans =45
```

(Ex2)

```
>> Syms t
 >> symsum (1/t,1 , 6)
        ans=49/20
```

26-( )simplify: تبسيط الدوال الى ابسط صورة

(Ex)

```
>>Syms x
>>f=2*x^2+1/x+12-5*x-4*x^2+3*x-44;
>> simplify (f)
ans =
    -(2*x^3-1+32*x+2*x^2)/x
```

27-( )plot: ايعاز للرسم الاعتيادي للدوال.

(Ex)

```
>> a=[1  2  3]
>>plot(a)
```

28-( )fplot: دالة الرسم بحدود [a, b].

(Ex1)
>> Fplot('exp(x)',[1, 2])
(Ex3)
>>Fplot('cos(2*x)*tan(x)',[1, 2])

(Ex2)
>> Fplot('sin(-2*x)-2',[0,1])
(Ex4)
>>Fplot('cos(2*x)*sin(2*x)',[0, 1])

29-( )subs : دالة تعويض قيمة معينة في الدوال.

(Ex)
>>Syms x
>>g=cos (x);
Subs(g,[1])

30-('Inline('fun:ايعاز يستخدم لتعريف الدوال, وهذا الايعاز ينفذعلى نافذةcommand فقط

(Ex1)
>> a=inline('sin(x)')
>> a(x)=sin(x)

(Ex2)
>>a= inline('sin(x+h)')
a(x+h)=sin(x+h)
ويمكن ان تعتمدالدالة اعلاه على اكثر من متغير.

تحويلات انظمة الاعداد

1-(' ')bin2dec : يقوم هذا الايعاز بتحويل النظام الى النظام العشري.

(Ex1)
```
>> bin2dec ('101')
ans =
    5
```

(Ex1)
```
>> bin2dec ('111001')
ans=
    57
```

2-(' ')dec2bin:يقوم هذا الايعاز بتحويل النظام العشري الى النظام الثنائي.

(Ex1)
```
>> dec2bin ('23')
ans =
10111
```

(Ex1)
```
>> dec2bin (60)
ans=
111100
```

3-(,)dec2bin: يقوم هذا الايعاز بتحويل النظام العشري الى النظام الثنائي وذلك بتحديد عدد البتات المطوبة.

(Ex1)
```
>> dec2bin (23,8)
ans =
00010111
```

(Ex1)
```
>> dec2bin (60,7)
ans=
  0111100
```

4-( )dec2hex: يقوم هذا الايعاز بتحويل النظام العشري الى النظام السداسي عشر.

(Ex1)
```
>> dec2hex (12)
ans =
    C
```

(Ex1)
```
>> dec2hex (42)
ans=
     2A
```

5-(,)dec2hex: يقوم هذا الايعاز بتحويل النظام العشري الى النظام السداسي عشر ايضا ولكن بتحديد عدد البتات المطلوبة.

(Ex)
```
>> dec2hex (42,5)
ans =
   0002A
```

6-(' ')hex2dec : يقوم هذا الايعاز بتحويل النظام السداسي عشر الى النظام العشري.

(Ex1)
```
>> hex2dec ('2F')
ans =
   47
```

(Ex1)
```
>> hex2dec ('AFA')
ans=
     2810
```

كتابة برامج بلغة MATLAB

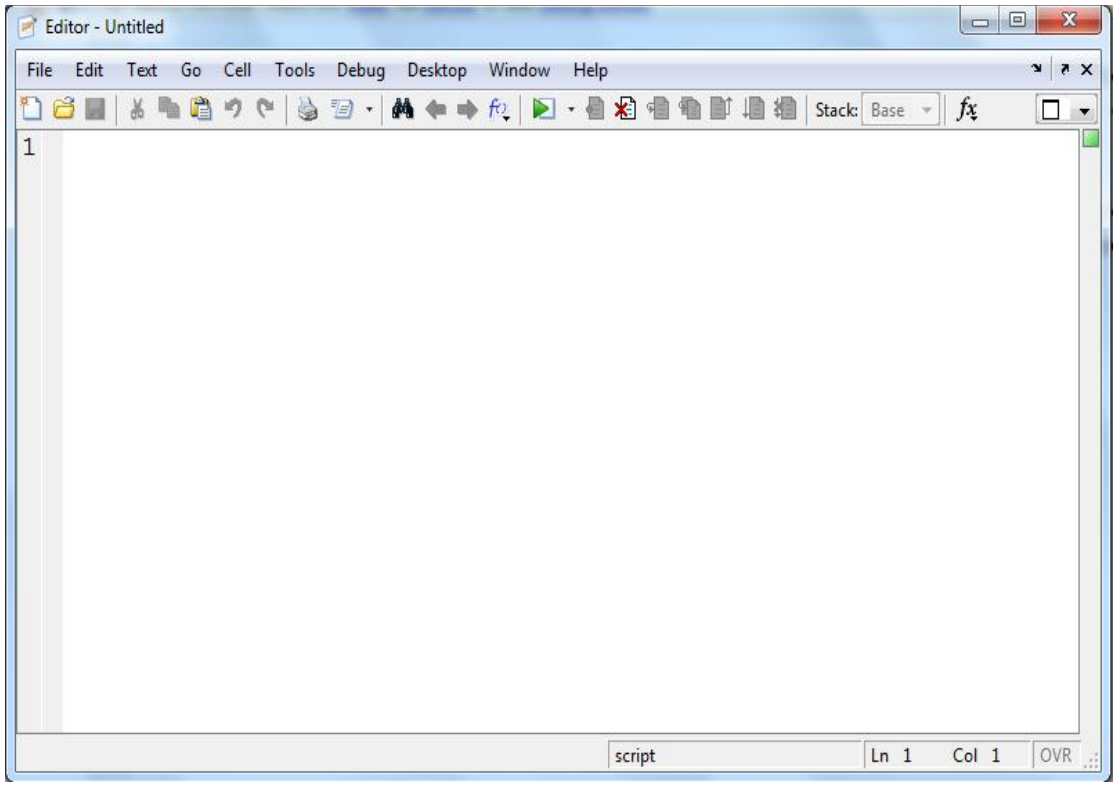
تعمدت ان اعطي الدوال من البداية وحتى الان وتطبيقها في نافذة Command Window حتى نملك من الخبرة الكافية مع الدوال لكي نصبح جاهزين على كتابة برامج بلغةMATLAB.
لكن لم ننشئ برنامجا متكاملا بلغة MATLAB يمكن التعامل معه من حيث التعديل او نقله. من حاسبة الى اخرى, هذا البرنامج لايمكن كتابته في نافذة Command Window بل يكتب في نافذة Editor/Debugger (محرر/ متتبع الاخطاء)

وهذه النافذة عادة لاتكون ظاهرة عند تشغيل برنامج MATLAB بل يمكن الوصول اليها عن طريق:

**من شريط الادوات الرئيسي(Tool Bar) نضغط ضغطة واحدة على اول ايقون والتي تكون على شكل صفحة فارغة اشارة الى ملف جديد.
او

**من شريط القوائم (Menu Bar) نختار قائمة File ومن ثم القائمة الجزئية new وبعد ذلك نختار M-File

حيث تظهر النافذة كما في بالشكل التالي:



بعد ان تظهر هذه النافذة يمكن كتابة البرنامج (عبارة برمجية او اكثر) ومن ثم نخزنه حتى يتم ذلك تنفيذه.

لناخذ مثال ونتعرف على كتابتةالى تنفيذه خطوة بخطوة
مثال/اكتب برنامج يقوم بادخال المصفوفة (A=(1 2 3;4 5 6;7 8 9 وايجاد المدور والمعكوس لها؟

الحل/
في البداية بعد تشغيل برنامج MATLAB نقوم بفتح نافذة Editor/Debugger وذلك الطريقتين السابقتين ولتكن من (File→New→M-File) بعد ذلك خطوات التالية والتي تمثل الحل للمثال:

```
clc
disp('this Program 1')
A=[1  0  3 ; 4  5  6 ; 7  8  9]
T=A'
I=inv(A)
```

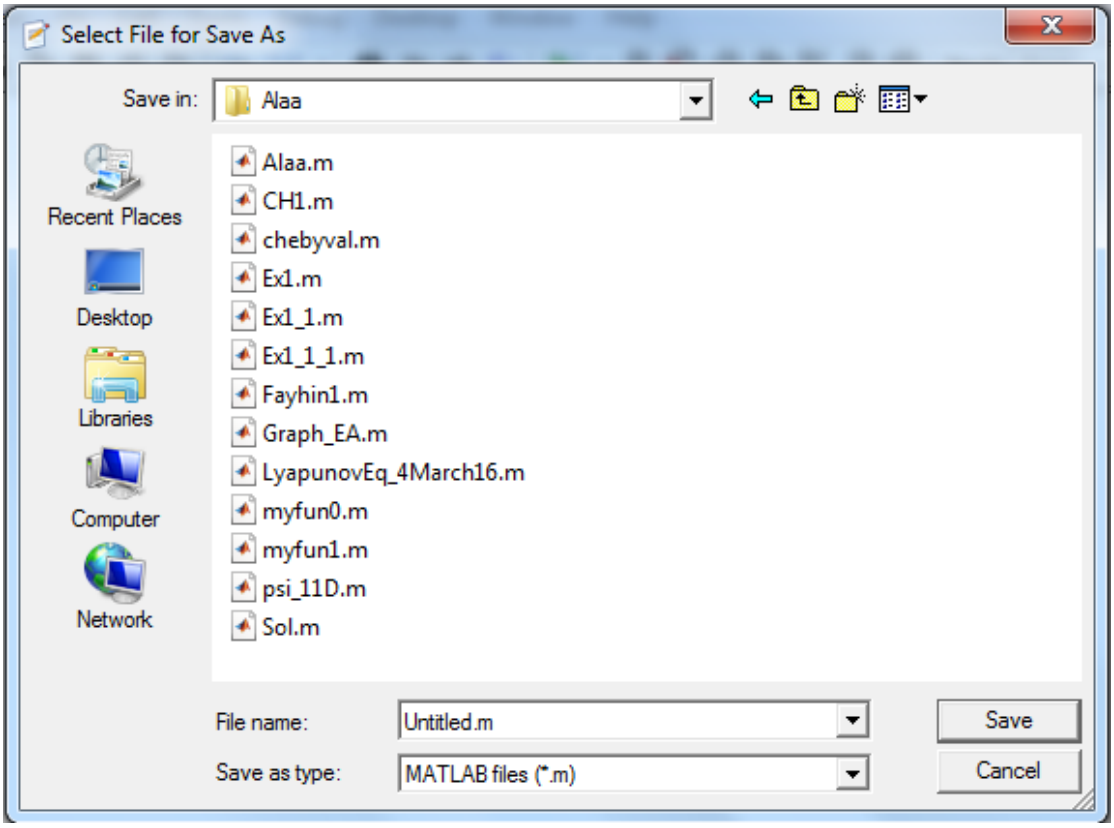
بعد ذلك نقوم بعملية حفظ البرنامج ولها عدة طرق:

** من شريط الادوات لنافذة (محرر او متتبع الاخطاء) نضغط ضغطة واحدة على الايقونة الثالث والتي تكون على شكل القرص المرن (Floppy Disk)

او

** من شريط القوائم (Menu Bar) نختار (File→save as)

سوف يظهر مربع الحوار الخاص بعملية الحفظ التالي:



اهم ماموجود في هذا المربع هو قائمة (save in) في اعلى المربع والذي يخصص فيه المكان الذي يخزن فيه الملف (البرنامج) ويظهر المجلد الافتراضي (Work) والذي بدوره يكون داخل مجلد البرنامج الرئيسي (MATLAB6p5).

والمربع (File name) الخاص بتسمية الملف حيث ينشئ اسم افتراضي (Untitled) واذا رغبت بتسمية الملف فعليك اتباع القواعد التالية:

1. يجب ان يبدأ اسم الملف بحرف ومن ثم يمكن ان يتبع بحرف او رقم.
2. تجنب الفراغات في اسم الملف.

يمكن استخدام اسم يحوي على 63رمزا وبعد ذلك يهمل اي رمز زائد عن 63.

بعد خزن البرنامج تاتي عملية التنفيذ (Run), وايضا تتم بعدة طرق:

**الضغط على F5 في لوحة المفاتيح (Keyboard).

او

** الضغط على الايقون الموجودة في شريط الادوات.

او

** من شريط القوائم يمكن اتباع (Debug→Run)

اين تظهر نتائج البرنامج بعد عملية التنفيذ؟

في نافذة Command Window

وفي مثالنا نجد النتائج بالشكل التالي:

This Program 1

A=

$$\begin{matrix} 1 & 0 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

T=

$$\begin{matrix} 1 & 4 & 7 \\ 0 & 5 & 8 \\ 3 & 6 & 9 \end{matrix}$$

I=

$$\begin{matrix} 0.2500 & -2.0000 & 1.2500 \\ -0.5000 & 1.0000 & -0.5000 \\ 0.25 & 0.6667 & -0.4167 \end{matrix}$$

ايعازات الرسم

1-Plot(x) :يقوم هذا الايعاز برسم المصفوفة او المتجه x.

(Ex)

```
>> x=[1 , 2 , 3]
>> Plot(x)
```

2-Plot(x,y): يرسم هذا الايعاز المصفوفة x بالنسبة للمصفوفة y.

(Ex)

```
>> x=[1 , 2 , 3]; y=[4 , 5 , 6]
>> Plot(x,y)
```

3-Plot(x,y,'from'): يقوم هذا الايعاز بنفس وضعية الايعاز السابق ولكن باضافة اشكال مختلفة.

ملاحظة ان هذه الاشكال والالوان يمكن وضعها داخل اكثر من ايعاز للرسم.

Form: y-yellow, m=magenta, c=cyan, r=red, g=green, b=blue, w=white, k=black, +=plus, o=circle, x=xmark, *=star, _=solid, .=point, s=square, d=diamond, v=trangle(down), ^=triagle(up), :=dotted, -=dashdot, --=dashed.

(Ex)

```
>> a=[1 , 2 , 3 ; 4 , 5 , 6]; b= [2 , 3  4 ; 6 , 7 , 8]
>> plot(a,b, 'r*')
```

يكون الرسم في هذا المثال على شكل نجوم حمراء.

4-polar(THETA,R): يقوم هذا الايعاز بالرسم بالنظام القطبي.

(Ex1)

```
>> Polar (pi ,5)
```

(Ex2)

```
>> a=[1 , 2 , 3]; th= [pi , pi/2 , pi/4]
>>Polar(th,a)
```

5-grid: يقوم هذا الايعاز بوضع شبكة في الرسم.

(Ex)

```
>> a=[1 , 2 , 3];
>> polar(a) , grid
```

6-Text(x,y, 'name'): يتم في هذا الايعاز وضع اسم في موقع محدد.

(Ex)

```
>> text(0.4 , 0.6 , 'AHMED')
```

7-mesh(big matrix): يقوم هذا الايعاز بالرسم ثلاثي الابعاد.

(Ex)

```
>> mesh(rand(5))
```

8-surf(big matrix): يقوم هذا الايعاز بالرسم ثلاثي الابعاد ايضا.

(Ex)

```
>> surf(rand(5))
```

9-plot3(x,y,z,'form):ايضا تقوم هذه الدالة بالرسم ثلاثي الابعاد.

(Ex)

```
>> t=0:pi/50:10*pi;
>> Plot3(sin(t) , cos(t) , t,'s')
```

10-plotmatrix( ):يقوم هذا الايعاز برسم المصفوفات.

(Ex)

```
>> x=rand(50,3)
>>y=x*[-1 , 2 , 1 ; 2 , 0 , 1 ; 1 , -2 , 3]
>> Plotmatrix ( y )
```

11-comet(,): يقوم هذا الايعاز برسم الحلزون.

(Ex)

```
>> t=-pi:pi/200: pi;
>> comet(t,tan(sin(t) –sin(tan(t)))
```

مثال1/ اكتب برنامج يرسم دالة الsin باستخدام ايعاز plot.

الحل/ تفتح نافذة M-file ومن ثم نبدا بكتابة البرنامج التالي:

```
1. i=0:pi/100:2*pi
2. z=sin(i);
3. plote(z)
```

مثال2\ اكتب برنامج يرسم الدالة sin باستخدام fplot.

الحل\ تفتح نافذة M-file ومن ثم تبدأ بكتابة البرنامج التالي:

```
1. fplot ('sin',[0,pi])
```

ايعاز function

في بعض الاحيان نحتاج الى دوال خاصه ولكنها لا توجد داخل البرنامج ال matlab وان هذه الدوال قد نحتاجها باستمرار لذلك نقوم بصنعها بواسطه هذا الايعاز

مثال (1) اكتب برنامج يحسب الداله (Q(fun او داله مركبه لحساب داله خطأ erfc (complementary error function) ومن ثم اعتمد على هذه الداله في تكوين داله اخرى QM تقوم بحساب داله خطا اخرى erf (regenerative repeated error).
الحل/نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج التالي

اولا/ نكتب داخل نافذه M-file الصيغه الثابته لايعاز function.

```
1. Function Q= f(x)
2.Q= erf(x/sqrt(2))/2
```

ملاحظه/ يمكن تغير الحرف f بأي حرف او كلمه وحتى يمكن تغير x.

ثانيا/ نقوم بخزن البرنامج باسم داخل مجلد (bin) ثم داخل مجلد (work) وليكن نفس اسم الدالة Q.

ثالثا/ نكتب على نافذه (command win)اسم البرنامج المخزون مع اعطاء قيم لكل المداخل وفي هذا البرنامج لدينا مدخل واحد وهوx.

```
>>Q(1)
ans= 0.1578
```

ولان نقوم بكتابه البرنامج ثاني لحساب الداله QM
نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج QM

```
1. Function QM= f(x,m)
2.QM= Q(x\ sqrt(m))
```

ثم نقوم بخزن البرنامج بنفس الطريقه السابقه تحت اسم QM او اي اسم اخر وليكن اسمك لتنفيذ البرنامج كما في الداله السابقه ولكن لدينا عدد المداخل اثنان .
بعد كتابه اسم البرنامج الجديد في نافذه (command win) نكتب:

```
QM(1,3)
```

حيث قيمه x=1 وقيمه m=3

ان داله function يمكن وضع داخلها عدة مداخل او عده مخارج او كلاهما فتكون الصيغة كالاتي:

نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج

```
1. Function a= f(x, y, z, r)
2.x*y*r⁄sin(z)
```

عند التنفيذ في نافذه (command win) يجب اعطاء قيم تساوي عدد المداخل مع الاخذ بنظر الاعتبار
تسلسل هذه المداخل في التعريف

```
>>a(1,2,4,5)
```

مثال/2 نفتح نافذه M-file ومن قم نبدا بكتابه البرنامج

```
1. Function [x,y]= f(m)
2. x =sin(m⁄(m-1))
3.y= cos (m⁄(m-1))
```

وبعد خزن البرنامج نكتب في نافذه command:

```
>> اسم البرنامج (m)
```

مثال/3 نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج

```
1. Function [x, y, z]= tr(m,n,o)
2. x =sin(m⁄0)
3.y= sin (m-n)
 4. z=cos(0⁄(m-n))
```

وبعد خزن البرنامج نكتب في نافذه command:

```
>> اسم البرنامج (2, 4, 3)
```

حيث قيمه m=2وقيمه n=4وقيمه o=3.

مثال1/ احسب في برنامج مشتقه المقدار y=sin(tan(x)) ثم احسب تكامل (1⁄x) من 2 الى 3 ومن ثم احسب تكامل tan(x).

الحل / نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج التالي:

```
1.clc
2. clear all
3. syms x
4. y=sin(tan(x));
5.a= diff(y)
6.b= int(1⁄x,2 ,3)
7.c=int (tan(x))
```

مثال/2 احسب ما ياتي في البرنامج

1. $\lim_{t\to 0}\frac{1+\cos(t)}{t}$ 2. $\lim_{t\to 2}\frac{t+2}{t^2-4}$

الحل/ نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج التالي

```
1.clc
2. clear all
3. syms t
4. limit((1+cos(t))⁄t)
5. limit((t+2)⁄(t^2-4),2)
```

مثال3/ احسب متسلسله تيلور لـ e^x عندما a=5 وبعدد حدود مقداره 6 وذلك في برنامج مكتوب بلغه matlab

الحل/ نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج التالي

```
1.clc
2. clear all
3. syms x
4. taylor (exp(x), 6 ,5)
```

مثال4/ حل المعادلتين الاتيتين التاليتين وذلك في برنامج بلغه matlab

2x-3y=7

X+6y=11

الحل/ نفتح نافذه M-file ومن ثم نبدا بكتابه البرنامج التالي:

```
1.clc
2. clear all
3. a= [2, -3; 1 ,6]; b=[7 ; 11]
4. z=a⁄b
```

مثال5/ احسب معكوس الداله $f=x^2-y$ بالنسبه للمتغير y .

الحل/ نفتح نافذه M- file ومن ثم نبدا بكتابه البرنامج التالي:

```
1.clc
2. clear all
3. syms x y
4. f=x^2+y
5.finverse(f,y)
```

مثال6/ اكتب برنامج بلغه matlab يحسب المقدار $1+\cos(\frac{x}{5})$ عندما x=3, x=9

الحل/ نفتح نافذه M- fileومن ثم نبدا بكتابه البرنامج التالي:

```
1.clc
2. clear all
3. syms x
4. f=1+cos(x/5);
5. subs(f,[3  9])
```

If: تنفذ العبارات بشكل مشروط(Conditionally execute statements)
عند استخدام elseif و او else ضمن عبارة If, فان الشكل العام لهذه الدالة يكون كالاتي :

```
if expressional
statements 1
elseif expression 2
statements 2
else
statements 3
end
```

حيث يقصد بـ(expression) تعبير بلغة MATLAB وعادة مايشمل على متغيرات او تعابير اصغر مرتبطة فيما بينها بعمليات علائقية (relational operators).

Relational Operators

```
< less than
<= less than or equal to
> greater than
>= greater than or equal to
== equal to
~= not equal to
```

ويمكن ان تربط التعابير البسيطة بعمليات بعمليات منطقية (logical operators) مثل (&) ويقصد بها علاقة (و) او العملية المنطقية (|) ويقصد بها (او) لكي تصبح تعابير مركبة.
اما (Statements) ويقصد بها واحدة او اكثر من العبارات بلغة MATLAB والتي تنفذ اذا كان expression صحيح (اي الشرط صحيح)

ملاحظات

* يمكن استخدام else مع if دون الحاجة الى elseif.
** يمكن استخدام elseif مع if مرة واحدة كما في الشكل العام اعلاه اوعدة مرات .
***end يكون وجودها ضروريا في كل الحالات.

مثال 1/ اكتب برنامج يقوم بادخال عدد ويظهر فيما اذا كان العدد اكبر من الصفر او اصغر اومساوي له؟

الحل/

```
clc
disp('*********This Program2*********)
x=input('Enter Any Number:');
if x>0
   disp ('This Number Greater Than 0')
elseif x<0
   disp ('This Number Less Than 0')
else
   disp (' This Number Equal to 0')
end
```

مثال 2/ اكتب برنامج يقوم بادخال درجة طالب ويظهر راسب اذا كانت الدرجة اقل من 50 ويظهر ناجح عدا ذلك؟

الحل/

```
clc
disp('*********This Program 3*********)
x=input('enter the degree:');
if x>=50
   disp ('This student is successful')
else
   disp ('This student is unsuccessful')
end
```

لاحظ في المثال الاول فاننا ان احتجنا الى استخدام كل Else من وelseif بينما في المثال الثاني استخدمنا else فقط مع الدالة if.

For: تكرار العبارات بعدد معين من المرات

الشكل العام لهذه الدالة يكون بالشكل الاتي:

```
for variable= expression
statements
end
```

غالبا مايكون التعبير(expression) بالشكل (scalar: scalar) مثل 1:5 (ياخذ المتغير القيم من 1 الى 5) اوقد يكون غير ذلك وحسب الحاجة لديك.

كان تكون القيم من 1 الى 11 ومقدار الزيادة 2 فيكون التعبير (1:2:11)
اومن 10 الى 1 فيكون بالشكل (10:-1:1) وهكذا....

```
>>for i=1:3
      s=i+2
  end
s=
   3
s=
   4
s=
   5
```

مثال/اكتب برنامج يقوم بقراءة مصفوفة ومن ثم يوجد مجموع عناصر كل صف من هذه المصفوفة؟

```
clc
disp('*********This Program 4*********)
A=input(' please enter your matrix:');
R=size(A,1);                %R is the number of Rows
for i=1:R
   S=sum(A(I,:))            %S is sum of i-th Row
End
```

While: تكرار العبارات بعدد غير محدد من المرات

الشكل العام لهذه الدالة يكون بالشكل الاتي:

```
while expression
statements
end
```

حيث ان (expression) كما في الدالة if (راجع الدالة if)

ويتكرر تنفيذ العبارات (statements) الى ان يصبح التعبير(expression) غير صحيح (اي لايحقق الشرط)

مثال/

```
>>a=-4
>>while 1+a<0
        a=a+1
    end
a=-3
a=-2
a=-1
```

لاحظ انه كلما كان الشرط (1+a<0) صحيح (متحقق) فان العبارة (a=a+1) سوف تكرر.
وبما ان هذا الشرط قد تحقق لثلاث مرات فقط , فقد تم تكرار عبارة الجمع ثلاث مرات.

Switch :التغير بين عدة حالات مستندة على التعبير المعطى الشكل العام لهذه الدالة هو:

```
switch switch-expr
    case case-expr
        statements
   case{case-expr2, case-expr3, case-expr4, …}
        statements
   otherwise
        statements
end
```

بصورة مختصرة فان الشكل العام للدالة يعني وسيلة لتنفيذ رمز بصورة مشروطة, حيث تنفذ الدالة switch واحد من مجموعة عبارات تحدد بواسطة بديل من البدائل يسمى case التي بدورها تتكون من :

** العبارة case

** واحد او اكثر من التعابير (expression)

** واحد او اكثر من العبارات (statements)

حيث تختبر في البداية فيما اذا كانت (switch-expr) مساوي الى (case-expr1) فاذا كانت مطابقة (محققة للشرط) فتنفذ العبارات التابعة لـ (case) الاولى ويذهب بعد ذلك الى end والافسوف يختبر (case-expr2) وهكذا فاذا لم يتطابق اي (expr) فسوف ينفذ العبارات بعد (otherwise) وينهي الدالة.

مثال/اكتب برنامج يقوم بقراءة حرف ومن ثم يظهر لك اسم حيوان يبدأ بهذا الحرف ؟

الحل/(سوف نكتفي بذكر اربع اسماء من الحيوانات)

```
clc
disp('*********This Program5*********)
A=input(' please enter letter ','s');      %read a letter
switch A                                   %A is the code
case{'|'}                                  %if A=='l'
     disp ('lion')
case{'r'}                                  %if A=='r'
     disp ('rabbit')
case{'m'}                                  %if A=='m'
     disp ('mouse')
case{'c'}                                  %if A=='c'
    disp ('cow')
otherwise                                  %if A ~='L , r , m and c'
     disp(' not found animal which match your letter please try again')
end
```

Exercises

1. Find the greatest relative error and the absolute error for the following:
   a) $z = a + x^2 + 2x + b - 1/x$.
   b) $y = (x_1+x_2).(x_3-x_4/x_5)$.
   c) $w = x^3 - 2x/x^2$.
2.
   a) Chop to three significance figures.
   b) Chop to three decimal places.
   c) Round to three significance figures.
   d) Round to three decimal places.

   Each of the numbers:
   2.46475; 43.4764; 0.000442; 0.8005; 8982334.
3. Find the normalized floating-point representation of each of the following numbers.
   a) 2312 b)32.56 c)0.01267 d)82,431.
4. Find the absolute and the relative error when a three-digit decimal normalized floating-point number approximates each of the real numbers in problem 3.
5. A percentile error e_p is defined as $e_{rel}\times 100$. Find e_p in problem 2 given the conditions of problem 4.
6. Represent each of the following as five-digit base 2 floating point numbers:
   a) 21 b)3/8 c)9.872 d) -0.1.
7. Do each of the following using four-digit decimal normalized floating point arithmetic and calculate the absolute and relative errors in your answer:
   a) 10,420+0.0018 b)10,424-10,416 c)(3626.6).(22.656).
8. 8. $\sqrt{19} = 4.359$ and $\sqrt{\pi} = 1.772$ correct to 4 significant figures. Find the relative and absolute errors in their sum and difference.