



MATLAB

❖ جاءت تسمية MATLAB كتركيب من الكلمتين **Matrix Laboratory** اي مختبر المصفوفات .

❖ برمجياً تعرف لغة MATLAB :-

وهي لغة برمجة عالية الاداء تستخدم لإجراء الحسابات التقنية وتقوم بحساب واخراج البيانات ضمن بيئة سهلة البرمجة، حيث يعبر عن المسألة وحلها بأشكال رياضية مشهورة.

❖ **البرنامج PROGRAM :-** وهو مجموعة الاوامر المتسلسلة (Commands or Statement) ، جملة او امر تكتب باحدى لغات البرمجة تعمل على مدخلات البرنامج (INPUT I/P) لها وظيفة محددة ضمن البرنامج للوصول الى النتائج وهي مخرجات البرنامج (OUTPUT O/P) .

وظائف لغة MATLAB :-

- 1- اجراء العمليات الرياضية والهندسية .
- 2- تطوير الخوارزميات .
- 3- النمذجة والمحاكات .
- 4- تحليل واظهار المعطيات .
- 5- اجراء الرسوم البيانية والهندسية .
- 6- تطوير التطبيقات .

تشغيل برنامج MATLAB :-

بعد الانتهاء من عملية تثبيت برنامج MATLAB ، الان ابدأ بتشغيله للتعرف على اهم سمات بيئة تطويره ، ويمكنك تشغيل برنامج MATLAB باتباع احدى الطرق التالية :-

- قم بالضغط المزدوج بزر الماوس الايسر double click على ايقونة الاختصار short cut الخاصة بالبرنامج والموجودة على سطح مكتبك Desktop وتعد هذه الطريقة من اسهل واسرع الطرق لتشغيل برنامج MATLAB .
- ولنلخص خطوات اظهار ايقونة برنامج MATLAB على سطح المكتب كما يلي :-

My computer → C:\Program Files → matlab → 2012a → bin → matlab

❖ وباستخدام احدى الطريقتين السابقة لتشغيل برنامج MATLAB سوف تظهر لك الواجهة الرئيسية للبرنامج وهي واجهة التخابط الاساسية مع المستخدم والتي تسمى سطح مكتب برنامج MATLAB ، وتتضمن هذه النافذة كافة النوافذ المرتبطة بسطح مكتب البرنامج التي سنتعرف عليها بالتفصيل .



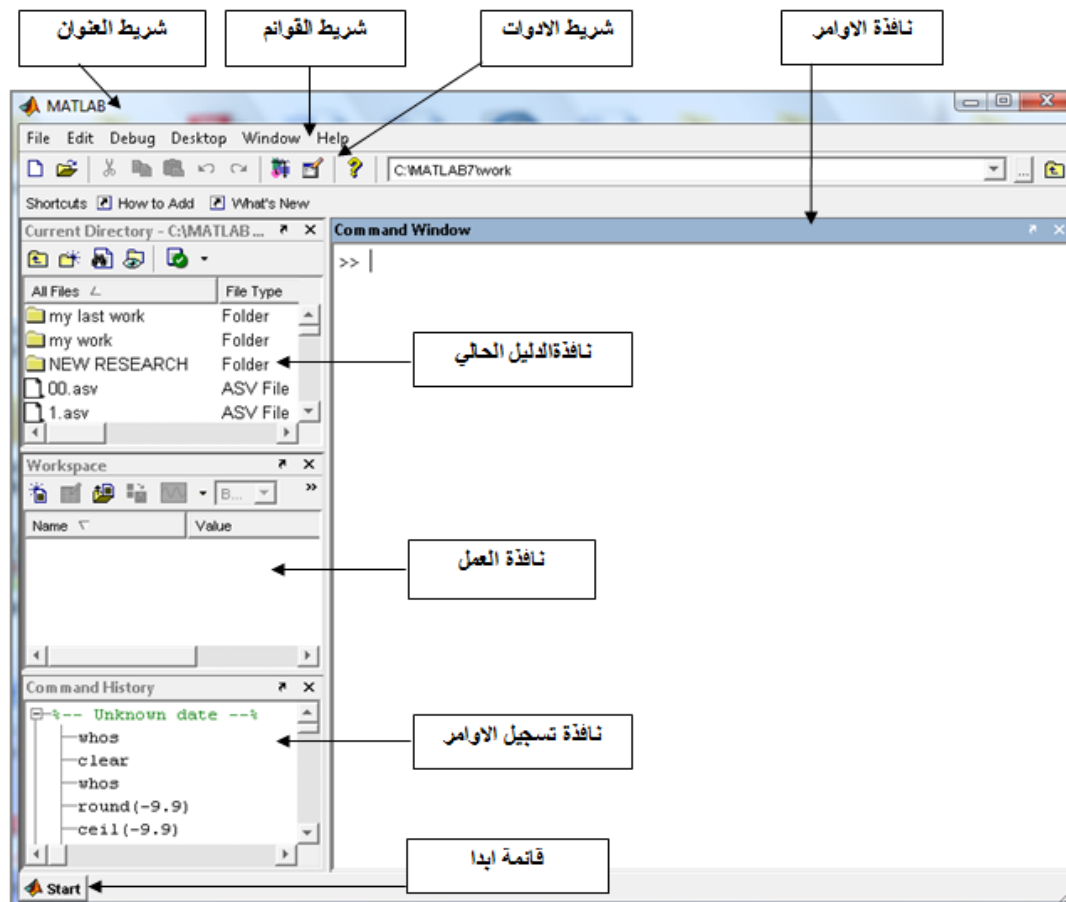
MATLAB واجهة برنامج

Components of the MATLAB Desktop

لا تختلف سمات واجهة برنامج MATLAB كثيرا عن سمات البرامج التي تعمل تحت نظام التشغيل WINDOW مثل برامج المكتب Office 2003 أو Visual Studio.Net، فكلاهما يستخدم نفس العناصر كشريط القوائم Menu Bar وشريط الادوات Tool Bar إضافة الى النوافذ الرئيسية التي يتعامل معها برنامج MATLAB.

يتكون سطح مكتب برنامج MATLAB من العناصر التالية :-

- 1- شريط العنوان Title Bar
- 2- شريط القوائم Menu Bar
- 3- شريط الادوات Tool Bar
- 4- نافذة الاوامر Command Window
- 5- منطقة العمل Workspace
- 6- المجلد الحالي Current Directory
- 7- تاريخ الاوامر Command History
- 8- شريط الحالة Status Bar
- 9- قائمة ابدأ Start Menu
- 10- منقح البرنامج Editor





1. شريط العنوان: Title Bar

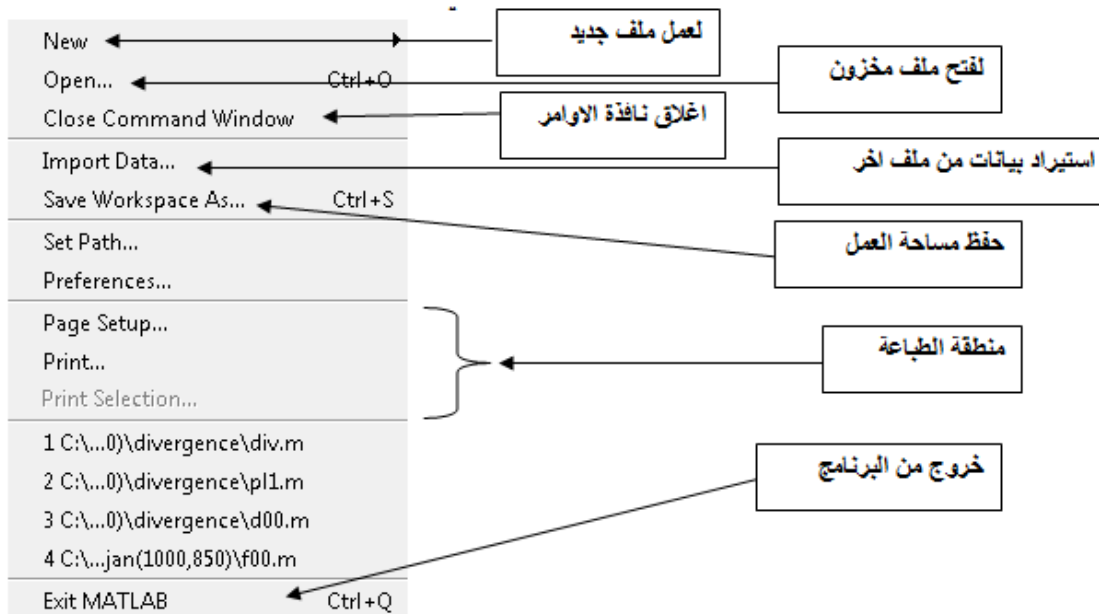
يحتوي هذا الشريط على اسم ورمز البرنامج واسم الملف او النافذة المفتوحة حاليا ، كما يحتوي في اقصى يمينه على مفاتيح التحكم الثلاثة :

Close عند الضغط على هذا المفتاح يتم اغلاق النافذة المفتوحة حاليا. 
Maximize/Restore عند الضغط على هذا المفتاح يتم تكبير النافذة الى الحد الاقصى ملئ الشاشة او استعادة النافذة الى حد اصغر. 
Minimize عند الضغط على هذا المفتاح يتم تصغير النافذة على شريط المهام Taskbar 
ولاستعادتها مرة اخرى يتم الضغط على اسم النافذة على شريط المهام.

2. شريط القوائم: Menu Bar

يوجد شريط القوائم اسفل شريط العنوان مباشرة ويحتوي على قوائم برنامج MATLAB الاساسية والتي تحتوي على الاوامر والخيارات اللازمة للتعامل مع البرنامج .
ويضم هذا الشريط القوائم التالية :

● **قائمة ملف File** : تتكون هذه القائمة من العديد من الخيارات ، والتي تنفذ كل منها وظيفة محددة



● **قائمة التعديل Edit** : توجد في هذه القائمة اوامر النسخ copy ، القص cut ، اللصق paste ، البحث find ولكنها تحتوي على ثلاث ادوات مهمة وهي : clear , Clear command window , clear workspace , command history كل منها يختص بمسح محتويات نافذة معينة كما مبين في الشكل ادناه



Undo	Ctrl+Z
Redo	
Cut	Ctrl+W
Copy	Alt+W
Paste	Ctrl+Y
Paste Special...	
Select All	
Delete	Ctrl+D
Find...	
Find Files...	
Clear Command Window	
Clear Command History	
Clear Workspace	

مسح محتويات نافذة الاوامر

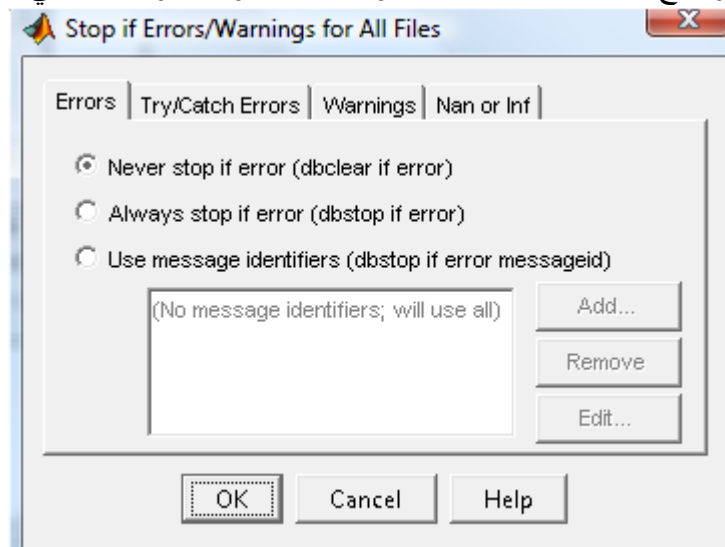
مسح محتويات نافذة تسجيل الاوامر

مسح محتويات منطقة العمل

- قائمة debug : هذه القائمة خاصة بمعالجة البيانات ، والطريقة المتبعة في مواجهة الاخطاء كما في الشكل

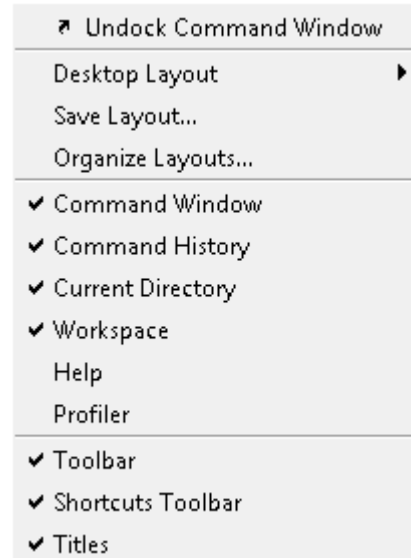
✓ Open M-Files when Debugging	
Step	F10
Step In	F11
Step Out	Shift+F11
Continue	F5
Clear Breakpoints in All Files	
Stop if Errors/Warnings...	
Exit Debug Mode	

- فمثلا عند اختيار Stop if Errors /Warnings... ستظهر نافذة تعطي حرية الاختيار في تصرف برنامج MATLAB عند حدوث اخطاء او تحذيرات كما في الشكل التالي :

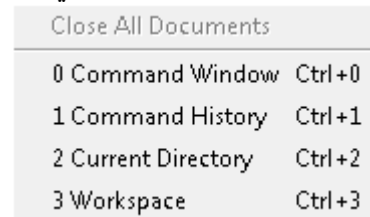




قائمة Desktop: من خلال هذه القائمة يتم التحكم بمحتوى الواجهة الخاصة ببرنامج MATLAB فمثلا يمكن اظهار نافذة الاوامر او اخفائها ، كما في الشكل :



قائمة Window: حيث يمكن التنقل بين ملفات برنامج MATLAB المختلفة ، وكذلك النوافذ مثل نافذة الاوامر وغيرها كما في الشكل التالي :



قائمة Help: هذه القائمة تقوم بتوفير المساعدات الضرورية في البرنامج ووسائل الاتصال بالشركة المصنعة واخر التحديثات وكذلك تحتوي على دليل تعليمي باللغة الانكليزية كما في الشكل التالي :

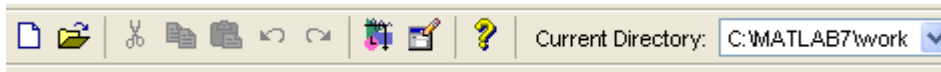
• **ملاحظة :-** يستخدم الامر doc للوصول الى دليل الاوامر ، وبالشكل التالي :

اسم الامر >> doc



3. شريط الادوات : Toolbar

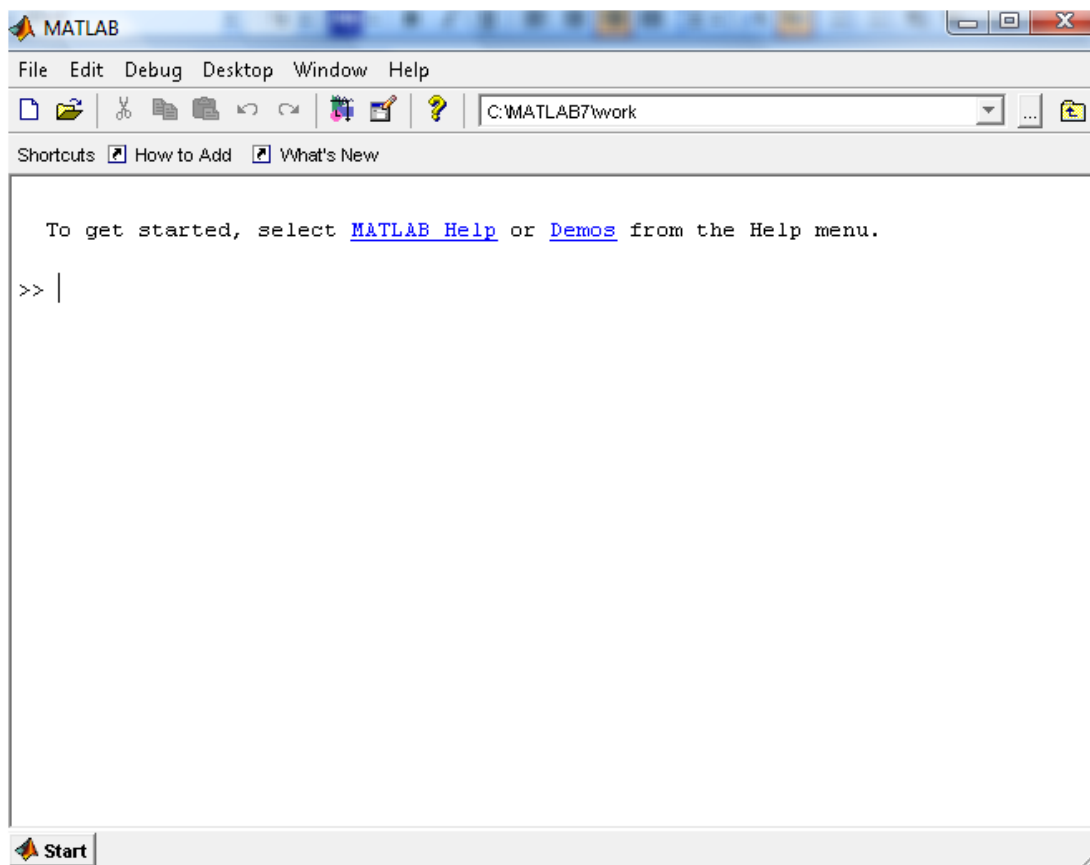
يحتوي هذا الشريط على ايقونات لبعض الادوات الموجودة في شريط القوائم ويعتبر وسيلة سريعة يمكننا من خلالها تنفيذ الاوامر بطريقة سريعة بدلا من فتح القوائم والبحث بداخلها عن الاوامر المطلوبة .



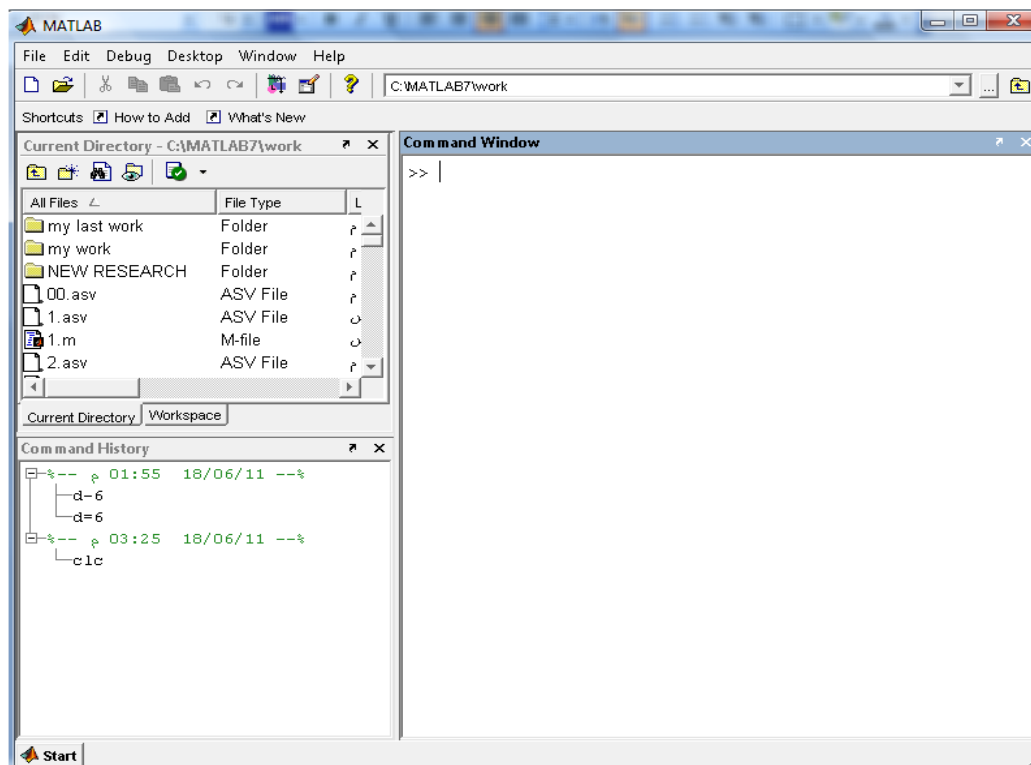
4. نافذة الاوامر: Command Window

تستخدم هذه النافذة لتحرير المدخلات Inputs والاوامر Commands الى البرنامج حيث يظهر بها المحرر على الشكل >> متبوعا بمؤشر يومض (يظهر ويختفي) بصورة متكررة بحيث يتم كتابة الاوامر Commands الى يمين المحرر ، ويعمل برنامج MATLAB على تحليل تلك البيانات ومدى مطابقة المدخلات للوظيفة المطلوبة منها . وفور الانتهاء من كتابة الامر والضغط على مفتاح Enter يتم الحصول على النتائج المنفذة ويتم عرضها في هذه النافذة .

يمكن فتح نافذة محرر الاوامر بشكل منفصل عن سطح برنامج MATLAB بالنقر على قائمة Desktop ثم القائمة الفرعية Desktop Layout واختيار command window only . كما في الشكل التالي :

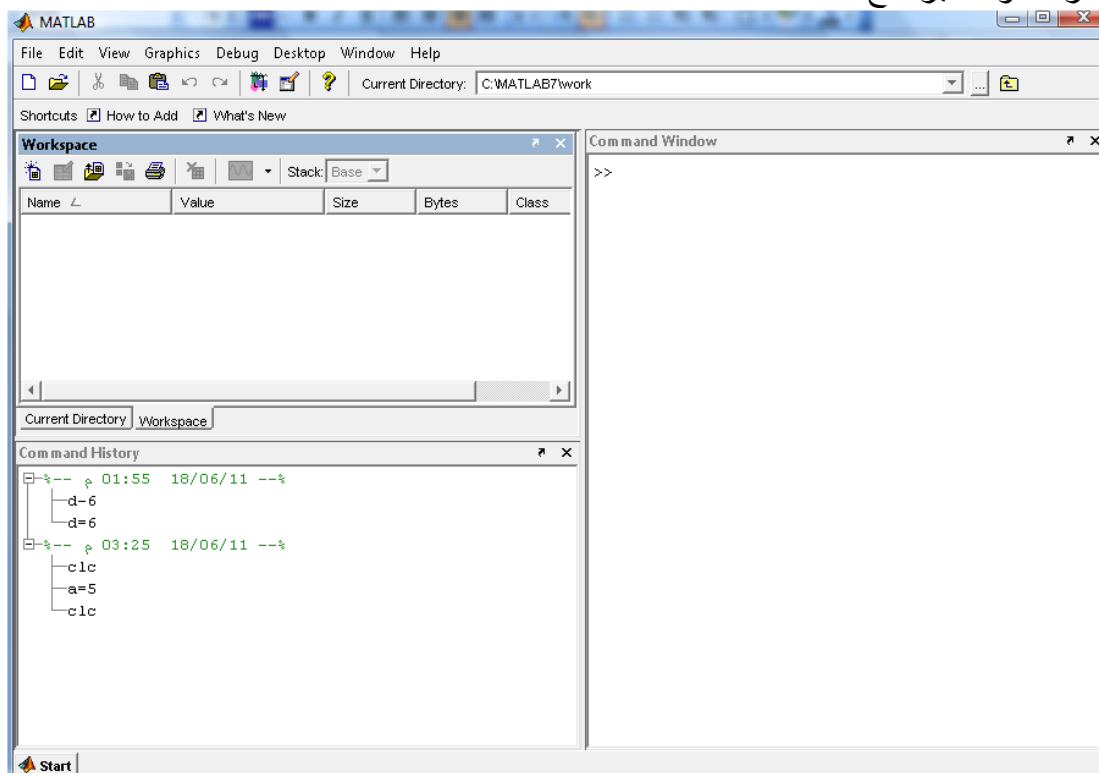


ولجعل النافذة ضمن نافذة سطح برنامج MATLAB نختار Default من خلال اختيار نفس القوائم المذكورة اعلاه وبهذا سوف تعود نافذة محرر الاوامر الى الوضع الافتراضي لها كما في الشكل التالي :



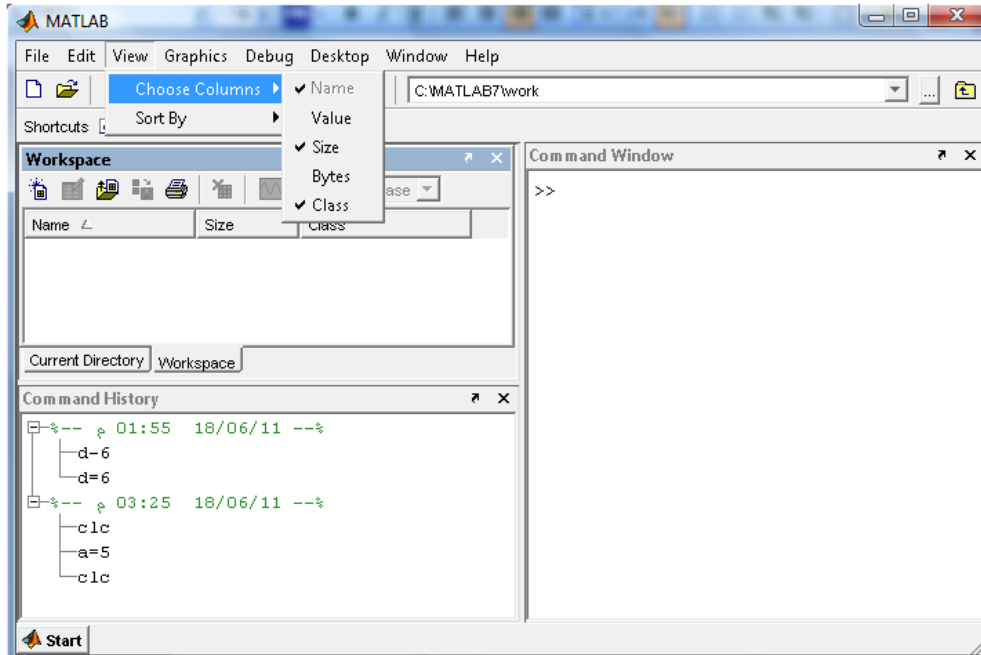
5. نافذة العمل: Workspace

ويتم فيها عرض أسماء جميع المتغيرات names وقيم هذه المتغيرات values ونوع المتغيرات class وحجم وابعاد هذه المتغيرات size التي تم استخدامها حالياً من قبل المستخدم الى حين اغلاق برنامج MATLAB ، ويمكن من خلالها اعادة تحرير وتعيين قيم هذه المتغيرات ، ولهذا فهي تعد بمثابة الذاكرة المؤقتة لبرنامج MATLAB، لاحظ الشكل

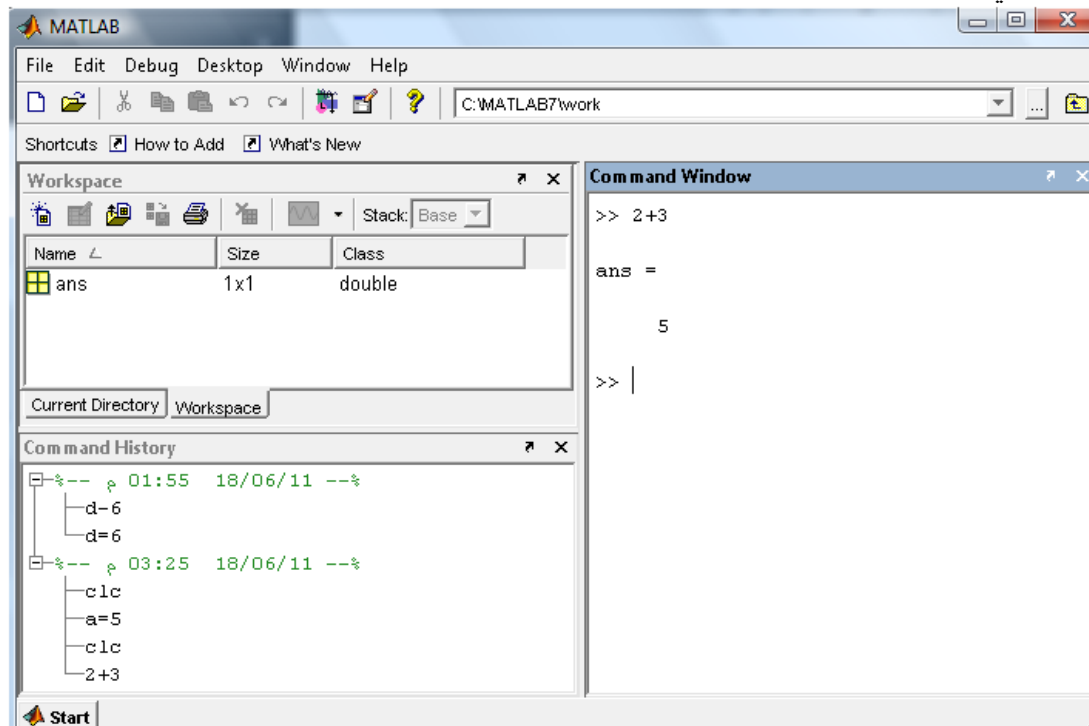




ويمكن التحكم في خصائص المتغيرات المسجلة في نافذة العمل وذلك بتنشيط نافذة العمل ثم اختيار الامر Choose Columns من قائمة View ووضع علامة صح بجانب الخيار المراد اظهاره . كما في الشكل التالي :



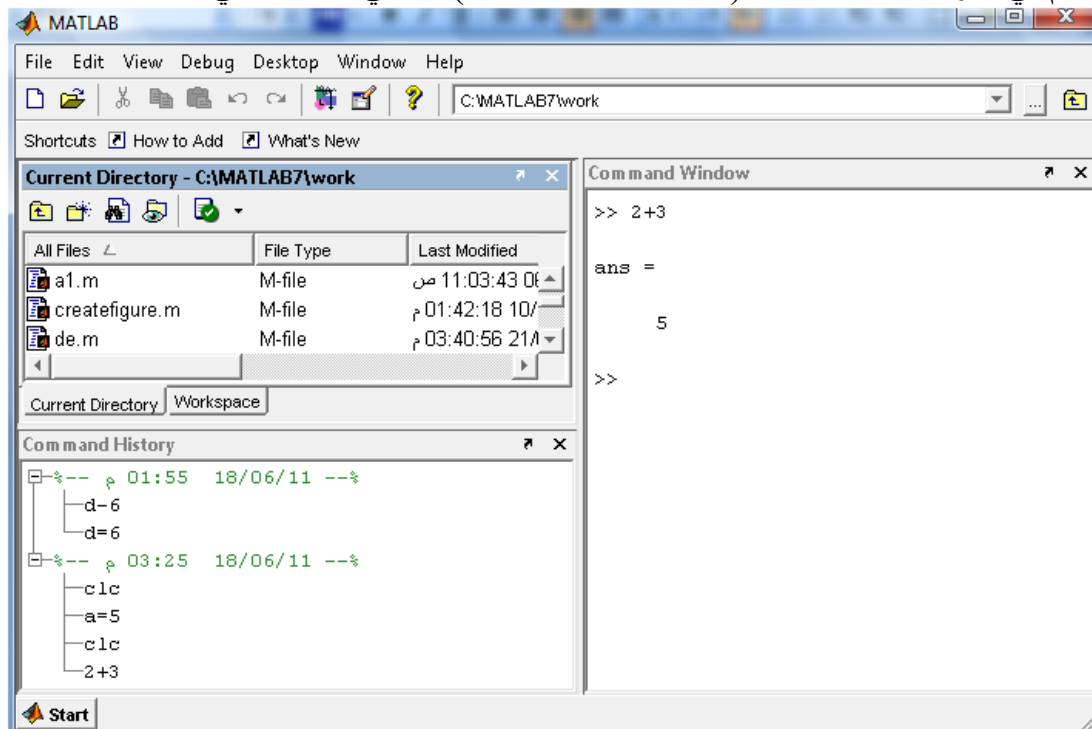
فمثلا عند القيام بعملية جمع $2+3$ في نافذة الاوامر سوف يتم تخزين النتيجة في نافذة العمل كما في الشكل التالي :



وكذلك لدى ادخال قيم المتغيرات او المصفوفات فانه سوف يتم تخزينها في نافذة العمل كل على حدة .

6. نافذة الدليل الحالي : Current Directory

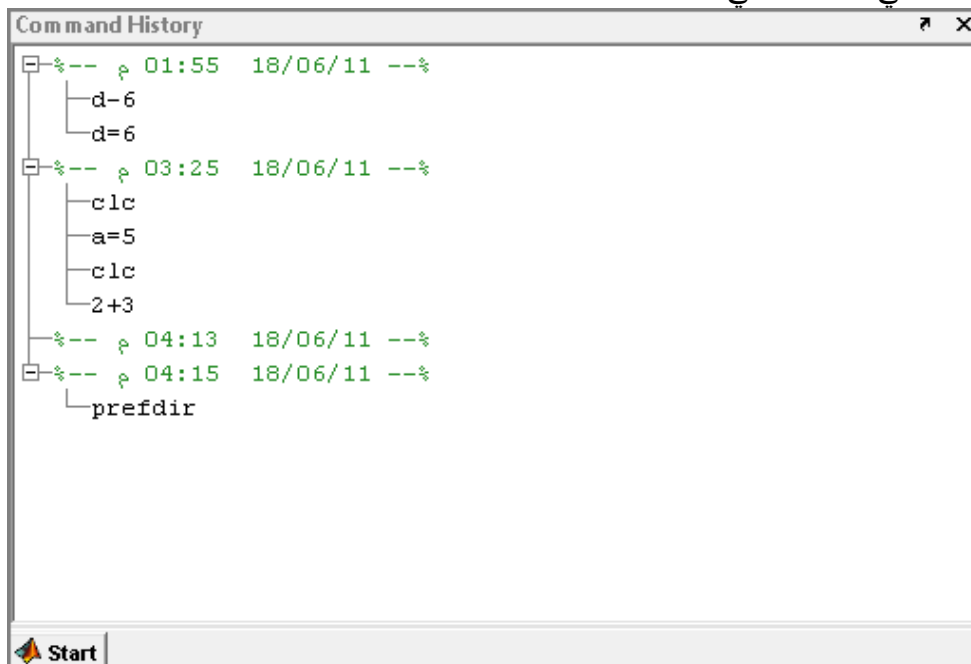
تستخدم في ادارة الملفات التنفيذية (مثل ملفات M- Files) كما في الشكل التالي :



كما يقوم الدليل الحالي بعرض مجلد العمل الحالي وهو الدليل الافتراضي الذي يتم حفظ ملفات البرنامج التنفيذية بداخله Work ويمكن تغيير المجلد الافتراضي بمجلد اخر بالضغط على مفتاح الاداة للذهاب الى المجلدات الاخرى او عمل مجلد جديد بالضغط على .

7. نافذة تاريخ الاوامر السابقة : Command History

يتم تسجيل جميع الاوامر التي تم ادخالها في نافذة الاوامر Command Window في نافذة تسجيل الاوامر بالتاريخ والوقت ، حيث يمكن استرجاع هذه الاوامر فيما بعد لتنفيذها مرة اخرى في نافذة الاوامر . كما في الشكل التالي :

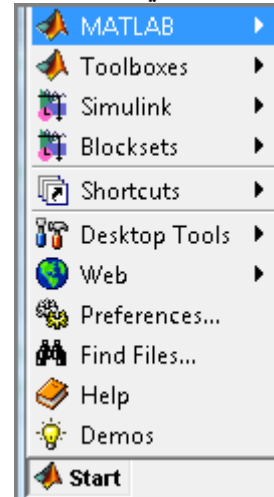


8. شريط الحالة : Status Bar

يمثل حالة البرنامج فيما ان يكون جاهزا Ready لتنفيذ الاوامر التي يقوم المستخدم بإدخالها او يكون مشغول Busy بإجراء حسابات معينة.

9. قائمة ابدأ : Start Menu

تستخدم هذه القائمة للوصول الى التطبيق المراد تنفيذه ، وتستخدم هذه القائمة في المراحل المتقدمة من برنامج MATLAB، فهي تشبه قائمة Start الموجودة في نظام التشغيل WINDOWS كما في الشكل التالي :

**التحكم بنوافذ البرنامج**

تظهر بعض النوافذ المرتبطة بسطح مكتب برنامج MATLAB مثل Command Window and Workspace , Command History بأحد الوضعين التاليين :

1. **Docked** حيث تكون النافذة غير قابلة للتحريك من مكانها (اي تكون ضمن سطح مكتب البرنامج) .
2. **Undocked** حيث تكون النافذة قابلة للتحريك (اي تكون منفصلة عن سطح مكتب البرنامج ويمكن تعديل ابعادها).

ولجعل اي نافذة من النوافذ المذكورة سابقا قائمة Undocked نضغط على زر Undocked Command Window



انواع البيانات في برنامج MATLAB

MATLAB Data Types

انواع البيانات : تنقسم انواع البيانات التي يتعامل معها برنامج MATLAB الى ثلاثة انواع اساسية

1. بيانات عددية Numerical Data وتنقسم الى :

- قيم عددية مفردة Scalars
- مصفوفات عددية Matrices
- متجهات Vectors او منظومات Arrays او كثيرات الحدود Polynomials .

2. بيانات رمزية Symbolic Data وتنقسم الى :

- قيم رمزية مفردة Symbolic Scalars
- مصفوفات رمزية Symbolic Matrices
- منظومات رمزية Symbolic Arrays او كثيرات الحدود Polynomials .

3. السلاسل الحرفية (Character Arrays (Strings

1- البيانات العددية Numerical Data

❖ المتغيرات العددية المفردة Scalars

يتم تعريف قيمة عددية مفردة (وحيدة) Scalar في برنامج MATLAB من خلال كتابة اسم المتغير (الذي سنخزن فيه القيمة العددية) ، ثم علامة (=) ، ثم القيمة العددية المفردة . مثلا نكتب الامر $a=3$ داخل نافذة محرر الاوامر Command Window ثم نضغط على مفتاح enter . وهكذا بالنسبة لبقية المتغيرات ذات القيم المفردة كما مبين في المثال التالي :

```
>> a=3
a =
    3
>> b=5
b =
    5
>> c=a+b
c =
    8
```

❖ المصفوفات Matrix

المصفوفة عبارة عن مجموعة من الأرقام تتكون من صفوف أفقية وأعمدة رأسية . يمكن تعريف مصفوفة عددية معينة من خلال كتابة اسم المتغير (الذي سنخزن فيه القيم العددية لعناصر المصفوفة) ثم علامة (=) ، ثم نفتح قوس مربع ايسر ([) ليتم ادخال قيم عناصر المصفوفة بكتابة عناصر الصف الاول ، ثم الثاني وهكذا ثم اغلق المصفوفة بقوس مربع ايمن (]) مثلاً لكتابة المصفوفة التالية

$$A = \begin{bmatrix} 1 & 3 \\ 6 & 4 \end{bmatrix} \quad -:$$

يتم كتابة عناصر الصف الاول ويتم الفصل بين كل عنصر من عناصر الصف الاول اما باستخدام علامة الفاصلة (,) Comma او بعمل مسافة Space بين كل عنصر والعنصر الذي يليه ، ويتم الفصل بين الصف الاول والصف الذي يليه باستخدام علامة الفاصلة المنقوطة (;) Semicolon او بالضغط على مفتاح Enter في لوحة المفاتيح بحيث يتم ادخال عناصر كل صف على سطر خاص به كما يلي

```
>> A=[1 3;6 4]
```

```
A =
```

```
1 3
6 4
```

```
>> A=[1,3;6,4]
```

```
A =
```

```
1 3
6 4
```

```
>> A=[1 3
```

```
6 4]
```

```
A =
```

```
1 3
6 4
```

```
>> A=[1,3
```

```
6,4]
```

```
A =
```

```
1 3
6 4
```

❖ المتجهات Vectors

هي مجموعة من الأرقام توضع في صورة صف واحد وتسمى في هذه الحالة متجهات صفية Row Vectors او عمود واحد وتسمى في هذه الحالة متجهات عمودية Column Vectors وبالتالي فهي تمثل مصفوفة احادية .

يمكن تعريف المتجه الصفّي من خلال كتابة اسم المتغير (الذي سنخزن فيه القيمة العددية) ثم علامة (=) ثم نفتح قوس مربع ايسر ([) ثم ندخل قيم عناصر المتجه ، علماً بأنه يتم الفصل بين كل عنصر والعنصر الذي يليه في المتجه اما بمسافة Space او فاصلة (,) Comma من لوحة المفاتيح ، وبعد الانتهاء من ادخال عناصر المتجه ، نغلق المتجه بقوس مربع ايمن (]) كما يلي :

```
>> A=[1 2 3 4 5]
```

```
A =
```

```
1 2 3 4 5
```



```
>> A=[1,2,3,4,5]
```

```
A =
```

```
1 2 3 4 5
```

اما المتجهات العمودية فانه كما في الطريقة اعلاه يتم تعريف المتجه العمودي باستثناء طريقة الفصل بين عناصر المتجه فيتم الفصل بين كل عنصرين باستخدام علامة الفاصلة المنقوطة Semicolon (;) او بالضغط على مفتاح Enter في لوحة المفاتيح كما يلي :

```
>> A=[1;2;3]
```

```
A =
```

```
1
```

```
2
```

```
3
```

```
>> A=[1
```

```
2
```

```
3]
```

```
A =
```

```
1
```

```
2
```

```
3
```

2- البيانات الرمزية Symbolic Data

القيم الرمزية المفردة Symbolic Scalars ❖

السلاسل الحرفية هي مجموعة من الحروف النصية والارقام والرموز التي يتعامل معها برنامج MATLAB على انها حروف نصية . في حالة اذا كانت القيمة المفردة سلسلة حرفية string فانه يمكن تعريفها من خلال كتابة اسم المتغير (الذي سنخزن فيه السلسلة الحرفية) ثم علامة (=) ثم نكتب الدالة sym ونضع القيمة المفردة بين اقواس صغيرة بداخلها علامتي اقتباس مفردة single quotations mark ، كما هو موضح في الامثلة التالية :

```
>> D=sym('K')
```

```
D =
```

```
K
```

```
>> E=sym('X')+sym('Y')
```

```
E =
```

```
X+Y
```

```
>> sym H
```

```
>> H=sym('welcome in MATLAB programming')
```

```
H =
```

```
welcome in MATLAB programming
```



2 - البيانات الرمزية Symbolic Data

❖ المصفوفات الرمزية Symbolic Matrices

يعرف برنامج MATLAB المصفوفات الرمزية Symbolic Matrices من خلال كتابة الامر **syms** متبوعا بالرموز المستخدمة في المصفوفة مع الفصل بين كل رمز والرمز الذي يليه بمسافة **space**، ثم نكتب اسم المتغير متبوعا بعلامة (=) ، ثم نفتح قوس مربع ايسر ([)) ثم ندخل عناصر المصفوفة الرمزية ويتم الفصل بين كل عنصر والعنصر الذي يليه في المصفوفة اما باستخدام **Space** او فاصلة **Comma (,)** ، وبعد الانتهاء من ادخال عناصر الصف الاول ، قم بإدخال عناصر الصف الثاني بحيث يتم الفصل بين كل صف والصف الذي يليه بعلامة الفاصلة المنقوطة **semicolon (;)** ، وبعد الانتهاء من ادخال عناصر المصفوفة قم بغلق عناصر المصفوفة بقوس مربع ايمن (]) ، كما هو موضح في الامثلة التالية :-

>> % Some Examples of Symbolic Matrices Defining

>> syms x z y

>> M1=[3*x 5*z ; 7*y 9*x]

M1 =

[3*x, 5*z]

[7*y, 9*x]

>> M2=[x 3*y -4*z ; z 5*x -2*y]

M2 =

[x, 3*y, -4*z]

[z, 5*x, -2*y]

>> M3=[3/(x+y) 2*y z/x ; x-y (y+z)/3 3*y-z]

M3 =

[3/(x + y), 2*y, z/x]

[x - y, y/3 + z/3, 3*y - z]

❖ ملاحظة :- يستخدم الامر **class** لمعرفة نوع بيانات المتغير المستخدم في برنامج MATLAB .

>> class(M1)

ans=

char

>> A=[1;2;3];

>> class(A)

ans =

double



انواع المتغيرات في برنامج MATLAB

1. متغيرات مسبقة التعريف في البرنامج Built in (Predefined) Variables

هي مجموعة من الثوابت constants والقيم الخاصة special values المحجوزة في البرنامج حيث تأتي معرفة تلقائياً في بنية البرنامج الداخلية ويمكن استخدامها مباشرة دون ان يتم تعريفها.

هو المتغير الافتراضي لأي ناتج عملية حسابية في برنامج MATLAB عند عدم اعطاء اسم متغير للقيمة الناتجة، مثلاً <pre>>> 5+3 ans = 8</pre> يقوم البرنامج تلقائياً بخزن ناتج عملية الجمع في المتغير ans لأننا لم نعرف اسم متغير لناتج هذه العملية الحسابية .	ans
هي النسبة الثابتة $\pi = 22/7$ وتعرف في البرنامج على الشكل التالي : <pre>>> pi ans = 3.1416</pre>	Pi
يعبر عن قيم اللانهاية infinity ∞ الناتجة من القسمة على صفر <pre>>> 1/0 ans = Inf</pre>	Inf
تعبر عن القيمة التي ليست رقم ، وهي اختصار جملة Not a Number وقد تنتج عندما تكون قيمة الناتج يساوي (0/0) او لتعبر عن ان المعلومات مفقودة ، او غير متوفرة ، والذي قد يكون سببه فشل البرنامج في الحساب . <pre>>> 0/0 Warning: Divided by zero. ans = NaN</pre>	NaN
يتم استخدام احد هذين الرمزین عند تعريف الاعداد المركبة (المعقدة) فهما يمثلان الجزء التخيلي للاعداد المركبة حيث يتم استخدامهم على الشكل التالي : <pre>>> 3+4*i ans = 3.0000 + 4.0000i >> 3+4*j ans = 3.0000 + 4.0000i</pre> مع ملاحظة ان كلا الرمزین يمثلان العدد المركب $\sqrt{-1}$	i , j



هي قيمة متناهية في الصغر يطلق عليها ايبسلون Epsilon تستخدم في بعض التطبيقات الرياضية الخاصة وتساوي 2^{-52} ، وتعرف بالشكل التالي :- <pre>>> eps ans= 2.2204e -016</pre>	العدد الطبيعي (ε)
---	--------------------------

2. متغيرات تعرف بواسطة المستخدم User-defined Variables

وهي المتغيرات التي يقوم المستخدم بتعريفها بإعطائها قيمة عددية أو نصية ، وسيتعرف البرنامج على نوع هذه المتغيرات دون تحديده كما ذكرنا سابقا ، ويتم تسمية المتغير في برنامج MATLAB ضمن شروط معينة .

❖ شروط تسمية المتغيرات داخل برنامج Matlab:-

1. يجب ان يبدأ اسم المتغير بحرف وليس برقم او برمز فمثلا لا يمكن كتابة اسم المتغير على الشكل $1a=5$ وبدلا من ذلك يمكننا كتابة اسم المتغير على الشكل $a1=5$.
2. لا يمكن ان يحتوي اسم المتغير على مسافة (فراغ)، فمثلا لا يمكن كتابة اسم المتغير على الشكل $a\ val$ وبدلا من ذلك يمكن استخدام علامة الشرطة السفلية (_) Underscore على الشكل a_val .
3. يجب ان لا يحتوي اسم المتغير على بعض الرموز الخاصة مثل $\# , @ , ^ , ? , \% , * , + , - , < , > , (,) , [,] , ! , \ , /$.
4. يجب ان لا يأخذ اسم المتغير اسم امر او دالة محجوزة في برنامج MATLAB ، فمثلا لا يمكن تسمية المتغير `if` لان هذا الاسم من الكلمات المحجوزة `reserved words` او الكلمات المفتاحية `keywords` داخل اللغة ، ولكن يمكن استخدام كلمات شبيهة لها من خلال دمج ارقام معها مثل `if1` او جعل اول حرف منها كبيرا `capital` مثل `IF` . وهذه قائمة ببعض الكلمات المحجوزة داخل البرنامج `if elseif else end for while break continue return switch case otherwise try catch function global persistent` يتم التعرف على قائمة الكلمات المحجوزة في برنامج MATLAB بكتابة الامر `iskeyword` في نافذة الاوامر `command window` كما يلي :

```
>> iskeyword
```

```
ans =
```

```
'break'
'case'
'catch'
'classdef'
'continue'
'else'
'elseif'
'end'
'for'
'function'
'global'
'if'
'otherwise'
'parfor'
```



'persistent'
'return'
'spmd'
'switch'
'try'
'while'

5. يجب ان لا يزيد عدد الاحرف التي يتكون منها اسم المتغير عن 63 حرف وسيهمل اي رمز يزيد عن 63 حرف .

6. برنامج MATLAB حساس لحالة الاحرف case sensitive حيث يميز بين الاحرف الكبيرة capital letters والاحرف الصغيرة small letters ، فمثلا عند القيام بتسمية متغير بالاسم a فان برنامج MATLAB يتعامل معه على ان له قيمة مختلف عن المتغير A .

❖ كيفية اجراء العمليات الرياضية في برنامج MATLAB :-

الجدول التالي يبين كيفية اجراء العمليات الحسابية (الرفع الى الاس , الضرب , القسمة , الباقي من القسمة , الجمع , الطرح) .

الرمز في برنامج MATLAB	العمليات الرياضية	الامثلة
^	الرفع الى القوة او الاس	2^8
*	الضرب	$6 * 3.14$
/	القسمة	$19.54 / 7$
\	القسمة العكسية	$7 \backslash 19.54 = 19.54 / 7$
rem , mod	باقي القسمة	$16 \% 5$
+	الجمع	$3 + 22$
-	الطرح	$54.4 - 16.5$

❖ اسبقيات (أولوية) العمليات الرياضية في برنامج MATLAB :-

- 1- الاقواس .
 - 2- الرفع الى القوة .
 - 3- الضرب والقسمة .
 - 4- الجمع والطرح .
- ملاحظة :- تحسب العمليات الرياضية من اليسار الى اليمين التي تحتوي على نفس الاسبقية .

Ex:- $>> (3 + 22) * (15.7 - 8)$

(1) (2)

└──────────┘

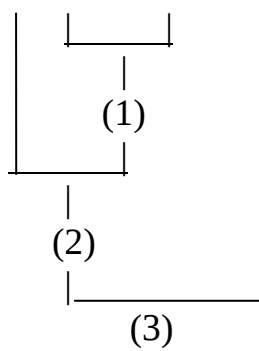
┆

(3)

ans=
192.5

Ex:-

>> 3 + 22 * 15.7 - 8



ans=

340.4

❖ ملاحظات :-

1. عند وضع علامة الفاصلة المنقوطة (;) semicolon في نهاية اي من الاوامر (المدخلات) السابقة فسوف يتم تنفيذ الامر دون اظهار الناتج . لاحظ الفرق في المثال التالي :

>> a=4+5

a =

9

>> a=4+5;

>>

2. اثناء كتابة البرنامج قد يحتاج المبرمج الى اضافة تعليق ليشرح سطر او جزء معين من الفقرات البرمجية للبرنامج ، او قد يحتاج الى تعطيل تنفيذ بعض الاوامر بصورة مؤقتة بدلا من حذفها او اعادة كتابتها مرة اخرى ، وفي برنامج MATLAB تستخدم علامة النسبة المئوية % ثم يتم كتابة التعليق بعد العلامة مباشرة او تستخدم نفس العلامة قبل الامر المراد تعطيل تنفيذه كما يلي :

>> % summation process

>> a=4+5

a =

9

عند كتابة نفس الامر السابق بالشكل

>> %a=4+5

فلا يتم تنفيذ الامر لأنه متبوع بالرمز %.

3. يمكن استدعاء متغير ومعرفة قيمته بعد ان تم ادخاله في البرنامج وذلك من خلال كتابة المتغير فقط والضغط على مفتاح enter كما يلي :

>> a

a =

5

**❖ بعض دوال التحكم في نافذتي `command window`، `workspace` :**

يمكن تلخيص دوال التحكم في نافذتي `workspace` , `command window` وتنظيمها في الجدول التالي :

Operation	Function
مسح جميع محتويات نافذة <code>command window</code> فقط دون مسحها من نافذة <code>workspace</code>	<code>Clc</code>
مسح جميع محتويات نافذة <code>workspace</code> المتضمنة جميع المتغيرات التي تم استخدامها في البرنامج	<code>Clear</code>
مسح المتغيرات <code>a b c</code> فقط من نافذة <code>workspace</code>	<code>clear a b c</code>
مسح جميع المتغيرات الموجودة في نافذة <code>workspace</code> والتي تبدأ بحرف <code>a</code>	<code>clear a*</code>
عرض محتويات نافذة <code>workspace</code> والمتضمنة جميع المتغيرات التي تم استخدامها في البرنامج	<code>Who</code>
عرض محتويات نافذة <code>workspace</code> بالتفصيل (الاسم ، الأبعاد ، الحجم ، النوع)	<code>Whos</code>
حفظ جميع محتويات نافذة <code>workspace</code> في الملف الافتراضي <code>matlab.mat</code>	<code>Save</code>



الدوال الرياضية :

1. الدوال الأسية: Exponential Functions:

Example	Function in MATLAB form	Operation
>> exp(0) ans = 1	exp(x)	الدالة الاسية
>> log(1) ans = 0	log(x)	دالة اللوغارتم الطبيعي ln
>> log10(2) ans = 0.3010	log10(x)	دالة اللوغارتم للاساس 10
>> log2(2) ans = 1	log2(x)	دالة اللوغارتم للاساس 2
>> pow2(3) ans = 8	pow2(x)	دالة الرفع الى قوة للاساس 2
>> sqrt(4) ans = 2	sqrt(x)	دالة الجذر التربيعي
>> power(3,3) ans = 27	power(X,Y)	دالة الرفع للاساس x

2. الدوال المثلثية: Trigonometric Functions:

مثال	الامر في برنامج MATLAB	الدالة المثلثية
>> sin(5) ans = -0.9589	sin(angle)	الدالة sin
>> cos(5) ans = 0.2837	cos(angle)	الدالة cos
>> tan(5) ans = -3.3805	tan(angle)	الدالة tan
>> sec(5) ans = 3.5253	sec(angle)	الدالة 1/cos
>> csc(5) ans =	csc(angle)	الدالة 1/sin



-1.0428		
>> cot(5) ans = -0.2958	cot(angle)	الدالة 1/tan
>> asin(5) ans = 1.5708 - 2.2924i	asin(angle)	معكوس الدالة sin
>> acos(5) ans = 0 + 2.2924i	acos(angle)	معكوس الدالة cos
>> atan(5) ans = 1.3734	atan(angle)	معكوس الدالة tan
>> asec(5) ans = 1.3694	asec(angle)	معكوس الدالة sec
>> acsc(5) ans = 0.2014	acsc(angle)	معكوس الدالة csc
>> acot(5) ans = 0.1974	acot(angle)	معكوس الدالة cot

***ملاحظة :-** جميع الدوال المثلثية اعلاه مقاسة بالتقدير الدائري Radian degree ، ولغرض تحويل التقدير الى الدرجات degrees فانه يمكن ذلك بثلاث طرق :

الاولى :- بإضافة الحرف d قبل اي دالة ، مثلاً نكتب دالة sin بالشكل sind.

الثانية :- بضرب الزاوية قبل تنفيذ الدالة بالمقدار (pi/180) كما في المثال التالي :

```
>> sind(30)
```

```
ans =
```

```
0.5000
```

```
OR
```

```
>> sin(30*(pi/180))
```

```
ans =
```

```
0.5000
```

ولتحويل الزاوية من تقدير الدرجات الى التقدير الدائري فنضرب الزاوية قبل تنفيذ الدالة بالمقدار (180/pi) كما في المثال التالي :

```
>> sin(30)
```

```
ans =
```

```
-0.9880
```

```
OR
```

```
>> sind(30*(180/pi))
```

```
ans =
```



-0.9880

الطريقة الثالثة :- يتم تحويل الزاوية في برنامج MATLAB باستخدام احد الامرين التاليين :-
rad2deg -1 : يستخدم هذا الامر لتحويل الزاوية نصف قطرية (radians) الى الدرجات (degrees) .

Example:-

There are 180° in π radians :

>> anglout=rad2deg(pi)

anglout=

180

deg2rad -2 : يستخدم هذا الامر لتحويل الزاوية من الدرجات (degrees) الى القياس نصف القطري (radians) .

Example:-

Show that there are 2 radians in full circle :

>> 2*pi - deg2rad(360)

ans=

0

❖ دوال التدوير والبقية : Routing & Remainder functions

1-fix :- Round toward zero

دالة التقريب الى اقرب رقم صحيح الى الصفر (اهمال الجزء الكسري)

Ex:-

>> a = [-1.9 , -0.2 , 3.4 , 5.6 , 7.0 , 2.4 + 3.6i]

a =

Columns 1 through 4

-1.9000 -0.2000 3.4000 5.6000

Columns 5 through 6

7.0000 2.4000 + 3.6000i

>> fix(a)

ans =

Columns 1 through 4

-1.0000 0 3.0000 5.0000

Columns 5 through 6

7.0 2.0000 + 3.0000i

2-round :- Round to nearest integer

دالة التقريب الى اقرب عدد صحيح .

Ex:-

>> a = [-1.9, -0.2, 3.4, 5.6, 7.0, 2.4+3.6i]

a =



```
Columns 1 through 4
-1.9000    -0.2000    3.4000    5.6000
Columns 5 through 6
7.0000    2.4000 + 3.6000i
>>round(a)
ans =
Columns 1 through 4
-2.0000    0    3.0000    6.0000
Columns 5 through 6
7.0000    2.0000 + 4.0000i
```

3- floor :- Round toward negative infinity

دالة التقريب باتجاه اللانهاية السالبة ($-\infty$) , دالة التقريب نحو اليسار .

Ex:-

```
>>a = [-1.9, -0.2, 3.4, 5.6, 7.0, 2.4+3.6i]
a =
Columns 1 through 4
-1.9000    -0.2000    3.4000    5.6000
Columns 5 through 6
7.0000    2.4000 + 3.6000i
>>floor(a)
ans =
Columns 1 through 4
-2.0000    -1.0000    3.0000    5.0000
Columns 5 through 6
7.0000    2.0000 + 3.0000i
```

4- ceil :- Round toward positive infinity

دالة التقريب باتجاه اللانهاية الموجبة ($+\infty$) , دالة التقريب نحو اليمين .

Ex:-

```
>>a = [ -1.9 , -0.2 , 3.4 , 5.6 , 7 , 2.4 + 3.6i ]
a =
Columns 1 through 4
-1.9000    -0.2000    3.4000    5.6000
Columns 5 through 6
7.0000    2.4000 + 3.6000i
>>ceil(a)
ans =
Columns 1 through 4
-1.0000    0    4.0000    6.0000
```



Columns 5 through 6

7.0000 3.0000 + 4.0000i

5- **rem** :- Remainder after division

دالة الباقي من القسمة .

 $R = \text{rem}(X,Y)$ if $Y \neq 0$, returns $X - (n.*Y)$ where $n = \text{fix}(X./Y)$

Ex:-

>> rem(8,5)

ans =

3

>> rem(8,8)

ans =

0

>> rem(8,10)

ans =

8

>> rem(-1,20)

ans =

-1

6- **mod** :- Modulus after division

دالة الباقي من القسمة .

 $M = \text{mod}(X,Y)$ if $Y \neq 0$, returns $X - (n.*Y)$ where $n = \text{floor}(X./Y)$

Ex:-

>> mod(13,5)

ans =

3

>> mod([1:5],3)

ans =

1 2 0 1 2

>> mod(magic(3),3)

ans =

2 1 0

0 2 1

1 0 2

Note:- magic(n) returns an n-by-n matrix constructed from the integers 1 through n^2 with equal row and column sums .

>> magic(3)

ans =



8 1 6
3 5 7
4 9 2

Notes:-

- $\text{rem}(X,Y)$ for $X \sim Y$ and $Y \sim 0$ has the same sign as X .
- $\text{mod}(X,Y)$ for $X \sim Y$ and $Y \sim 0$ has the same sign as Y .
- $\text{rem}(X,Y)$ and $\text{mod}(X,Y)$ are equal if X and Y have the same sign, but differ by Y if X and Y have different signs.

Ex:-

```
>>rem(-5,2)
```

```
ans=
```

```
-1
```

```
>>mod(-5,2)
```

```
ans=
```

```
1
```

7-sign :- Signum function

دالة الاشارة :-

إذا كان الرقم المرسل اكبر من الصفر فان ناتج الدالة 1
إذا كان الرقم المرسل اصغر من الصفر فان ناتج لدالة -1
إذا كان الرقم المرسل مساوي الى الصفر فان ناتج الدالة 0

Ex:-

```
>> sign(0)
```

```
ans =
```

```
0
```

```
>> sign(-9)
```

```
ans =
```

```
-1
```

```
>> sign(0.1)
```

```
ans =
```

```
1
```

**Complex Numbers: (العقدية) المركبة**

تأخذ الاعداد المركبة صيغة واحدة وهي تواجد جزء للأعداد الحقيقية Real Numbers وجزء للأعداد التخيلية Imaginary Numbers وتكون على الصورة العامة التالية:

$$Z=X+Y*i$$

يعمل برنامج MATLAB على جراء العديد من العمليات على الاعداد المركبة مثل

1. ايجاد الجزء الحقيقي من العدد المركب (X) .
2. ايجاد الجزء الخيالي من العدد المركب (Y) .
3. ايجاد زاوية الطور ويتم الحصول عليها رياضيا من العلاقة التالية :

$$\text{angle} = \tan^{-1} \left(\frac{\text{Imaginary part}}{\text{Real part}} \right)$$

4. ايجاد القيمة المطلقة Absolute Value ويتم الحصول عليها من خلال العلاقة التالية

$$\text{Absolute Value} = \sqrt{x^2 + y^2}$$

ويمكن تلخيص الدوال التي تقوم بهذه العمليات الرياضية كما يلي بعد ادخال قيمة العدد المركب z :

```
>> z=2+4i
```

```
z =
```

```
2.0000 + 4.0000i
```

Example	Operation	Function in MATLAB form
>> real(z) ans = 2	تستخدم لإيجاد الجزء الحقيقي من العدد المركب z	real(z)
>> imag(z) ans = 4	تستخدم لإيجاد الجزء الخيالي من العدد المركب z	imag(z)
>> abs(z) ans = 4.4721	تستخدم لإيجاد القيمة المطلقة للعدد المركب z	abs(z)
>> angle(Z) ans = 1.1071 >> angle(Z) ans = 1.1071	تستخدم لإيجاد زاوية الطور phase angle للعدد المركب z مقدرة بالراديان radian	angle(z)

ملاحظة : لحساب قيمة زاوية الطور بالدرجات يجب تحويل التقدير من radian الى degree بضرب قيمة الزاوية بالمقدار $180/\pi$.

مثال / اكتب برنامج بلغة MATLAB لإيجاد ما يلي :-

1. الجزء الحقيقي .



2. الجزء الخيالي .
 3. القيمة المطلقة العدد المركب .
 4. زاوية الطور .
- للعدد المركب التالي :-

$$C = 5 \sqrt{-9} + 13$$

Ex:-Write MATLAB program to calculate the following:-

- 1-Real part
- 2-Imaging part
- 3-Absolute Value
- 4-Angle phase for complex number

$$C = 5 \sqrt{-9} + 13$$

Sol.

```
>> C = 5 * sqrt ( -9 ) + 13
```

```
C =
```

```
13.000 + 15.000 i
```

```
>> real ( C )
```

```
ans =
```

```
13
```

```
>> imag ( C )
```

```
ans =
```

```
15
```

```
>> angle ( C )
```

```
ans =
```

```
0.8567
```

```
>> angle ( C ) * 180 / pi
```

```
ans =
```

```
49.0856
```

OR

```
>> rad2deg ( angle ( C ) )
```

```
ans =
```

```
49.0856
```

```
>> abs ( C )
```

```
ans =
```

```
19.8494
```



اوامر الادخال والاخراج في برنامج MATLAB Input I/P & Output O/P Commands

اوامر الادخال : input

يطبع الامر input رسالة نصية للمستخدم على الشاشة كطلب إدخال بيانات عددية او حرفية وتعيينها الى متغير يعرفه المستخدم. ويستخدم الامر input على احدى الصورتين التاليتين :

اولا : ادخال بيانات عددية

```
X=input('displayed strings')
```

Displayed strings: هي مجموعة من الكلمات تمثل رسالة نصية يتم عرضها للمستخدم لتعبر عن القيمة التي سيقوم المستخدم بإدخالها ، X هو المتغير الذي يتم ادخال قيمته.

```
>> x=input('x=');
```

يبقى المؤشر في الانتظار
لحين ادخال قيمة X من قبل
المستخدم .

```
x=
```

```
x =5
```

ملاحظة: تستخدم عبارة الادخال هذه عوضا عن الطريقة السابقة للإدخال المباشر للمتغيرات في برنامج MATLAB للتحكم بالقيم المعطاة عند كل تنفيذ للبرنامج، طريقة الادخال المباشر سوف تعطي قيمة ثابتة لا يمكن تغييرها ولكن باستخدام طريقة الادخال هذه يمكن اعطاء قيم اخرى.

مثال: لإدخال درجة الحرارة وعرض رسالة نصية لتدل على ان المتغير المدخل هو درجة الحرارة

```
>> T=input('Enter the temperature')
```

```
Enter the temperature
```

وبعد ظهور الجملة أعلاه قم بإدخال قيمة T الذي يمثل درجة الحرارة ولتكن 12

```
T =
```

```
12
```

ثانيا : ادخال بيانات رمزية

```
X=input('displayed strings','s')
```

تستخدم هذه الصورة لاستقبال سلسلة حرفية يقوم المستخدم بإدخالها، حيث يستخدم الحرف 's' الذي يرمز للكلمة (string) ويفيد تحديد نوع البيانات في تحديد نطاق التخزين المستغل من الذاكرة المؤقتة للبرنامج مما يساعد على عدم اهدار الذاكرة المؤقتة للبرنامج.

مثال : نريد عرض رسالة نصية لتدل على ان درجة الحرارة مقاسة بالمقياس السيليزي

```
>> T=input('enter the measure of temperature ','s')
```

```
enter the measure of temperature
```

وبعد ظهور الجملة اعلاه قم بإدخال المقياس وليكن المقياس السيليزي Celsius

```
T =
```

```
Celsius
```

**اوامر الاخراج : disp/display/fprintf**

تستخدم اوامر الاخراج لعرض قيم واسماء المتغيرات او التعبيرات النصية في نافذة محرر الاوامر Command window .

1. **الامر disp** : يستخدم الامر disp في عرض قيمة المتغير فقط سواء كانت عددية او نصية ،ويستخدم هذا الامر على احدى الصورتين :

disp(x)

disp('displayed strings')

حيث يستخدم الامر الاول لعرض قيمة المتغير X بينما يستخدم الامر الثاني لعرض تعبير نصي معين يتم ادخاله بين علامتي اقتباس single quotation marks .

مثال : لعرض قيمة عددية

```
>> x=100;
```

```
>> disp(x)
```

```
100
```

OR

او يكتب اسم المتغير مباشرة

```
>> x
```

```
x =
```

```
100
```

OR

لعرض تعبير نصي

```
>> disp(' the value of x is ')
```

```
the value of x is
```

ويمكن وضع القيمة العددية والتعبير النصي معا بالشكل التالي :

```
disp(['messege',num2str(variable)])
```

حيث تستخدم الدالة num2str والتي تعني numerical to string اي تحويل القيمة العددية الى سلاسل حرفية ، وتستخدم في اعطاء القيمة العددية بعد الرسالة (التعبير النصي) .

```
>> disp(['the value of x is ',num2str(x)])
```

```
the value of x is 100
```

2. **الامر display** : يستخدم الامر display في عرض اسم المتغير ثم قيمته سواء كانت رقمية او نصية ، ويستخدم على الصورة التالية :

display(x)

حيث يستخدم هذا الامر لعرض اسم المتغير x ثم قيمته على نافذة command window ، ولفهم الفرق الواضح بين الصورتين السابقتين لأوامر الاخراج لاحظ المثالين التاليين :

```
>> x=5;
```

```
>> disp(x)
```

```
5
```

%display only variable value

```
>> display(x)
```

%display variable name and value

```
x =
```

```
5
```



مثال/ مصفوفة الوحدة (التي جميع عناصرها مكونة من رقم 1) لاحظ الفرق في طريقة عرض النتائج:

```
>> disp(ones(3))
```

```
1 1 1
1 1 1
1 1 1
```

```
>> display(ones(3))
```

```
ans =
```

```
1 1 1
1 1 1
1 1 1
```

نلاحظ انه عندما استخدم الامر `disp(ones(3))` قام البرنامج بعرض قيم عناصر المصفوفة الناتجة فقط ، اما عندما استخدم الامر `display(ones(3))` قام البرنامج بعرض اسم المتغير الافتراضي `ans` (لان المستخدم لم يقوم بتعيين متغير معين لتخزين المصفوفة الناتجة عن `ones(3)` ثم يقوم البرنامج بعرض قيم عناصر المصفوفة الناتجة .

3. الامر `fprintf` (File Print Format) :-

يستخدم هذا الامر لتنسيق طباعة النتائج على نافذة `command window` ، الحرف `f` في بداية الكلمة `fprintf` يخص التنسيق `format` حيث يمكننا اختيار الطريقة المناسبة لتنسيق البيانات لكي تسهل قراءتها .

❖ في حالة طباعة تعبير نصي فان الامر يكتب بالصيغة التالية `fprintf('text')` مثال :

```
>> fprintf('the amount of precipitation is')
```

```
the amount of precipitation is
```

❖ اما في حالة طباعة تعبير نصي وقيمة لمتغير فان الامر يكتب بالصيغة التالية :-

`fprintf('format string', list of variable)`

ونعني بكلمة `format` هنا تنسيق البيانات ، اما `variable` فهي القيمة العددية للمتغير . لتوضيح التنسيق `format` تستخدم الصيغة التالية :

`fprintf('text % -3.1g', variable)`

text : يمثل النص المراد طباعته .
% : تمثل بداية تغيير تنسيق الرقم ، ويجب ملاحظة ان هذا الرمز هنا ليس للتعليقات كما تم توضيحه سابقا وانما يجب ان يكتب لكل متغير يراد طباعته في هذه الجملة .
- الاشارة : يمكننا من التحكم بتنسيق المخرجات ، كما موضح في الجدول التالي :

الرمز	المعنى	صيغة MATLAB	الامر وناتج التنفيذ
'-'	محاذاة نحو اليسار اي %-4g for 1 results in 1xxx اي ان العدد سيكون الى اليسار وعلى يمين العدد سيكون الفراغ الذي يرمز هنا بالرمز x	%-5.2f	>> fprintf('%-5.2f',9) 9.00 >>
'+'	يطبع اشارة العدد سواء كانت + او -	%+5.2f	>> fprintf('%+5.2f',9) +9.00>>
' '	يترك فراغ قبل طباعة العدد	% 5.2f	>> fprintf('% 5.2f',9)



9.00>>			
>> fprintf('%05.2f',9) 09.00>>	%05.2f	يطبع اصفار بدلا من الفراغات	'0'

- 3 : يمثل عرض الحقل ويمثل اقل عدد يمكن طباعته
 1. : يمثل عدد المراتب بعد الفارزة.
 g : تمثل الصيغة الرياضية التي ستستخدم لتغيير صيغة الرقم كما في الجدول التالي :

الصيغة	الوصف	مثال /
%d Or %i	يطبع العدد كاملا فيطبعه بدلالة الدالة الاسية اذا كان عشريا ويطبعه كعدد صحيح اذا كان صحيحا	>>a=5.5; >> fprintf('%d',a) 5.500000e+000 OR 5.000000>> fprintf('%d',5) 5
%e	صيغة اسية باستخدام حالة الاحرف الصغيرة e	>> fprintf('%e',a) 5.500000e+000
%E	صيغة اسية باستخدام حالة الاحرف الكبيرة E	>> fprintf('%E',a) 5.500000E+000
%f	صيغة العدد الحقيقي (العشري)	>> fprintf('%f',a) 5.500000
%g OR %G	صيغة بين f و e اكثر اختصارا ، تظهر العدد كما هو فيطبعه كعدد صحيح او عدد عشري من دون استخدام الدالة الاسية	>> fprintf('%g',a) 5.5 >> fprintf('%G',a) 5.5
%s	يطبع سلسلة حرفية	>>n='MATLAB'; >> fprintf('%s \n',n) MATLAB

مثال :

```
>> fprintf('the amount of precipitation is %g',0.6)
```

```
the amount of precipitation is 0.6
```

تم طباعة التعبير النصي the amount of precipitation is ثم وضعنا العلامة % التي يجب ان تستخدم لتحديد صيغة العدد الذي سوف يستخدم لاحقا والمتمثلة بالرمز g ، يجب ان يكون كل ذلك بين علامتي اقتباس '' ، ثم وضعنا قيمة المتغير والتي تساوي 0.6 ويمكن ادخال قيمة المتغير مسبقا تحت اسم ما ووضعه بدلا من قيمة الرقم .



❖ تستخدم الصيغ التالية لتنسيق المخرجات مع الامر fprintf كما في الجدول التالي :-

الصيغة	الوظيفة	الامر بصيغة MATLAB	نتائج التنفيذ
\n	يذهب لبداية سطر جديد	fprintf('hello') fprintf('\n') fprintf('bye') or >> fprintf('hello \n bye')	hello bye
\t	يترك مسافة افقية مساوية لـ Tab	fprintf('hello') fprintf('\t') fprintf('bye') or >> fprintf('hello \t bye')	hello bye
\\	يطبع الشكل \	fprintf('hello') fprintf('\\') fprintf('bye') or >> fprintf('hello \\ bye')	hello\bye
\%	يطبع الشكل %	fprintf('hello') fprintf('%') fprintf('bye') >> fprintf('hello \% bye')	Hello%bye

❖ بعض الامثلة المختلفة في طريقة اظهار النتائج :-

Program	Results
clc a=[12 55.5 43]; fprintf('the format for d') fprintf('\n') fprintf('%d \t',a)	the format for d 12 5.550000e+001 43
clc a=[12 55.5 43]; fprintf('the format for 2.2d') fprintf('\n') fprintf('%2.2d \t',a)	the format for 2.2d 12 5.55e+001 43
clc a=[12 55.5 43]; fprintf('the format for e') fprintf('\n') fprintf('%e \t',a)	the format for e 1.200000e+001 5.550000e+001 4.300000e+001
clc a=[12 55.5 43];	the format for 2.2e 1.20e+001 5.55e+001 4.30e+001



fprintf('the format for 2.2e') fprintf('\n') fprintf('%2.2e \t',a)	
clc a=[12 55.5 43]; fprintf('the format for f') fprintf('\n') fprintf('%f \t',a)	the format for f 12.000000 55.500000 43.000000
clc a=[12 55.5 43]; fprintf('the format for 2.2f') fprintf('\n') fprintf('%2.2f \t',a)	the format for 2.2f 12.00 55.50 43.00
clc a=[12 55.5 43]; fprintf('the format for g') fprintf('\n') fprintf('%g \t',a)	the format for g 12 55.5 43
clc a=[12 55.5 43]; fprintf('the format for 2.2g') fprintf('\n') fprintf('%2.2g \t',a)	the format for 2.2g 12 56 43 يقرب النتائج لأقرب عدد صحيح

❖ جد ناتج تنفيذ البرامج التالية :-

command	Results
fprintf('%d %f %g %2.2e\t',5.5,5.5,5.5,5.5)	5.500000e+000 5.500000 5.5 5.500000e+000
x=97.5; fprintf('it "works"%g %%of the time\n',x)	it 'works'97.5 %of the time
x1=10; x2=2; x3=x1-x2; fprintf('Difference of %g and %g is %g \n',[x1 x2 x3])	Difference of 10 and 2 is 8
matrix=[1 2 3;4 5 6;7 8 9]; fprintf(' +---+---+---+\n'); fprintf(' %g %g %g \n +---+---+---+\n',matrix);	+---+---+---+ 1 2 3 +---+---+---+ 4 5 6 +---+---+---+ 7 8 9 +---+---+---+



❖ طباعة المخرجات الى ملف لحفظ البيانات باستخدام الامر fprintf :-

يستخدم الامر fprintf لحفظ مخرجات البرنامج بانشاء ملف يمكن استدعاه فيما بعد بدون تنفيذ البرنامج مرة اخرى ، الصيغة العامة للامر تكتب بالشكل التالي :-

`fprintf ( ' filename' , ' format string ' , list of variables )`

e.g.

`fprintf ( ' myfile ' , '%g ' , x )`

يرسل القيمة x الى ملف اسمه myfile .

Ex:-

```
>>x=0:0.1:1;
>>A=[x ; exp(x)];
>> fileID=fopen('exp.txt','w');
>> fprintf(fileID,'%6s%12s\n','x','exp(x)');
>> fprintf(fileID,'%6.2f%12.8f\n',A);
>> fclose(fileID);
>> type exp.txt
```

x	exp(x)
0.00	1.00000000
0.10	1.10517092
0.20	1.22140276
0.30	1.34985881
0.40	1.49182470
0.50	1.64872127
0.60	1.82211880
0.70	2.01375271
0.80	2.22554093
0.90	2.45960311
1.00	2.71828183



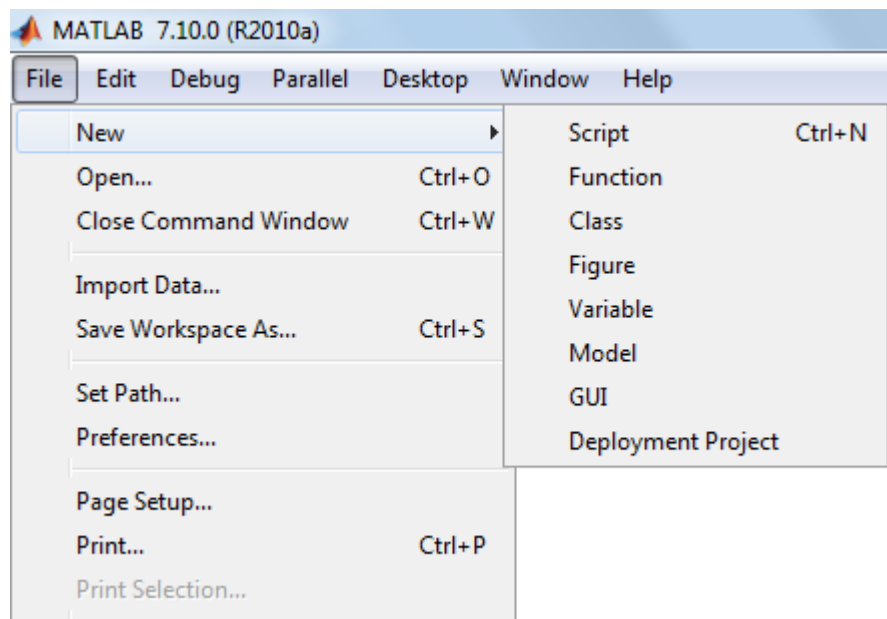
ملفات M النصية

Script M – Files

ملفات M – Files :- هي نوع من انواع الملفات النصية Script Files التي يعمل عليها برنامج MATLAB كوسيلة لإدخال الاوامر والرموز البرمجية ، حيث يتم تحرير اوامر البرنامج في ملف نصي Script File (يسمى هذا الملف "M-File") .

انشاء ملف M – File جديد:-

هناك ثلاث طرق لإنشاء ملف جديد لكتابة برنامج MATLAB هي: -
الطريقة الاولى: -من قائمة File اختر الامر New حيث تظهر قائمة فرعية اختر منها الامر Script في برنامج MATLAB 2010، او الامر M-File في MATLAB 7 كما في الشكل التالي: -



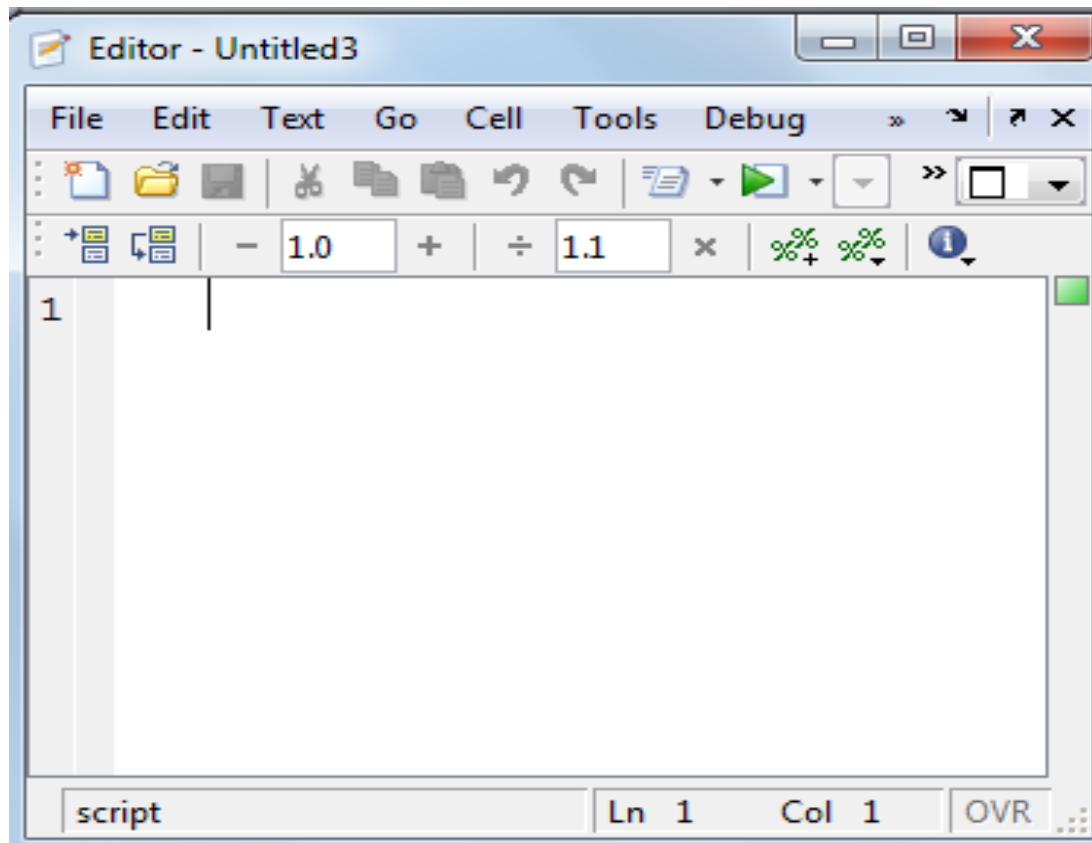
الطريقة الثانية :- اضغط على ايقونة الامر New M-File والتي لها شكل ورقة بيضاء والموجودة في شريط الادوات Tool Bar ، كما في الشكل التالي :-



الطريقة الثالثة :- اكتب الامر edit داخل نافذة الاوامر Command Window كما يلي :-

```
>> edit
```

وباتباع احدى الطرق الثلاثة السابقة سوف تظهر نافذة جديدة ، تأخذ الشكل التالي :-



سوف يحدد برنامج MATLAB اسما افتراضيا لهذا الملف هو Untitled وعند حفظ هذا الملف يعمل برنامج MATLAB على اضافة الامتداد (*. m) الى اسم هذا الملف .

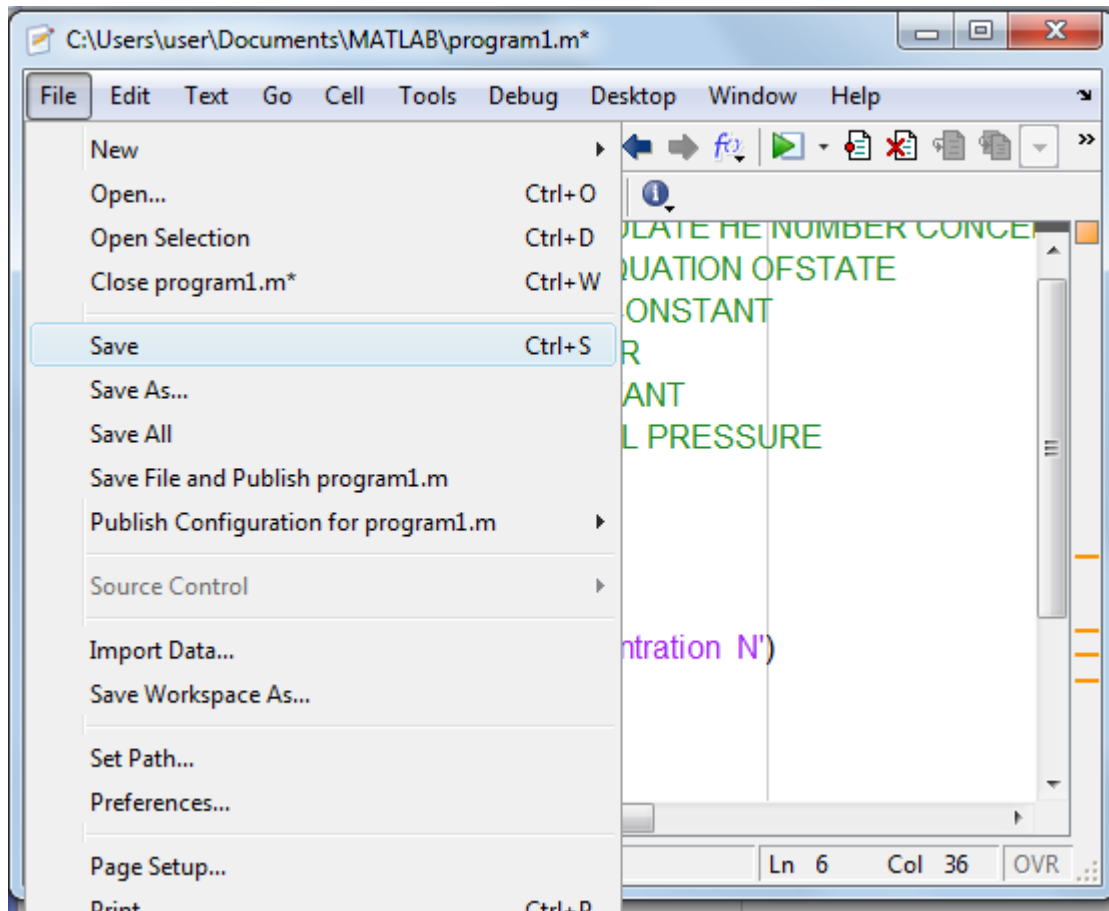
شروط حفظ ملف M-File :-

- (1) يجب ان يبدأ اسم الملف بحرف وليس برقم او برمز فمثلا لا يمكن كتابة اسم الملف على الشكل test.m وبدلا من ذلك يمكننا اسم الملف على الشكل test1.m
- (2) يجب ان لا يسمى اسم الملف على اسم امرا معروفا او دالة مبنية داخل برنامج MATLAB فمثلا لا يمكن تسمية الملف بالكلمة if لان هذا الاسم يمثل احدى الدوال الداخلية Built in functions للبرنامج.
- (3) يجب ان لا يحتوي اسم الملف على مسافات فاصلة Space فمثلا لا يمكن كتابة اسم الملف على الشكل التالي test a وبدلا من ذلك يمكن استخدام علامة الشارحة التحتية under score (_) على الشكل التالي test_a .
- (4) يجب ان لا يحتوي اسم الملف على بعض الرموز الخاصة مثل * , \ , / , ، ، ! ، ؟ باستثناء علامة الشارحة التحتية (_) under score حيث يمكن استخدامها سابقا

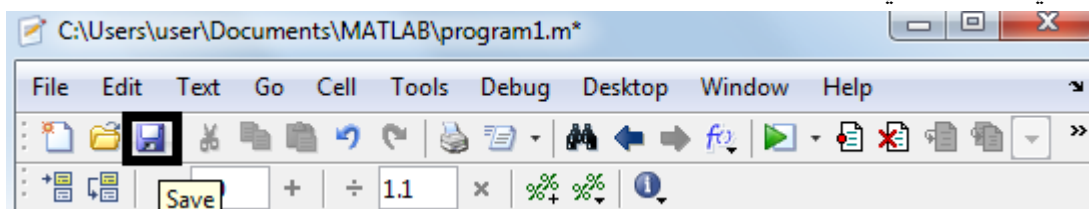
حفظ ملف M – File :-

يحفظ ملف برنامج MATLAB المكتوب في صفحة منقح البرامج Editor باتباع احدى الطريقتين التالية :-

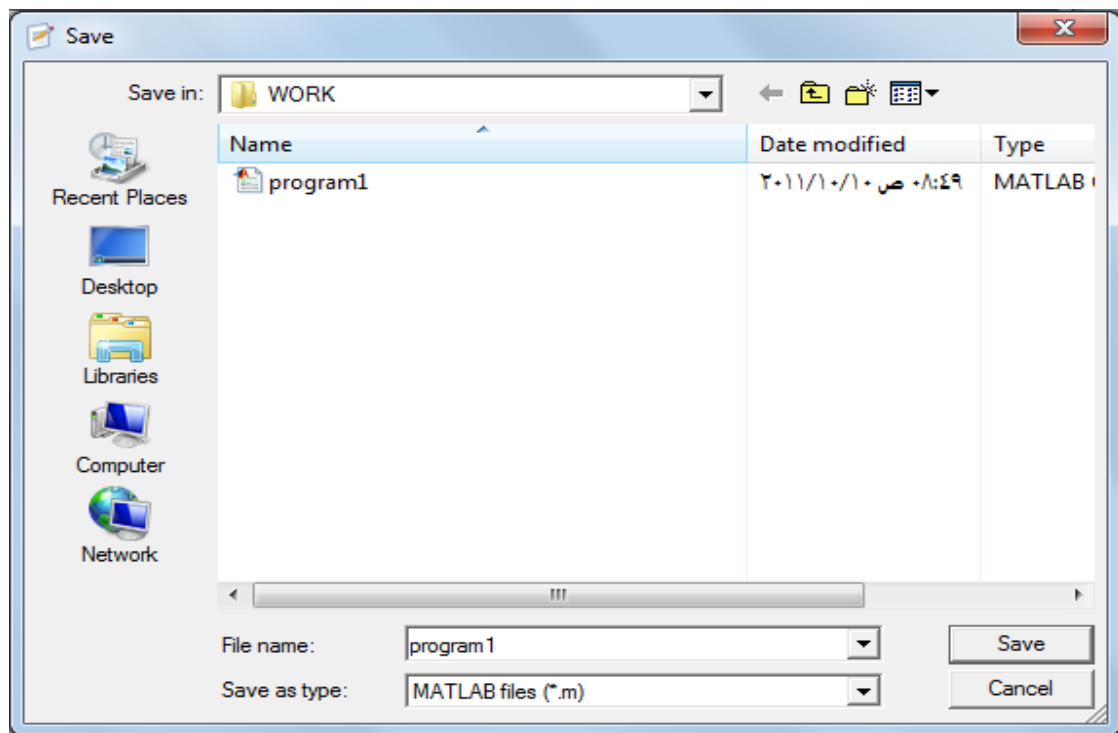
الطريقة الاولى :- اذهب الى القائمة File اختر منها امر الحفظ Save او اضغط على مفتاحي Ctrl+S من لوحة المفاتيح keyboard ، او اختر الامر حفظ باسم Save As ايضا من قائمة File لحفظ نسخة اخرى من الملف ، او اختر امر حفظ الكل Save All الموجود ضمن قائمة File لحفظ جميع الملفات المفتوحة حاليا كما في الشكل التالي :-



الطريقة الثانية :- اضغط على ايقونة امر الحفظ Save الموجودة في شريط الادوات Tool Bar ، كما في الشكل التالي :-



وعند حفظ ملف M – File المتضمن كود البرنامج باستخدام اي من الطرق السابقة يظهر مربع الحوار Save file as لاختيار مسار حفظ الملف M – File (حيث ان المسار الافتراضي لحفظ جميع انواع ملفات البرنامج هو C:\MATLAB\WORK) ، كما في الشكل التالي :-

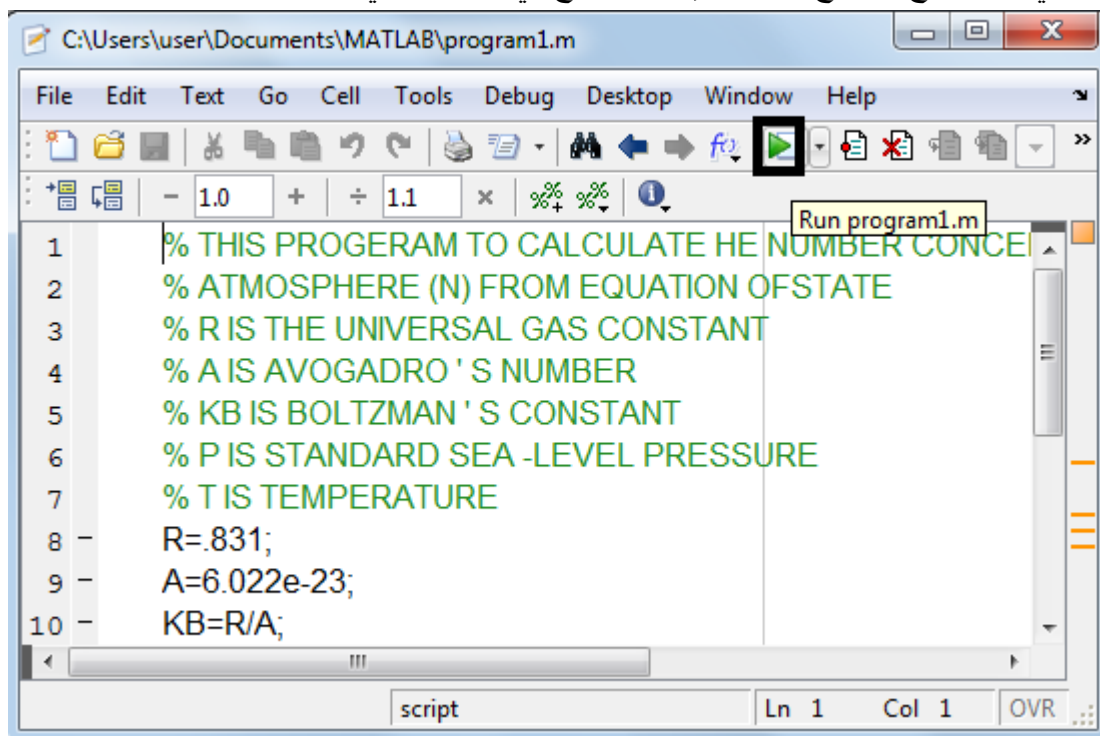


اكتب اسم الملف المراد حفظه في حقل File name ، مع مراعاة شروط اختيار اسم الملف السابق ذكرها ، كما نلاحظ ان الملف المحفوظ يأخذ الامتداد (*. m) .

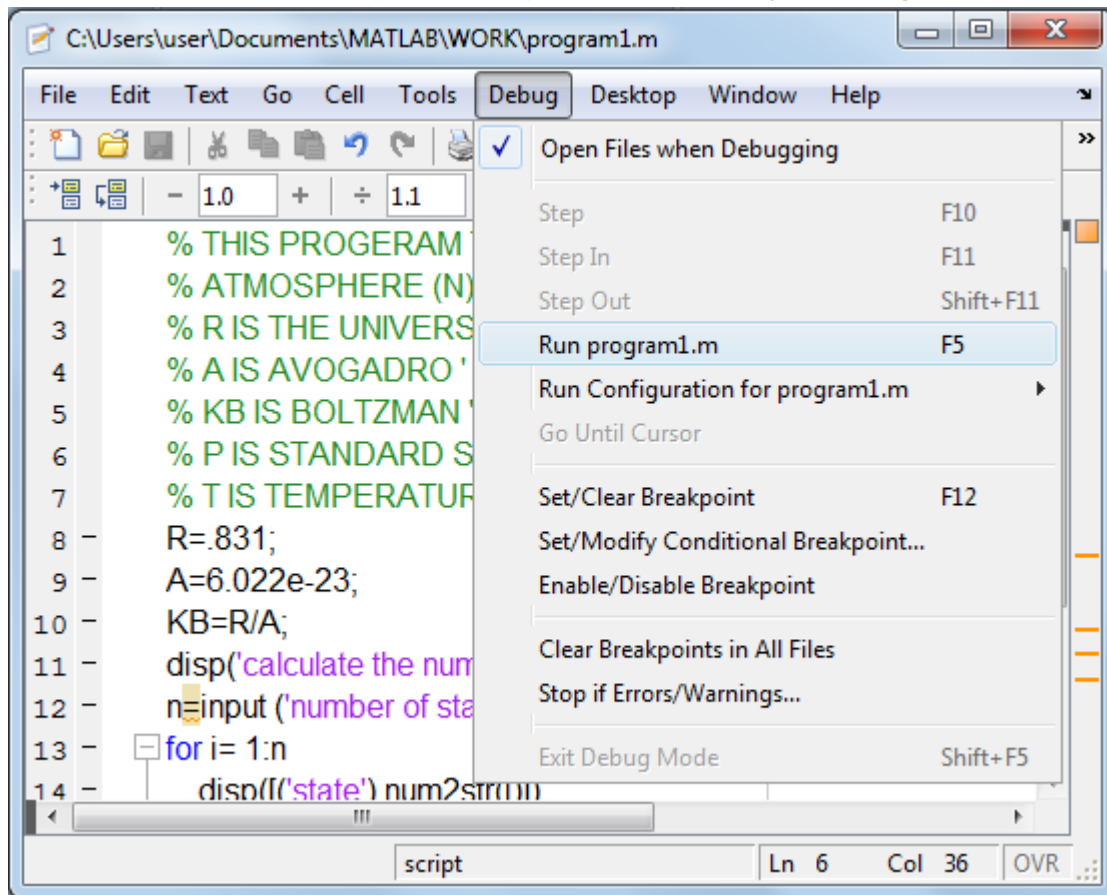
تشغيل البرنامج :-

يتم تشغيل برنامج MATLAB المكتوب داخل ملف M – File باتباع احدى الطرق الثلاثة التالية :-

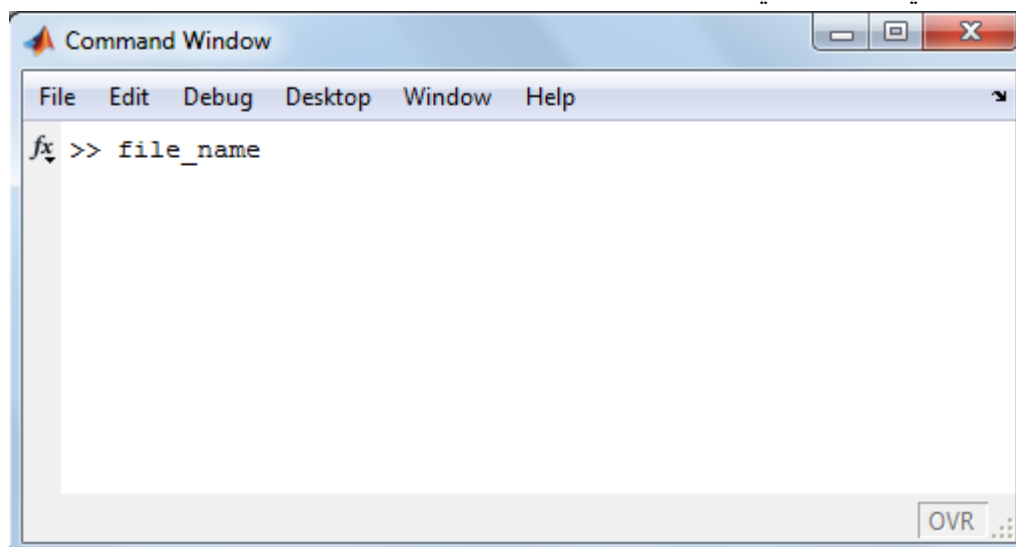
الطريقة الاولى :- اضغط على ايقونة زر التشغيل Run  الموجودة في شريط الادوات Tool Bar في نافذة منقح البرامج Editor , كما موضح في الشكل التالي :-



الطريقة الثانية :- اذهب الى القائمة Debug اختر الامر Run file_name او اضغط على المفتاح F5 من لوحة المفاتيح Keyboard مباشرة ، كما في الشكل التالي :-



الطريقة الثالثة :- تستخدم هذه الطريق لتشغيل اي ملف برنامج MATLAB مباشرة من نافذة الاوامر Command Window بكتابة اسم الملف في نافذة الاوامر دون الحاجة لفتح نافذة محرر البرامج Editor كما في الشكل التالي :-

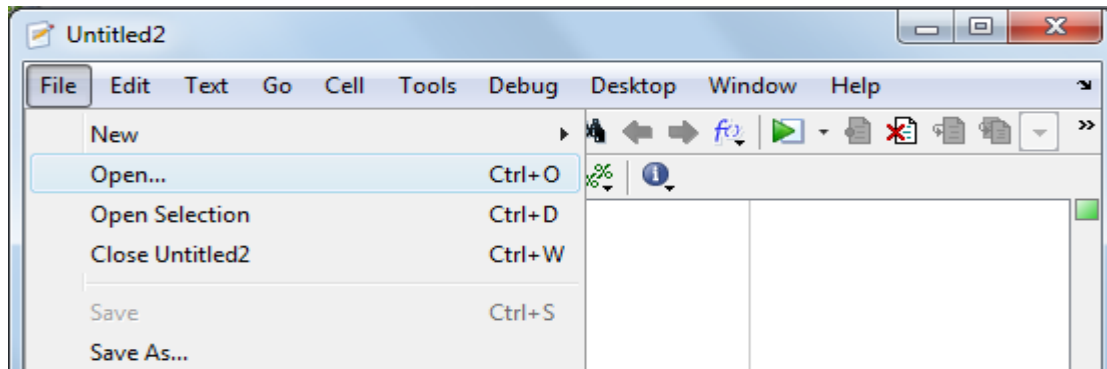




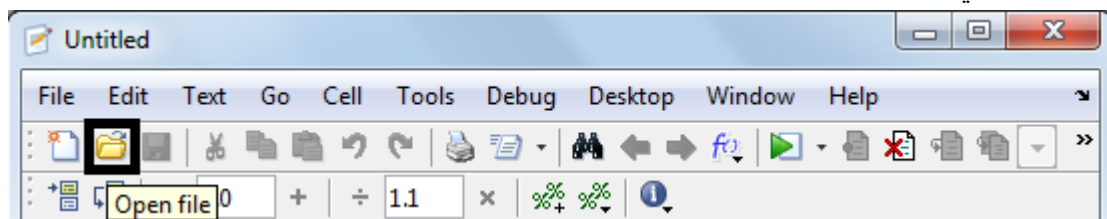
فتح ملف File – M سبق حفظه :-

تستخدم احدى الطرق الثلاثة التالية لفتح open ملف برنامج MATLAB (M – File) سبق حفظه :-

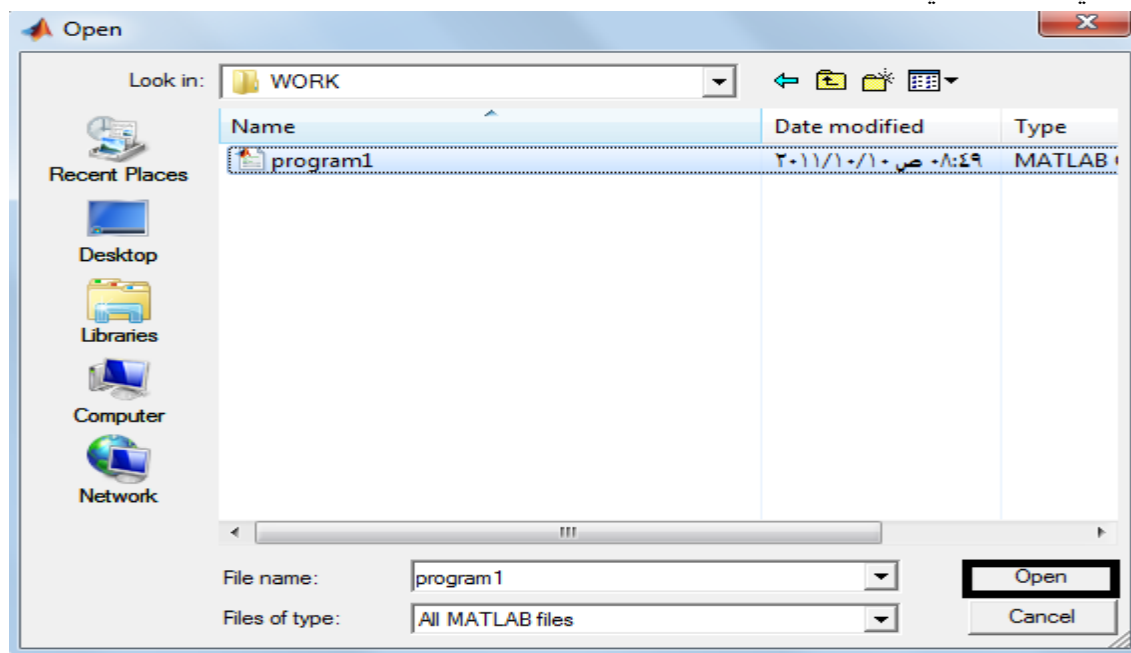
الطريقة الاولى :- اذهب الى القائمة File اختر امر الفتح Open من نافذة سطح مكتب برنامج MATLAB Desktop او من نافذة منقح البرامج Editor او بالضغط على مفتاحي Ctrl+O من لوحة المفاتيح Keyboard ، كما في الشكل التالي :-



الطريقة الثانية :- اضغط على ايقونة الامر Open File  الظاهرة في شريط الادوات Tool Bar على سطح مكتب برنامج MATLAB Desktop او من نافذة منقح البرامج Editor ، كما في الشكل التالي :-



سوف يظهر لك مربع الحوار Open لاختيار اسم الملف المراد فتحه ثم اضغط على مفتاح Open كما في الشكل التالي :-





الطريقة الثالثة :- استخدام احد الامرين التاليين :-

>>open file_name

Or

>>edit file_name

من خلال نافذة الاوامر Command Window اكتب احد الامرين السابقين متبوعا باسم الملف المحفوظ مسبقا .

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 لوحدتها 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 :-

الوظيفة Operation	Logic Operators المعاملات المنطقية
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 أولا 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 السنتيمتر .

```
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 أخرى

**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 .

مثال 1 / اكتب برنامج بلغة MATLAB لحساب المعادلات التالية :-

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

s= 0 ;

for n = 1 : 5000

s = s + n ;

end

disp (s)

sol.



```
>>file_name  
12502500
```

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

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

```
>>file_name  
41.679
```

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

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

```
>>file_name  
1.6447
```

مثال 2 / اكتب برنامج بلغة MATLAB لإيجاد مفاكوك 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 ويزيادة 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- $1^2 + 2^2 + 3^2 + \dots + 1000^2$

2- $1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$

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

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

s = 0 ;

for k = 1 : 1000

s = s + k ^ 2 ;

end

display (s)

نتائج تنفيذ البرنامج

>>file_name

s = 338350

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

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

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 ,

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

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

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

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

sol.

% 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 $\omega=2$ 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
...
```

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` ووصولاً 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 .

**دوال ملفات M****M – File Function**

الدوال function :- الدوال هي مجموعة من الاوامر او الجمل البرمجية المكتوبة في ملفات M – File والتي تستخدم لتؤدي وظيفة معينة ولها اسم مميز يعبر عن وظيفتها لتؤدي امر او مجموعة من الاوامر الجاهزة .

انواع الدوال : function types

1. يحتوي برنامج MATLAB على مئات من الدوال الداخلية الجاهزة مبنية في بنية البرنامج **MATLAB Built in Function** بدلا من كتابة او برمجة هذه الدوال في كل مرة مثل `sum , prod , mean , inv , det , size , length , round , fix , rem , angle , abs` وغيرها من الدوال التي تم برمجتها مسبقا في برنامج MATLAB .
2. دوال يتم انشاءها من قبل المستخدم **User – Defined Function** تكتب بصيغة برنامج MATLAB ويسمى هذا النوع من البرمجة بدوال ملفات M (M – File Function) .
 - تضاف هذه الدوال بعد انشاءها الى مكتبة الدوال الداخلية الجاهزة (MATLAB Built in Function) ، ويتم تسميتها من قبل المستخدم بتصميمه لها حسب وظيفتها ، ويتم كتابة البرامج بصيغة **function** بالشكل التالي :-

`function [ list of output variables ] = function_name (list of input variables)`
`function (o/p) = function_name(i/p)`

OR

`function function_name (list of input variables)`
`function function_name(i/p)`

شروط تسمية وانشاء دوال ملفات M (M – File Function) :-

- 1- يسمى اسم الملف بنفس اسم الدالة التي تم كتابتها في البرنامج .
- 2- السطر الاول من البرنامج يجب ان يبدأ بالامر `function` .
- 3- يتبع اسم الدالة شروط تسمية المتغيرات في برنامج MATLAB .
- 4- لايجوز استخدام اسم الدالة من اسماء مشابهة لاسماء المتغيرات المستخدمة في البرنامج .

الفرق بين برمجة الدوال (function) والبرمجة النصية (script) :-

- 1- برنامج الدوال يبدأ بالامر `function` بينما برنامج الـ `script` يبدأ بالبرنامج مباشرة .
- 2- يتطلب كتابة برنامج الدوال `function` تعريف المخرجات والمدخلات مع اسم الدالة بينما في البرمجة النصية الـ `script` لا يحتاج الى ذلك .
- 3- تستطيع في برنامج الدوال `function` عند تنفيذه تغيير قيم المدخلات بينما في البرمجة النصية الـ `script` لا يمكن ذلك .



4- المتغيرات التي تحسب داخل الـ function لا تحفظ في الـ workspace بينما في الـ script تحفظ في الـ workspace ، وتسمى متغيرات محلية (local variable) .

معرفة عدد المدخلات والمخرجات للدوال (nargin ,nargout) :-

يستخدم الأمر nargin لمعرفة عدد المعاملات (المتغيرات) المستخدمة داخل الدالة بالشكل التالي

```
a = nargin ( ' function_name ')
```

ex :-

```
a = nargin ('sum')
```

```
a =
```

```
3
```

ويستخدم الأمر nargout لمعرفة عدد المعاملات (المتغيرات) المستخدمة كمخرجات لهذه الدالة وبالشكل التالي :-

```
a = nargout ( ' function_name ')
```

ex :-

```
a = nargout ( ' sum ')
```

```
a =
```

```
1
```

**بعض الامثلة :-**

مثال 1:- اكتب برنامج يحل معادلة من الدرجة الثانية بطريقة الدستور (Quadratic equation)

باستخدام Function M-file

$$x = \begin{cases} \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} & b^2 - 4ac > 0 \\ \frac{-b}{2a} & b^2 - 4ac = 0 \end{cases}$$

واذا كان قيمة تحت الجذر اقل من صفر اطبع العبارة (the root is complex)

Sol.

```
function quadratic_equation ( a , b , c )
```

```
delta = b^2 - 4 * a * c
```

```
if delta > 0
```

```
x1 = ( - b + sqrt ( delta )) / ( 2 * a )
```

```
x2 = ( - b - sqrt ( delta )) / ( 2 * a )
```

```
elseif delta < 0
```

```
disp ( ' the root is complex ')
```

```
else
```

```
x1_2=( - b / ( 2 * a ) )
```

```
end
```

نتائج تنفيذ البرنامج

```
>> quadratic_equation ( 4 , 6 , 2 )
```

```
delta =
```

```
4
```

```
x1 =
```

```
-0.5000
```



$$x^2 =$$

-1

تمرين للطالب ، نفذ البرنامج لقيم اخرى لكل من a , b , c .

مثال 2 :- اكتب برنامج بلغة MATLAB لايجاد القيمة العظمى من بين ثلاث قيم ندخلها للبرنامج باستخدام M-file Function .

Sol.

```
function max1( a , b , c )
```

```
if a > b
```

```
max = a ;
```

```
    if c > max
```

```
        max = c ;
```

```
    end
```

```
else
```

```
max = b ;
```

```
    if c > max
```

```
        max = c ;
```

```
    end
```

```
end
```

```
max
```

لتنفيذ البرنامج اكتب اسم البرنامج مع ادخال القيم مع اسم البرنامج

```
>> max1 ( 10 , 2 , 18 )
```

```
max =
```

```
18
```

تمرين للطالب ، ادخل قيم اخرى مع هذه الدالة .



مثال 3 :- اكتب برنامج بلغة MATLAB لادخال رقم واختبار هل هو زوجي ام فردي .

Sol.

```
function n= test_number( a )
```

```
if mod ( a , 2 ) == 0
```

```
n = ' even no.';
```

```
else
```

```
n = ' odd no.';
```

```
end
```

لتنفيذ البرنامج ، اكتب اسم البرنامج مع اعطاء قيمة بين القوسين :-

```
>> test_number ( 4 )
```

```
even no.
```

تمرين للطالب ، ادخل قيم اخرى .

مثال 4 :- اكتب برنامج بلغة MATLAB لايجاد مضروب العدد .

Sol.

```
function c = fact ( n )
```

```
v = 1 ;
```

```
for i = 1 : n
```

```
v = v * i ;
```

```
end
```

```
v
```

لتنفيذ البرنامج ، اكتب اسم البرنامج مع اعطاء قيمة للعدد المطلوب ايجاد قيمة المضروب له :-

```
>> fact ( 10 )
```

```
v =
```

```
3628800
```



تمرين للطالب ، ادخل قيم اخرى .

مثال 5 :- اكتب برنامج بلغة MATLAB لظهار جدول الضرب لاي رقم .

Sol.

```
function production ( num )  
if length ( num ) ~= 1 | ~ isnumeric ( num )  
disp ( ' error : please enter one number ' )  
else  
for i = 1 : 10  
disp ( [ num2str ( num ) , ' x ' , num2str ( i ) , ' = ' , num2str ( num * i ) ] )  
end  
end
```

ملاحظة :- 1- يفحص البرنامج اذا كان الرقم المدخل هو ليس عنصر واحد او قيم ليست عددية ، فيطبع العبارة التصحيحية لتبين عدم ادخال صحيح .

2- يعمل الامر disp على طباع مصفوفة مكونة من 5 عناصر يجب تحويل كل عنصر من عناصرها الى نصوص (سلسلة حرفية string) , لايمكن استخدام مصفوفة مكونة من ارقام و حروف ، يجب ان تكون المصفوفة اما حروف لوحدها او ارقام لوحدها :-

```
[ num2str ( num ) , ' x ' , num2str ( i ) , ' = ' , num2str ( num * i ) ]  
ex :-          1          x          1          =          1  
لتنفيذ البرنامج اكتب اسم الدالة في الـ command window وادخل الرقم بين قوسين صغيرين
```

```
>> production([12 5])
```

```
error : please enter one number
```

```
>>production('g')
```

```
error : please enter one number
```

يجب ادخال رقم واحد وليس مصفوفة او سلسلة حرفية

```
>> production(12)
```

```
12 x 1 = 12
```



$$12 \times 2 = 24$$

$$12 \times 3 = 36$$

$$12 \times 4 = 48$$

$$12 \times 5 = 60$$

$$12 \times 6 = 72$$

$$12 \times 7 = 84$$

$$12 \times 8 = 96$$

$$12 \times 9 = 108$$

$$12 \times 10 = 120$$

تمرين للطالب ، ادخل عدة قيم اخرى .



المصفوفات والمتجهات MATRICES AND VECTORS

طرق ادخال المتجهات في برنامج MATLAB :

1. الادخال المباشر للبيانات :

يتم ادخال البيانات في المتجهات الصفية row vector كالتالي :-

```
>> a=[1 2 3 4 5]
```

```
a =
```

```
1    2    3    4    5
```

Or

```
>> a=[1,2,3,4,5]
```

```
a =
```

```
1    2    3    4    5
```

اما للمتجهات العمودية column vector ، يكون ادخال البيانات كالتالي :-

```
>> b=[1;2;3;4;5]
```

```
b =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

OR

```
>> b=[1
```

```
2
```

```
3
```

```
4
```

```
5]
```

```
b =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

2. الادخال غير المباشر للبيانات :

تستخدم النقطتين المتعامدتين colon notation (:) في المتجهات الصفية row vector لادخال عناصر المتجه حسب الصيغة التالية :-

Name of array=first value:increment:last value

اي نبدأ بكتابة القيمة الاولى ثم مقدار الزيادة ثم القيمة النهائية . لاحظ الامثلة التالية :

لانشاء متجه صفى يبدأ ب 1 وينتهي ب 100 ومقدار الزيادة 1

```
>> a=1:100 or >> a=1:1:100 or >> a=(1:100)
```

لانشاء متجه صفى يبدأ ب 1 وينتهي ب 100 ومقدار الزيادة 2

```
>> a=1:2:100
```

لانشاء متجه يبدأ ب 100 وينتهي ب 1 وبمقدار التناقص 1-

```
>> a=100:-1:1
```

اما بالنسبة للمتجهات العمودية column vector فانه يمكن استخدام نفس الطريقة السابقة لانشاء

المتجهات الصفية ولكن يجب ايجاد مبدل المتجه الصفى row vector transpose باستخدام علامة

الفاصلة العليا apostrophe (') بعد اسم المتجه المراد الحصول على مبدله . لاحظ المثال التالي :

```
>> x=1:5
```

```
x =
```



```
1 2 3 4 5
>>x'
ans =
1
2
3
4
5
```

العمليات التي يمكن إجراؤها على المتجهات :

أولاً:- استدعاء عناصر المتجه :

لتحديد عنصر واحد من عناصر المتجه الصفّي / العمودي في لغة Matlab يتم استخدام الأقواس الصغيرة ، ولتحديد مجموعة من العناصر في خطوة واحدة نستخدم النقطتين المتعامدتين (:) وبالصيغة X(first:increment:last) ، حيث نكتب اسم المتجه ثم قوس صغير ثم تسلسل العدد الأول ، ومقدار الزيادة ، وتسلسل العدد الأخير ، كما في الأمثلة التالية :-

```
>>x=[1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 6 ; 7 ; 8 ; 9 ; 10 ] ;
```

1- استدعاء العنصر الأول من المتجه x

```
>>x(1)
ans=
1
```

2- استدعاء العنصر الخامس من المتجه x

```
>>x(5)
ans=
5
```

3- استدعاء العناصر ابتداء من العنصر الأول الى العنصر الخامس

```
>>x(1:5)
ans =
1
2
3
4
5
```

4- استدعاء العنصر الأخير في المتجه x

```
>>x(end)
ans=
10
```

5- استدعاء العنصر ما قبل الأخير في المتجه x

```
>>x(end-1)
ans=
9
```

6- استدعاء العناصر بداية من العنصر السادس ونهاية بالعنصر العاشر

```
>>x(6:10)
ans=
```



6
7
8
9
10

7- استدعاء العنصر الاول الى العنصر السادس بزيادة 2

```
>>x(1:2:6)
ans =
     1
     3
     5
```

8- استدعاء العنصر السادس الى العنصر الاول بمقدار 2-

```
>>x(6:-2:1)
ans =
     6
     4
     2
```

9- استدعاء العناصر من 7 الى نهاية المتجه x

```
>>x(7:end)
ans =
     7
     8
     9
    10
```

10- استدعاء عناصر معينة لا يشترط تسلسلها بخطوة واحدة

```
>>x([8 2 9 1])
ans =
     8
     2
     9
     1
```

ثانيا :- اضافة عنصر جديد الى المتجه :

لاضافة عنصر جديد للمتجه الصفّي او العمودي فيمكن ذلك من خلال كتابة اسم المتجه ونحدد تسلسل العنصر المراد اضافته بين قوسين صغيرين ثم علامة المساواة وقيمة العنصر كما في المثال التالي :

```
>> x=[1 2 3 4 5 6 7 8 9 10 ]
x =
     1     2     3     4     5     6     7     8     9    10
>>x(11)=11
x =
     1     2     3     4     5     6     7     8     9    10    11
```



لاحظ في حالة اختيار التسلسل الثالث عاشر (العنصر x(13)) فإنه سوف يعتبر القيم غير الموجودة والتي تقع بين x(11) و x(13) مساوية الى الصفر لاحظ المثال التالي :

```
>>x(13)=8
```

```
x =
```

```
1 2 3 4 5 6 7 8 9 10 11 0 8
```

```
>>x(4)=8
```

```
1 2 3 8 5 6 7 8 9 10 11 0 8
```

ثالثا :-دمج متجهين في متجه واحد :

لدمج متجهين صفي او عمودي معا في متجه واحد فيمكن استخدام الصيغة التالية :

Name of new vector=[name of first vector name of the second vector]

```
>>a=[1 3 5 7]
```

```
a =
```

```
1 3 5 7
```

```
>>b=[9 10]
```

```
b =
```

```
9 10
```

```
>>c=[a b]
```

```
c =
```

```
1 3 5 7 9 10
```

رابعا :-حذف عنصر/عدة عناصر في المتجهات :

تستخدم الاقواس المربعة [] في عملية الحذف وبالشكل التالي ، لحذف العنصر العاشر من المصفوفة a :-

```
>> a=[1 2 3 4 5 6 7 8 9 10]
```

```
a =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>>a(10)=[ ]
```

```
a =
```

```
1 2 3 4 5 6 7 8 9
```

● لحذف مجموعة متتالية من العناصر (مثلا من العنصر السادس الى العاشر)

```
>> a=[1 2 3 4 5 6 7 8 9 10]
```

```
a =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>>a(6:10)=[ ]
```

```
a =
```

```
1 2 3 4 5
```

خامسا :-ايجاد حاصل جمع عناصر المتجه باستخدام الامر sum :-

```
>> v=[4 5 3 2]
```

```
v =
```

```
4 5 3 2
```

```
>> v1=sum(v)
```

```
v1 =
```

```
14
```

**سادسا :- ايجاد حاصل ضرب عناصر المتجه باستخدام الامر prod :-**

```
>> v2=prod(v)
v2 =
120
```

سابعا :- ايجاد العنصر الاكبر في المتجه باستخدام الدالة max :-

```
>> a=[1 2 5 8 11]
a =
1 2 5 8 11
>> a1=max(a)
a1 =
11
```

ثامنا :- ايجاد العنصر الاصغر في المتجه باستخدام الدالة min :-

```
>> a2=min(a)
a2 =
1
```

طرق ادخال المصفوفات MATRIX في برنامج MATLAB

المصفوفة هي مجموعة من الارقام (القيم العددية) تكتب على شكل صفوف rows واعمدة columns

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

حيث m يمثل عدد الصفوف ، n عدد الاعمدة .

لادخال مصفوفة نستخدم الطريقة المباشرة لادخال المتجهات الصفية او العمودية ، وبالشكل التالي :-

```
>>a=[1 2 3;4 5 6;7 8 9]
```

Or

```
>>a=[1,2,3;4,5,6;7,8,9]
```

Or

```
>>a=[1 2 3
```

```
4 5 6
```

```
7 8 9]
```

```
a =
```

```
1 2 3
4 5 6
7 8 9
```

العمليات التي يمكن إجراؤها على المصفوفات :اولا:- استدعاء عنصر معين من عناصر المصفوفة :

لاستدعاء عنصر معين من عناصر المصفوفة نكتب اسم المصفوفة وقوس صغير ثم تسلسل الصف وتسلسل العمود ثم نغلق القوس الصغير . لاحظ المثال التالي :

```
>> a=[1 2 3;4 5 6;7 8 9]
```

```
a =
```

```
1 2 3
4 5 6
7 8 9
```

• لاستدعاء العنصر في الصف الثاني من العمود الثالث :

```
>>a(2,3)
```

```
ans =
```

```
6
```

• لتحديد مجموعة معينة من قيم عناصر المصفوفة (انشاء مصفوفة جزئية من المصفوفة الاصلية (تستخدم الصيغة التالية :

(start row: step : end row ,start column : step : end column)

اختيار الاعمدة → | ← اختيار الصفوف

مثال : لانشاء مصفوفة جديدة b من المصفوفة a بحيث يتم تحديد الصف الاول والثالث والعمود الثاني والثالث

```
>> b=a(1:2:3,2:3)
```

```
b =
```

```
2 3
8 9
```

• لاستدعاء الصف الثاني والثالث وجميع الاعمدة ابتداء من العمود الثالث وحتى العمود الاول

```
>> c=a(2:3,3:-1:1)
```

```
c =
```

```
6 5 4
9 8 7
```

• اختيار جميع الاعمدة : لتحديد الصف الثاني وجميع الاعمدة

```
>> d=a(2,:)
```

```
d =
```

```
4 5 6
```

• اختيار جميع الصفوف : لتحديد جميع الصفوف اما الاعمدة فتبدأ من العمود الاول وحتى الثاني

```
>> e=a(:,1:2)
```

```
e =
```

```
1 2
4 5
7 8
```

• اختيار جميع الصفوف والاعمدة

```
>> f=a(:,:)
```



```
f =
     1     2     3
     4     5     6
     7     8     9
```

- اختيار جميع الصفوف من العمود الأخير

```
>> g=a(:,end)
g =
     3
     6
     9
```

- اختيار الصف الأخير وجميع الأعمدة

```
>> h=a(end,:)
h =
     7     8     9
```

- ثانياً : إضافة عنصر إلى عناصر المصفوفة :
لإضافة عنصر إلى عناصر المصفوفة التالية

```
>> a=[1 23
      2 5 6
      7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
```

- لإضافة الرقم 20 في الصف الثالث والعمود الخامس حدد اسم المصفوفة ثم تسلسل العدد المراد إضافته يوضع بين قوسين صغيرين وعلامة المساواة ثم قيمة العدد . لاحظ تضاف الأصفار في أماكن الأعداد التي ليست لها قيمة محددة

```
>>a(3,5)=20
a =
     1     2     3     0     0
     4     5     6     0     0
     7     8     9     0    20
```

- إضافة عدة عناصر إلى المصفوفة مثلاً إضافة الأرقام 11 , 13 , 15 إلى عناصر العمود الرابع

```
>>a=[12 3;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
```

```
>>a(:,4)=[11 13 15]
a =
     1     2     3    11
```



```
4 5 6 13
7 9 8 15
```

- إضافة عمود ولكل الصفوف ، لإضافة عمود قيم عناصره 5 للمصفوفة a التالية :-

```
>> a=[1 5 3;2 9 7]
```

```
a =
```

```
1 5 3
2 9 7
```

```
>> a(:,end+1)=5
```

```
Or
```

```
>> a(:,4)=5
```

```
a =
```

```
1 5 3 5
2 9 7 5
```

ثالثا :-استبدال قيم عنصر او عدة عناصر من عناصر المصفوفة بقيم اخرى :

- تغيير قيمة عنصر في المصفوفة : استبدال العنصر الموجود في الصف الثالث والعمود الاول بالقيمة صفر

```
>> b=[1 3 7 2;3 4 6 1;7 9 8 4]
```

```
b =
```

```
1 3 7 2
3 4 6 1
7 9 8 4
```

```
>> b(3,1)=0
```

```
b =
```

```
1 3 7 2
3 4 6 1
0 9 8 4
```

- تغيير قيم صف بكامله : لجعل قيم الصف الثالث جميعها مساوية لـ 5

```
>> b(3,:)=5
```

```
b =
```

```
1 3 7 2
3 4 6 1
5 5 5 5
```

- تغيير قيم عمود بكامله : لجعل قيم العمود الثاني جميعها مساوية لـ 11

```
>> b=[1 3 7 2;3 4 6 1;7 9 8 4]
```

```
b =
```

```
1 3 7 2
3 4 6 1
7 9 8 4
```

```
>> b(:,2)=11
```



b =

1	11	7	2
3	11	6	1
7	11	8	4

- ولتغيير قيم عمود بكامله بقيم ادخال جديدة (يجب وضع الفارزة المنقوطة):

b =

1	11	7	2
3	11	6	1
7	11	8	4

>>b(:, 3) = [4 ; 5 ; 77]

b =

1	11	4	2
3	11	5	1
7	11	77	4

- استبدال العددين في العمودين الاول والثالث (دون الثاني) في الصف الاول بالعدد 10 :

>> c=[1 3 7;2 4 6;7 8 9]

c =

1	3	7
2	4	6
7	8	9

>>c(1,[1 3])=10

c =

10	3	10
2	4	6
7	8	9

- استبدال الاعداد في العمودين الاول والثالث (دون الثاني) لجميع الصفوف بالعدد 10 :

>> c=[1 3 7;2 4 6;7 8 9]

c =

1	3	7
2	4	6
7	8	9

>>c(:, [1 3])=10

c =

10	3	10
10	4	10
10	8	10

- استبدال العناصر في الصفين ابتداءً من الاول الى الثالث والعمودين ابتداءً من الاول الى الثاني بالقيمة صفر

>>b(1:3,1:2)=0

b =

0	0	4	2
0	0	5	1



0 0 77 4

رابعاً :- حذف عنصر او عدة عناصر من المصفوفات باستخدام الاقواس المربعة [] :

- لا يمكن حذف عنصر واحد فقط

```
>> b=[ 1 3 7 2;3 4 6 1;7 9 8 4]
```

```
b =
```

```
1 3 7 2
3 4 6 1
7 9 8 4
```

```
>>b(1,1)=[]
```

```
??? Indexed empty matrix assignment is not allowed.
```

- يمكن حذف صف كامل او عمود كامل (مثلا حذف العمود الثالث)

```
>>b(:,3)=[]
```

```
b =
```

```
1 3 2
3 4 1
7 9 4
```

او حذف الصف الثالث (لحذف صف معين حدد رقم السطر او الصف وضع (:) للاعمدة)

```
>>b(3,:)=[]
```

```
b =
```

```
1 3 7 2
3 4 6 1
```

خامساً :- معرفة ابعاد المصفوفة (حجم المصفوفة) (size) :

```
>> a=[1 2 3 4; 5 6 7 8 ]
```

```
a =
```

```
1 2 3 4
5 6 7 8
```

```
>>size(a)
```

```
ans =
```

```
2 4
```

عدد الصفوف

عدد الاعمدة

```
>>size(a,1)
```

لمعرفة عدد الصفوف فقط

```
ans =
```

```
2
```

```
>>size(a,2)
```

لمعرفة عدد الاعمدة فقط

```
ans =
```

```
4
```

سادساً :- لمعرفة طول المتجه الاكبر في المصفوفة ، باستخدام الامر length:

```
>> q=[1 2 ;3 4;5 6]
```

```
q =
```

```
1 2
3 4
```



```
5 6
>>size(q)
ans =
3 2
>>length(q)
ans =
3
```

سابعاً :- لظهار كل عناصر المصفوفة بشكل عمود واحد، نستخدم النقطتين المتعامدتين (:)

```
>>q(:)
ans =
1
3
5
2
4
6
```

ثامناً :- لمعرفة عدد عناصر المصفوفة ، ويكون ذلك باستخدام الامر numel:

```
>>numel(q)
ans =
6
Or
>>length(q(:))
ans =
6
```

تاسعاً :- لاستبدال عناصر الصفوف بدل الاعمدة او لتحويل المتجه العمودي الى صفى او العكس ، وذلك بوضع علامة اقتباس مفردة (') بعد اسم المصفوفة :

```
>> a=[1 2 3;4 5 6;7 8 9]
a =
1 2 3
4 5 6
7 8 9
>>a'
ans =
1 4 7
3 5 8
2 6 9
```

10:- ايجاد عناصر القطر الرئيسى للمصفوفة :

• تستخدم الدالة diag لايجاد عناصر القطر الرئيسى للمصفوفة على النحو التالي :

```
>> a= [ 1 2 3 ; 4 5 6 ; 7 8 9 ]
a =
1 2 3
4 5 6
```



```
7 8 9
>> b=diag(a)
b =
1
5
9
```

11. جمع عناصر المصفوفة ،يستخدم الامر sum لجمع عناصر المصفوفة الواحدة بالشكل التالي:-

```
>> a=[1 2 3 4;4 5 6 7;7 8 9 10]
a =
1 2 3 4
4 5 6 7
7 8 9 10
>> s1=sum(a)  ايجاد حاصل جمع قيم عناصر اعمدة المصفوفة كل على حدة
```

```
s1 =
12 15 18 21
```

or

```
>> s1=sum(a,1)  طريقة ثانية للحصول على نفس النتيجة
```

```
s1 =
12 15 18 21
```

```
>> s2=sum(a,2)  ايجاد حاصل جمع قيم عناصر صفوف المصفوفة كل على حدة
```

```
s2 =
10
22
34
```

```
>> s3=sum(sum(a))  ايجاد حاصل جمع جميع عناصر قيم المصفوفة
```

```
s3 =
66
```

or

```
>> s3=sum(a(:))
s3 =
66
```

• لايجاد حاصل جمع عناصر القطر الرئيسي للمصفوفة

```
>> b1=sum(diag(a))
b1 = 15
```

**12- إيجاد حاصل ضرب قيم عناصر المصفوفة :-**

تستخدم الدالة **prod** وهي اختصار لكلمة product لضرب قيم عناصر المصفوفة بحيث يتم ضرب عناصر كل عمود من اعمدة المصفوفة كل على حدة ويكون الناتج على شكل متجه صفي . لاحظ الامثلة التالية :

```
>> a=[1 2 3;4 5 6;7 8 9]
```

```
a =
```

```
1 2 3
4 5 6
7 8 9
```

```
>> p1=prod(a)
```

إيجاد حاصل ضرب عناصر اعمدة المصفوفة كل على حدة

```
p1 =
```

```
28 80 162
```

```
or
```

```
>> p1=prod(a,1)
```

طريقة ثانية للحصول على نفس النتيجة السابقة

```
p1 =
```

```
28 80 162
```

```
>> p2=prod(a,2)
```

إيجاد حاصل ضرب عناصر صفوف المصفوفة كل على حدة

```
p2 =
```

```
6
120
504
```

```
>> p3=prod(prod(a))
```

إيجاد حاصل ضرب جميع عناصر المصفوفة

```
p3 =
```

```
362880
```

```
or
```

```
>> p3=prod(a(:))
```

طريقة ثانية للحصول على نفس النتيجة السابقة

```
p3 =
```

```
362880
```

● لإيجاد حاصل ضرب عناصر القطر الرئيسي للمصفوفة

```
>> b2=prod(diag(a))
```

```
b2 =
```

```
45
```

13- إيجاد العنصر الأكبر في المصفوفة :-

تستخدم الدالة **max** لإيجاد العنصر الأكبر لعناصر كل عمود من اعمدة المصفوفة كل على حدة بحيث يكون الناتج متجه صفي (فيه الرقم الأكبر من كل عمود) لاحظ الامثلة التالية :

```
>> a=[1 2 3;4 5 6;7 8 9]
```



a =

```
1  2  3
4  5  6
7  8  9
```

>> m1=max(a) لايجاد قيمة العنصر الاكبر لكل عمود كل على حدة في المصفوفة

m1 =

```
7  8  9
```

or

>> m1=max(a,[],1) طريقة ثانية للحصول على نفس النتيجة السابقة

m1 =

```
7  8  9
```

>> m2=max(a,[],2) لايجاد قيمة العنصر الاكبر لكل صف كل على حدة في المصفوفة

m2 =

```
3
6
9
```

>> m3=max(max(a)) لايجاد قيمة العنصر الاكبر لجميع عناصر المصفوفة

m3 =

```
9
```

or

>> m3=max(a(:)) طريقة ثانية للحصول على نفس النتيجة السابقة

m3 =

```
9
```

14- ايجاد العنصر الاصغر في المصفوفة :-

تستخدم الدالة **min** (وهي اختصار لكلمة minimum) في ايجاد العنصر الاصغر لكل عمود من اعمدة المصفوفة كل على حدة بحيث يكون الناتج متجه صفي (فيه الرقم الاصغر من كل عمود) لاحظ الامثلة التالية :

>> a=[1 2 3;4 5 6;7 8 9]

a =

```
1  2  3
4  5  6
7  8  9
```

>> n1=min(a) لايجاد قيمة العنصر الاصغر لكل عمود كل على حدة في

المصفوفة

n1 =

```
1  2  3
```

or



```
>> n1=min(a,[],1)
```

طريقة ثانية للحصول على نفس النتيجة

السابقة

n1 =

```
1 2 3
```

```
>> n2=min(a,[],2)
```

لايجاد قيمة العنصر الاصغر لكل صف كل على حدة في

المصفوفة

n2 =

```
1
```

```
4
```

```
7
```

```
>> n3=min(min(a))
```

لايجاد قيمة العنصر الاصغر لجميع عناصر المصفوفة

n3 =

```
1
```

or

```
>> n4=min(a(:))
```

طريقة ثانية للحصول على نفس النتيجة السابقة

n4 =

```
1
```

المصفوفات الخاصة :

1. المصفوفة المربعة Square Matrix:- وهي المصفوفة التي يكون فيها عدد الصفوف مساوي لعدد الاعمدة .

```
>> a=[1 2 3;4 5 6;7 8 9]
```

a =

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

2. المصفوفة الصفريّة Zero Matrix:- وهي المصفوفة التي تحتوي جميع قيم عناصرها على اصفار ويمكن انشاؤها باستخدام الدالة **zeros** بالصيغة التالية :

```
z=zeros(m,n) or z=zeros([m n])
```

حيث z اسم المصفوفة ، m عدد الصفوف ، n عدد الاعمدة .

للحصول على مصفوفة صفريّة مربعة (عدد الصفوف يساوي عدد الاعمدة) فيمكن استخدام الصيغة التالية $z=zeros(n)$ ، والموضحة في الامثلة التالية :-

```
>>zeros(3,4)
```

ans =

```
0 0 0 0
```

```
0 0 0 0
```

```
0 0 0 0
```

```
>>zeros(3)
```

```
ans =
```

```
0 0 0
0 0 0
0 0 0
```

3. المصفوفة الاحادية Ones Matrix :- وهي المصفوفة التي تحتوي جميع قيم عناصرها على الواحد الصحيح ، ويمكن انشاؤها باستخدام الدالة **ones** كما مبين ادناه :

```
O=ones(m,n) or O=ones([m n])
```

حيث O اسم المصفوفة ، m عدد الصفوف ، n عدد الاعمدة .

اذا اردنا الحصول على مصفوفة احادية مربعة (عدد الصفوف يساوي عدد الاعمدة) فيمكن استخدام الصيغة التالية $O=ones(n)$ ، والموضحة في الامثلة التالية :-

```
>>ones(3,4)
```

```
ans =
```

```
1 1 1 1
1 1 1 1
1 1 1 1
```

```
>>ones(2)
```

```
ans =
```

```
1 1
1 1
```

4. المصفوفة المحايدة Identity Matrix :- وهي مصفوفة تتكون من القيم 0 , 1 والرقم واحد يمثل جميع عناصر القطر الرئيسي لها ، اما باقي عناصرها الاخرى اصفار ، ولانشاء هذا النوع من المصفوفات نستخدم الدالة **eye**

```
e=eye(m,n) or e=eye([m n])
```

حيث e اسم المصفوفة ، m عدد الصفوف ، n عدد الاعمدة .

اذا اردنا الحصول على مصفوفة محايدة مربعة (عدد الصفوف يساوي عدد الاعمدة) فيمكن استخدام الصيغة التالية $e=eye(n)$ ، والموضحة في الامثلة التالية :-

```
>>eye(2,3)
```

```
ans =
```

```
1 0 0
0 1 0
```

```
>>eye(2)
```

```
ans =
```

```
1 0
0 1
```



5. مصفوفة القيم العشوائية Random Matrix:- وهي مصفوفة قيم عناصرها عشوائية ويمكن انشاؤها باستخدام الدالة **rand** وتكون قيم عناصرها محصورة بين 0,1 من القيم العشرية كما في المثال التالي :

`r=rand(m,n) or r=rand([m n])`

حيث **r** اسم المصفوفة ، **m** عدد الصفوف ، **n** عدد الاعمدة .
اذا اردنا الحصول على مصفوفة عشوائية مربعة (عدد الصفوف يساوي عدد الاعمدة) فيمكن استخدام الصيغة التالية `r=rand(n)` ، والموضحة في الامثلة التالية :-

```
>>rand(2,3)
```

```
ans =
```

```
0.8913 0.4565 0.8214  
0.7621 0.0185 0.4447
```

```
>>rand(2)
```

```
ans =
```

```
0.6154 0.9218  
0.7919 0.7382
```

6. المصفوفة السحرية Magic Matrix:- وتستخدم لانتاج مصفوفة مربعة بشكل عشوائي ، ومن ميزات ان مجموع قيم عناصر كل صف يساوي مجموع قيم عناصر كل عمود ويساوي مجموع عناصر القطر الرئيسي وتكون قيمها محصورة بين الواحد الصحيح وبين مربع طول المصفوفة كما في الصيغة التالية :

`M=magic(n)`

حيث **M** اسم المصفوفة ، **m** عدد الصفوف ، **n** عدد الاعمدة .

```
>> M=magic(3)
```

```
M =
```

```
8 1 6  
3 5 7  
4 9 2
```



العمليات الرياضية الأساسية على المصفوفات

قبل البدء في اجراء عمليات (جمع ،ضرب، طرح،قسمة) وبعض العمليات الاخرى بين مصفوفتين عدديتين يشترط برنامج MATLAB ان يكون للمصفوفتين نفس عدد الصفوف m ونفس عدد الاعمدة n او ان يكون احدهما قيمة عددية مفردة scalar علما ان ناتج العملية بين المصفوفتين سيكون عبارة عن مصفوفة لها نفس الابعاد $(m*n)$. لاحظ العمليات التي ستجرى على المصفوفتان A,B

```
>> A=[2 4;6 8;10 12]    >> B=[1 3;5 7;9 11]
```

```
A =
     2     4
     6     8
    10    12

B =
     1     3
     5     7
     9    11
```

```
>> D=[2 4 6 8 10]
```

```
D =
     2     4     6     8    10
>> E=[1 3 5 7 9]
E =
     1     3     5     7     9
```

1. عملية الجمع والطرح للمصفوفات والمتجهات :-

تتم عملية الجمع بجمع العنصر في الصف الاول من العمود الاول للمصفوفة A مع ما يناظره في المصفوفة B ثم العنصر في الصف الاول من العمود الثاني للمصفوفة A مع ما يناظره للمصفوفة B وهكذا لبقية الصفوف . اما جمع المتجهات فانها تتم بجمع العنصر الاول للمصفوفة D مع ما يناظره في المصفوفة E وهكذا . وكذلك لعملية الطرح .

```
>> C=A+B
C =
     3     7
    11    15
    19    23
```

```
>> F=D+E
F =
     3     7    11    15    19
```

```
>> C=A-B
C =
     1     1
     1     1
     1     1
```

```
>> F=D-E
F =
     1     1     1     1     1
```



اما اذا كانت عملية الجمع او الطرح بين عدد ومصفوفة فستكون بين ذلك العدد وكل عنصر من عناصر المصفوفة ، لاحظ الامثلة التالية :

```
>>b=[ 8 1 6;3 5 7;4 9 2 ]
```

```
b=
```

```
8 1 6
```

```
3 5 7
```

```
4 9 2
```

```
>>c=b-3
```

```
C=
```

```
5 -2 3
```

```
0 2 4
```

```
1 6 -1
```

مثال/ جد ناتج المعادلة التالية :
 $P = 9 * b + 3$
 *(يجب مراعاة اسبقية العمليات)

```
>>p = 9 * b + 3
```

```
P=
```

```
75 12 57
```

```
30 48 66
```

```
39 84 21
```

مثال/ جد ناتج جمع المصفوفتين b و c اذا كانت

$c = b - 3$, $b = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$

```
>> b=[8 1 6;3 5 7;4 9 2]
```

```
b=
```

```
8 1 6
```

```
3 5 7
```

```
4 9 2
```

```
>> c = b - 3
```

```
c=
```

```
5 -2 3
```

```
0 2 4
```

```
1 6 -1
```

```
>> a = b + c
```

```
a=
```

```
13 -1 9
```



```
3 7 11
5 15 1
```

مثال/ جد ناتج المعادلة التالية $c = 2 * b - 3$ اذا كان

$$b = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}, c = b - 3$$

```
>> b = [8 1 6; 3 5 7; 4 9 2]
```

```
b =
```

```
8 1 6
3 5 7
4 9 2
```

```
>> c = b - 3
```

```
c =
```

```
5 -2 3
0 2 4
1 6 -1
```

```
>> 3 * b - 2 * c
```

```
ans =
```

```
14 7 12
9 11 13
10 15 8
```

2. ضرب المصفوفات والمتجهات

اولا : ضرب المصفوفات :- تتم عملية الضرب باحدى الطريقتين التاليتين :

(A) الطريقة الاولى : يتم ضرب كل عنصر من عناصر المصفوفة الاولى مع ما يقابلها في المصفوفة

الثانية في حالتين : الاولى يجب ان تكون المصفوفتين المراد ضرب عناصرهما متماثلتين في عدد

الصفوف والاعمدة والثانية ان تكون احدى المصفوفتين قيمة عددية مفردة scalar .

وتتحقق هذه العملية باضافة علامة النقطة (.) قبل علامة الضرب (*) لاحظ الامثلة التالية :

```
>> c=[5 -2 3; 0 2 4; 1 6 -1]
```

```
c =
```

```
5 -2 3
0 2 4
1 6 -1
```

```
>> b=[8 1 6; 3 5 7; 4 9 2]
```

```
b =
```



```

8  1  6
3  5  7
4  9  2
>> c.*b
ans =
    40    -2    18
     0    10    28
     4    54    -2
>> w=2.*c

```

```

w =
    10    -4     6
     0     4     8
     2    12    -2

```

(B) الطريقة الثانية:- عند اجراء عملية ضرب مصفوفتين، فشرط ضربهما هو ان يكون عدد اعمدة المصفوفة الاولى مساوي لعدد صفوف المصفوفة الثانية، وتتم عملية الضرب بضرب عناصر السطر الاول من المصفوفة الاولى مع عناصر العمود الاول من المصفوفة الثانية مع الجمع لينتج العنصر الاول (في الصف الاول والعمود الاول) من المصفوفة الجديدة وكذلك الحال بالنسبة لبقية العناصر الاخرى. لاحظ الشكل التالي :

لنفرض ان a و b مصفوفتان :-

$$a = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad b = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

فان حاصل ضربهما ينتج المصفوفة التالية :-

$$a * b = \begin{matrix} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} & a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} & a_{11} * b_{13} + a_{12} * b_{23} + a_{13} * b_{33} \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} & a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} & a_{21} * b_{13} + a_{22} * b_{23} + a_{23} * b_{33} \\ a_{31} * b_{11} + a_{32} * b_{21} + a_{33} * b_{31} & a_{31} * b_{12} + a_{32} * b_{22} + a_{33} * b_{32} & a_{31} * b_{13} + a_{32} * b_{23} + a_{33} * b_{33} \end{matrix}$$

مثال / جد حاصل ضرب المصفوفتين a * b

$$a = \begin{bmatrix} 5 & -2 & 3 \\ 0 & 2 & 4 \\ 1 & 6 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

```

>> a=[5 -2 3;0 2 4;1 6 -1]
a =
     5     -2     3

```



```
0 2 4
1 6 -1
>> b=[8 1 6; 3 5 7; 4 9 2]
b =
8 1 6
3 5 7
4 9 2
>> a * b
ans =
46 22 22
22 46 22
22 22 46
```

ثانيا : ضرب المتجهات:- تتم عملية الضرب باحدى الطريقتين التاليتين :

(A) **الطريقة الاولى:-** في حالة ضرب متجهين صفيين او متجهين عموديين فيشترط ان يكون للمتجهين المراد ضرب عناصرهما نفس عدد الابعاد، ولتنفيذ العملية يجب اضافة النقطة (.) قبل علامة الضرب .

(B) **الطريقة الثانية:-** في حالة ضرب متجه صفي باخر عمودي يشترط ان يكون لهما نفس عدد العناصر ، ويتم ضرب العنصر الاول للمتجه الاول بالعنصر الاول للمتجه الثاني وهكذا لبقية العناصر ثم ايجاد مجموع حاصل الضرب ، كما في الامثلة التالية :-

```
>> x=[1 3 5 7 9]
x =
1 3 5 7 9
>> y=[2 4 6 8 10]
y =
2 4 6 8 10
>> z=x.*y
z =
2 12 30 56 90
```

```
>> x=[1 3 5 7 9]
x =
1 3 5 7 9
>> y=[2 ; 4; 6; 8; 10]
y =
2
4
```



```
6
8
10
>> z=x*y
z =
190
```

ثالثا : ضرب كل عنصر من عناصر المصفوفة في نفسه :-

إذا كانت x تمثل مصفوفة أبعادها 3×4 ، لايجاد حاصل ضرب كل عنصر في نفسه يكون بالشكل التالي :-

```
>>x=[8 1 6 1;3 5 7 5;4 9 2 6]
x =
     8     1     6     1
     3     5     7     5
     4     9     2     6
>> x.^2
ans =
```

```
    64     1    36     1
     9    25    49    25
    16    81     4    36
```

ملاحظة:- لا يمكن استخدام x^2 أي رفع المصفوفة في المثال أعلاه لأن عدد عناصر الصفوف لا يساوي عدد عناصر الأعمدة ، وتستخدم x بدلا من x^2 ، أما إذا كانت المصفوفة متساوية في عدد الصفوف والأعمدة فيمكن استخدام x^2 من دون إضافة النقطة مع ملاحظة طريقة الضرب بدون استخدام الـ $\text{Dot}(\cdot)$ لاحظ المثال التالي :

```
>> x=[8 1 6;3 5 7;4 9 2]
x =
     8     1     6
     3     5     7
     4     9     2
>>x^2
ans =
    91    67    67
    67    91    67
    67    67    91
```

3- قسمة المصفوفات والمتجهات

يمكن اجراء القسمة بين مصفوفتين اذا كانتا مربعيتين او ان تكون احدهما قيمة عددية مفردة scalar ، ويكون ذلك باستخدام (" \ ") slash لاجراء عملية القسمة من اليسار الى اليمين ، كما يمكن استخدام (" / ") backslash لاجراء عملية القسمة من اليمين الى اليسار (قسمة المقام على البسط) .

A (قسمة عناصر مصفوفتين مباشرة كل عنصر مع ما يقابله : ويكون ذلك باستخدام النقطة (.) Dot

```
>> a=[4 6 ;8 10]
```

```
a =
```

```
4    6
```

```
8    10
```

```
>> b=[2 3;4 5]
```

```
b =
```

```
2    3
```

```
4    5
```

```
>> c = a. / b
```

```
c =
```

```
2    2
```

```
2    2
```

(B قسمة مصفوفة على اخرى : مثلا قسمة المصفوفة c على b تمثل ضرب المصفوفة c في مقلوب b وتتم عملية قسمة المصفوفة a كوحدة واحدة على المصفوفة b كوحدة واحدة . (لاحظ عدم استخدام النقطة (.) dot

```
>> c = a / b
```

```
c =
```

```
2    0
```

```
0    2
```

قسمة المتجهات

لاجراء عملية قسمة المتجهات ، يجب ان يكون المتجهين متساويين في عدد العناصر او تكون احدهما ذات قيمة عددية مفردة .

A(قسمة عنصر على عنصر : وتتم باستخدام النقطة (.) dot ، لاحظ المثال التالي :-

```
>> a=[1 2 3 4]
```

```
a =
```

```
1    2    3    4
```

```
>> b=[2 4 6 8]
```



b =

2 4 6 8

(B) قسمة متجه كوحدة واحدة على متجه اخر كوحدة واحدة :- تمثل ضرب المتجه c في مقلوب b وجمع كل قيمة الضرب وتتم بدون استخدام النقطة dot ، لاحظ المثال التالي :-

>> c=b/a

c =

2.0000

Exercise (A)

Q1) Use MATLAB code to evaluate the following functions for x from 1 to 2 in steps of 0.1

1. $y = x^3 + 3x^2 + 1$
2. $y = \sin x^2$
3. $y = (\sin x)^2$
4. $y = \sin 2x + x \cos 4x$

sol.

```
x = 1 : 0.1 : 2
% part 1
y = x.^3 + 3*x.^2 + 1;
% part 2
y = sin (x.^2);
% part 3
y = (sin ( x ) ).^2;
% part 4
y = sin ( 2*x ) + x. * cos ( 4 * x ) ;
```

Q2) The following code is supposed to evaluate the function :

$$f(x) = \frac{x^2 \cos \pi x}{(x^2 + 1)(x + 2)}$$

for $x \in [0, 1]$ (using 200 steps). Correct the code :-

```
x = linspace (0,1);
clear all
g = x^3+ 1;
H = x + 2;
z = x.^2;
y = cos xpi;
f = y*z/g*h
```

```
clear all
x = linspace ( 0,1,200);
g = x.^3+1;
h = x+2;
z = x. ^2;
y = cos (x*pi);
f = y.*z./(g.*h);
```

Q3) Write MATLAB program to find the minimum element in matrix a(4,2).

Solution

```
clc ; clear; close all;
```



```

a = input ('the matrix');
min=a(1,1);
for i = 1:4
    for j = 1:2
        if a(i , j)<= min
min=a(i , j);
        end
    end
end
disp(min);

```

Q4) Suppose a and b are defined as follows :

a = [2 -1 5 0] ;

b = [3 2 -1 4] ;

Evaluate by hand the vector c in the following statements :

- c = a - b ;
- c = b + a - 3 ;
- c = 2 * a + a .^ b ;
- c = b ./ a ;
- c = b .\ a ;

solution

```

c = a - b
c = -1 -3 6 -4
c = b + a - 3
c = 2 -2 1 1
c = 2 * a + a .^ b
c = 12.0000 -1.0000 10.2000 0
c = b ./ a
c = 1.5000 -2.0000 -0.2000 inf
c = b .\ a
c = 0.6667 -0.5000 -5.0000 0

```

Q5) Write MATLAB program to calculate the following sum

$$\sum_{i=1}^N \frac{1}{i} + \frac{1}{(i+2)(i+3)}$$

Solution

```

N = input (' Enter N ');
sum = 0;
for I = 1 : N
sum = sum + 1/j + 1/ ((j + 2 ) * ( j + 3 ));
end

```



```
disp ( [ 'The answer is ' , num2str(sum) ] )
```

ملاحظة/ يجب ترك فراغ او وضع فارزة
بين عنصر وآخر لوجود اقواس مصقوفة

Q6) Work out the results of the following expression before checking them at the command line :

```
a = [ -1  0  3 ];
```

```
b = [ 0  3  1 ];
```

```
~ a
```

```
a & b
```

```
a | b
```

solution

```
a = [ -1  0  3 ];
```

```
b = [ 0  3  1 ];
```

```
~ a
```

```
ans =
```

```
0  1  0
```

```
a & b
```

```
ans =
```

```
0  0  1
```

```
a | b
```

```
ans =
```

```
1  1  1
```

Exercise (B)

Q1) Set up a vector n with elements 1 , 2 , 3 , 4 , 5 . Use MATLAB array operations on the vector n to set up the following four vectors , each with five elements :

- 2 , 4 , 6 , 8 , 10
- $1/2$, 1 , $3/2$, 2 , $5/2$
- 1 , $1/2$, $1/3$, $1/4$, $1/5$
- 1 , $1/2^2$, $1/3^2$, $1/4^2$, $1/5^2$

Solution

```
n = [ 1 2 3 4 5 ]
m = n . * 2
m = 2 4 6 8 10
m = n . / 2
m = 0.5000 1.0000 1.5000 2.0000 2.5000
m = 1 . / n
m = 1.0000 0.5000 0.3333 0.2500 0.2000
m = 1 . / n.^2
m = 1.0000 0.2500 0.1111 0.0625 0.0400
```

Q2) Determined the output of the following program :

```
Clear all
a = input ( ' please enter a ' );
b = input ( ' please enter b ' );
c = input ( ' please enter c ' );
v1 = a + b + c ;
v2 = a / (( b + c ) * ( c + a )) ;
v3 = a / ( b * c ) ;
disp ( [ ' a + b + c = ' num2str ( v1 ) ] )
disp ( [ ' v2 = ' num2str ( v2 ) ] )
disp ( [ ' v3 = ' num2str ( v3 ) ] )
```

solution

```
please enter a 2
please enter b 1
please enter c 3
a + b + c = 6
v2 = 0.1
v3 = 0.66667
```

Q3) Given that $a = [1 \ 0 \ 2]$ and $b = [0 \ 2 \ 2]$ determine the values of the following expressions . Check your answers with MATLAB .

- $a \sim b$
- $a < b$
- $a < b < a$
- $a < b < b$

Solution



```

a = [ 1  0  2 ];
b = [ 0  2  2 ];
a ~ = b
ans = 1  1  0
a < b
ans = 0  1  0
a < b < a
ans = 1  0  1
a < b < b
ans = 0  1  1

```

Q4) Write some MATLAB statements on the command line which use logical vectors to count how many elements of a vector x are negative (-ve), zero or positive (+ve) . check that they work , e.g. with the vector :

```
[ -4  0  5  -3  0  3  7  -1  6 ]
```

Solution

```
a = [ -4  0  5  -3  0  3  7  -1  6 ];
```

```
[ i  j ] = find ( a < 0 )
```

```
i =
```

```
1  1  1  —————> رقم السطر
```

```
j =
```

```
1  4  8  —————> رقم العمود
```

```
[ i  j ] = find ( a == 0 )
```

```
i =
```

```
1  1
```

```
j =
```

```
2  5
```

```
[ i  j ] = find ( a > 0 )
```

```
i =
```

```
1  1  1  1
```

```
j =
```

```
3  6  7  9
```

ملاحظة :- يستخدم الامر find لايجاد موقع (index) من عناصر المصفوفة .

Q5) Write MATLAB program to calculate the following sum :

$$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots$$

Solution

```
clc ; clear ; close all;
```

```
x = input ( ' enter the value of x : ' );
```

```
n = input ( ' enter end of series n = ' );
```

```
sign = -1 ;
```



```
sum = 0 ;  
for i = 1 : 2 : n  
sign = - sign ; % sign = sign * -1  
sum = sum + sign * x ^ n / n ;  
end  
disp ( sum );
```

Q6) Translate the following expressions into MATLAB :

$$P + \frac{w}{u}$$

$$P + \frac{w}{u+v}$$

$$P + \frac{\frac{w}{u+v}}{u-v}$$

Solution

$$(a) \ p + w / u$$

$$(b) \ p + w / (u + v)$$

$$(c) \ (p + w / (u + v)) / (p + w / (u - v))$$



Exercise (C)

Q1) Work out the result of the following expressions :-

$T1 = [1 \ 1; 0 \ 1]; T2 = [1 \ 0; 0 \ 0];$

$T = T1 \ \& \ T2$

$T = T1 \ | \ T2$

$T = \sim T1$

Solution

$T1 = [1 \ 1; 0 \ 1];$

$T2 = [1 \ 0; 0 \ 0];$

$T = T1 \ \& \ T2$ gives the matrix $T1 = [1 \ 0; 0 \ 0]$

$T = T1 \ | \ T2$ gives the matrix $T = [1 \ 1; 0 \ 1]$

$T = \sim T1$ gives the matrix $T = [0 \ 0; 1 \ 0]$

Q2) Write a short MATLAB program to input an integer n and build a n by n matrix with the numbers 1 , 2 ,.... N on the main diagonal and zeros everywhere else .

Solution

```
clc ; clear ; close all ;
n = input ( ' integer number : ' );
A = zeroes ( n , n );
for i = 1 : n
    for j = 1 : n
        if i == j
            A( i , j ) = n ;
        end
    end
end
disp (A) ;
```

Q3) The following code is supposed to evaluate the values :-

$a + b + c$, $\frac{a}{(b+c)(c+a)}$, $\frac{a}{bc}$ corrected the cod.

$a = \text{input} (\text{please enter } a) ;$

$b = \text{input} (\text{please enter } b) ;$

$c = \text{input} (\text{please enter } c) ;$

$v1 = a \ b \ c ;$

$v2 = a / (b + c) (c + a) ;$

$v3 = a / b \ c ;$

clear all

$\text{disp} ([' a + b + c = ' \ \text{num2str}(v3)])$

$\text{disp} ([' v2 = ' \ \text{num2str}(v1)])$

$\text{disp} ([' v3 = ' \ \text{num2str}(v2)])$



```

clear all
a = input ( 'please enter a' ) ;
b = input ( 'please enter b' ) ;
c = input ( 'please enter c' ) ;
v1 = a + b + c ;
v2 = a / ( ( b + c ) * ( c + a ) ) ;
v3 = a / ( b * c ) ;
disp ( [ ' a + b + c = '   num2str(v1) ] )
disp ( [ ' v2 = '   num2str(v2) ] )
disp ( [ ' v3 = '   num2str(v3) ] )

```

Q4) Explore the use of the function round , ceil ,floor and fix for the values x = 0.3 , x = 0.5 , x = 1.65 and x = -1.34

Solution

```

x = [ 0.3 , 0.5 , 1.65 , -1.34 ];
y = round (x)
y = 0     1     2     -1
y = ceil (x)
y = 1     1     2     -1
y = floor (x)
y = 0     0     1     -2
y = fix (x)
y = 0     0     1     -1

```

Q5) Work out by hand the output of the following script :-

```

s = 1 : 6 ;
t = 6 : -1 : 1 ;

```

solution

```

s = [ 1  2  3  4  5  6 ];
t = [ 6  5  4  3  2  1 ];
s + t = 7     7     7     7     7     7
s - t = -5    -3    -1     1     3     5
s . * t = 6     10    12    12    10     6
s . / t = 0.1667    0.4000    0.7500    1.3333    2.5000    6.000
s . ^ t = 1     32    81    64    25     6
1 ./ s = 1.0000    0.5000    0.3333    0.2500    0.2000    0.1667
s / 2 = 0.5000    1.0000    1.5000    2.0000    2.5000    3.0000
s + 2 = 3     4     5     6     7     8

```

Q6) Determine all integer n between 1 and 50 for which $n^3 - n^2 + 40$ is greater than 1000 and n is not divided by 3 .

Solution

Clear all;



```
n = 1 : 50 ;  
x = find ( ( ( n.^3 - n.^2 + 40 ) > 100 ) & ( mod ( n ,3 ) ~ = 0 ) )  
x =
```

Columns 1 through 13

5 7 8 10 11 13 14 16 17 19 20 22 23

Columns 14 through 26

25 26 28 29 31 32 34 35 37 38 40 41 43

Columns 27 through 31

44 46 47 49 50

Q7) Write a script, which inputs any two numbers (which may be equal), and displays the larger one with a suitable message, or if they are equal, displays a message to that effect.

Solution

```
a = input ( ' enter the first no. : ' );  
b = input ( ' enter the second no. : ' );  
if a > b  
    disp ( [ num2str( a ) ' is greater than ' num2str( b ) ] );  
elseif a < b  
    disp ( [ num2str( b ) ' is greater than ' num2str( a ) ] );  
else  
    disp ( [ num2str( a ) ' is equal with ' num2str( b ) ] );  
end  
>> q7  
enter the first no. :5  
enter the second no. : 9  
9 is greater than5
```