

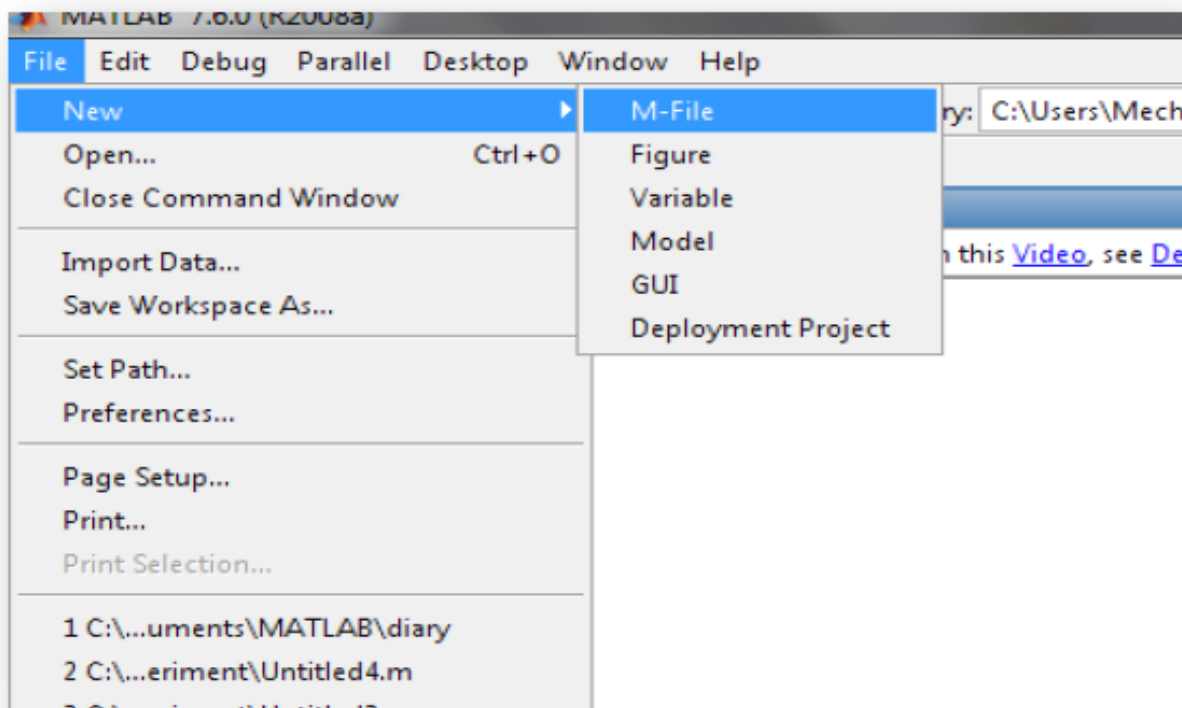
4.1 التعريف ب M-file

في الفصول الماضية قمنا باستخدام الماتلاب كآلة حاسبة متطورة نوعاً ما , لكن لم تتعدى تطبيقاتنا عن الحسابات وادخال المعلومات بشكل يدوي او تعريفها مسبقاً لكن بدون عمليات منطقية لكن في هذا الفصل الذي يعتبر الأكثر أهمية ومتعة في ذات الوقت سوف نتعرف على لائحة ال M-file في الماتلاب التي يشبهها البعض الى ال C++ ولكنني اختلف معهم كثير لان تطبيقاتها اوسع بكثير واشمل وأروع . وما يميز ال M-file عن ال command windows انه يتم كتابة جميع الأوامر والتعليمات بداية وبعدها يتم تطبيق ما تم كتابته اما في ال CW كنا نكتب أمر تلو الأمر ويتم تطبيق الأمر تلو الأمر مما يقلل من فرصة تصحيح الأخطاء وأن حدث خطأ عليك الأعادة .

لفتح قائمة ال M-file قم باتباع التالي

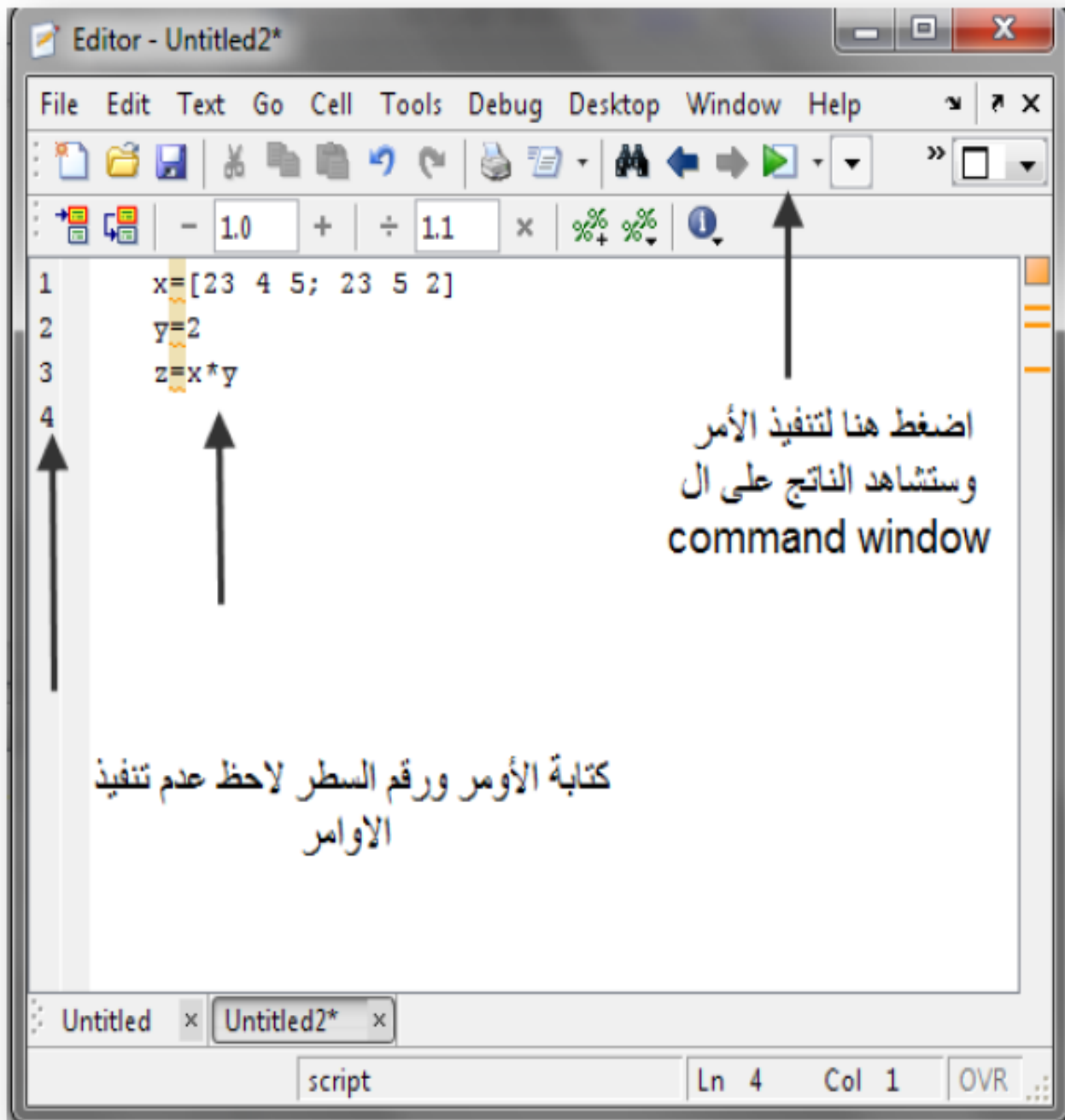
File >> New >> M-file

انظر الشكل



فلنقم بتجريب كتابة أوامر أو الحساب عليه

انظر الشكل



بعد تنفيذ الأمر سيطلب التخزين حتى يقوم بتنفيذه على ال cw , قم بالتخزين والأكمال ويتم
دائماً تخزين ملفات الماتلاب بصيغة m.

```
x =
    23     4     5
    23     5     2

y =
     2

z =
    46     8    10
    46    10     4

>>
```

النتيجة في ال CW
لاحظ ظهور المتغيرات لعدم وجود
فاصلة منقوطة في نهاية تعريف
المتغيرات

ملاحظة : ينصح دائماً في بداية تنفيذ اي برنامج على ال M-file كتابة الأمرين clear ثم clc وذلك لضمان عدم حصول اخطاء في تعريف المتغيرات اذا كانت معرفة مسبقاً و clc لحذف الشاشة في بداية كل أمر حاول كتابة ما تشاء من البرامج وتطبيقها .

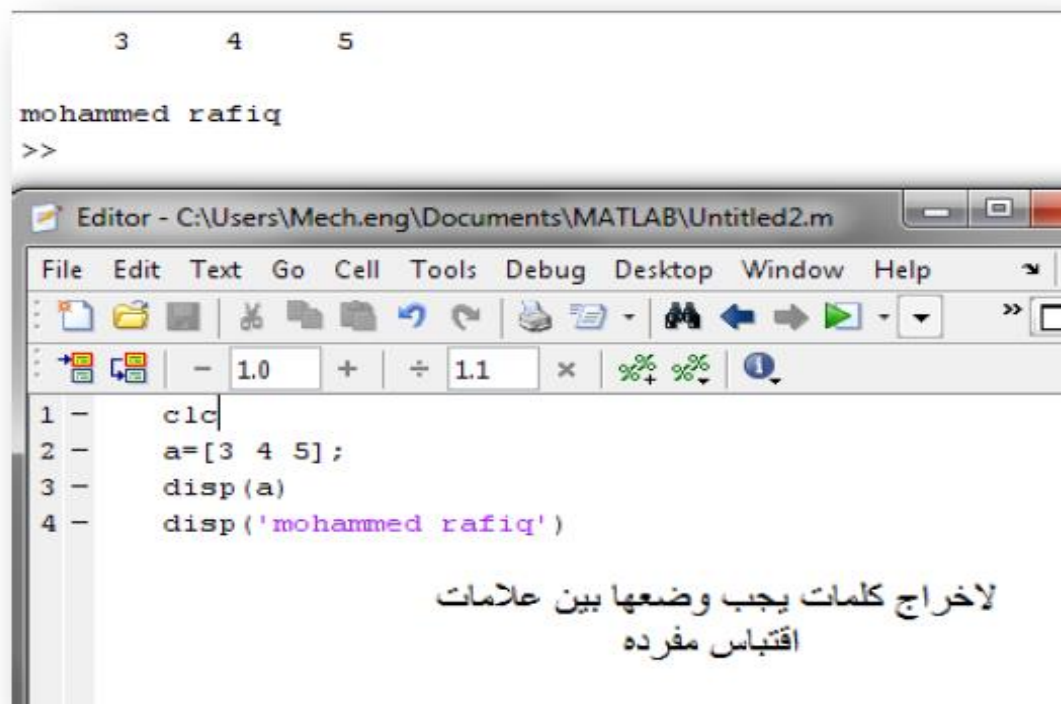
4.2 أوامر ال M-files

disp -1

هذا الامر يساوي cout في لغة ال C++ ويعني اظهار او اخراج

```
3      4      5

mohammed rafiq
>>
```



لاخراج كلمات يجب وضعها بين علامات
اقتباس مفردة

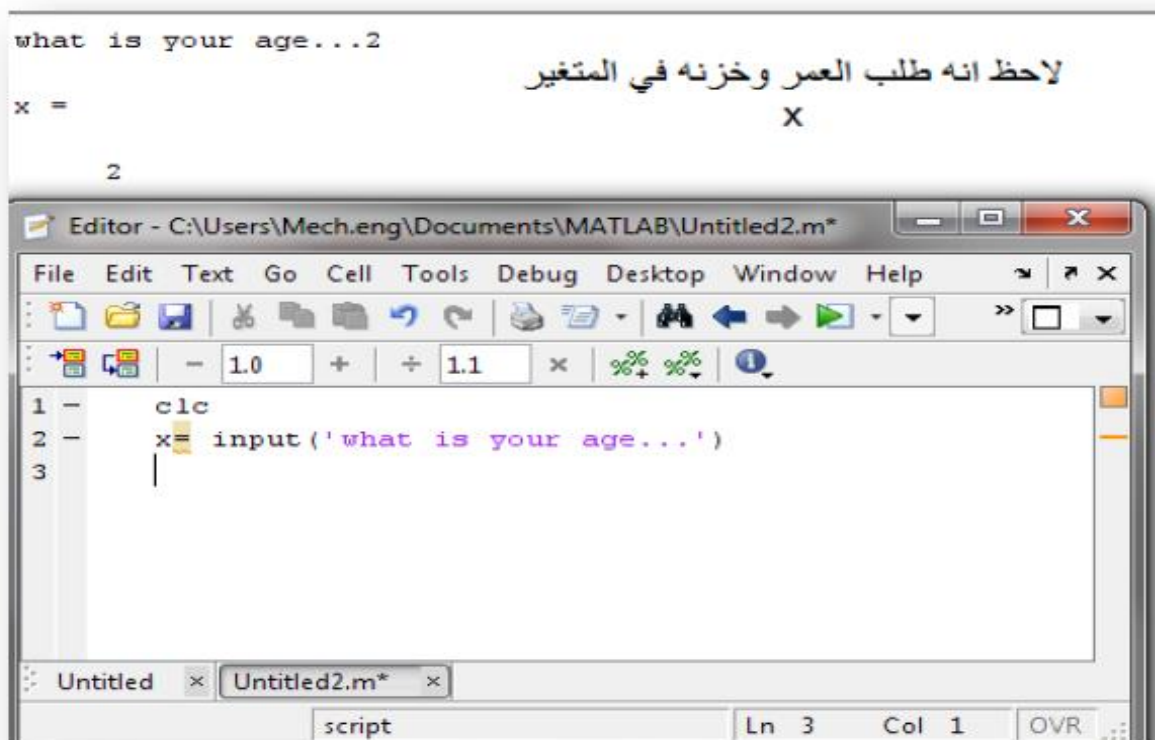
input -2

ويساوي الامر cin في لغة ال ++c ويعني الادخال

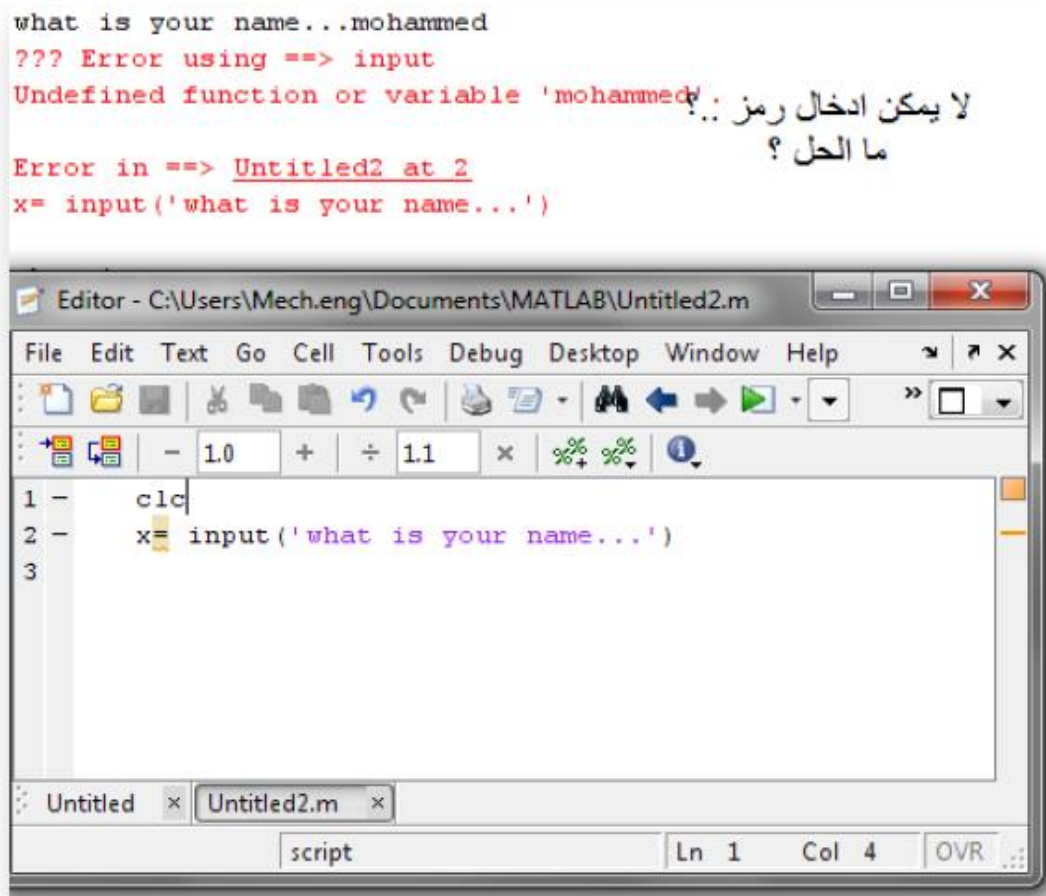
```
what is your age...2

x =

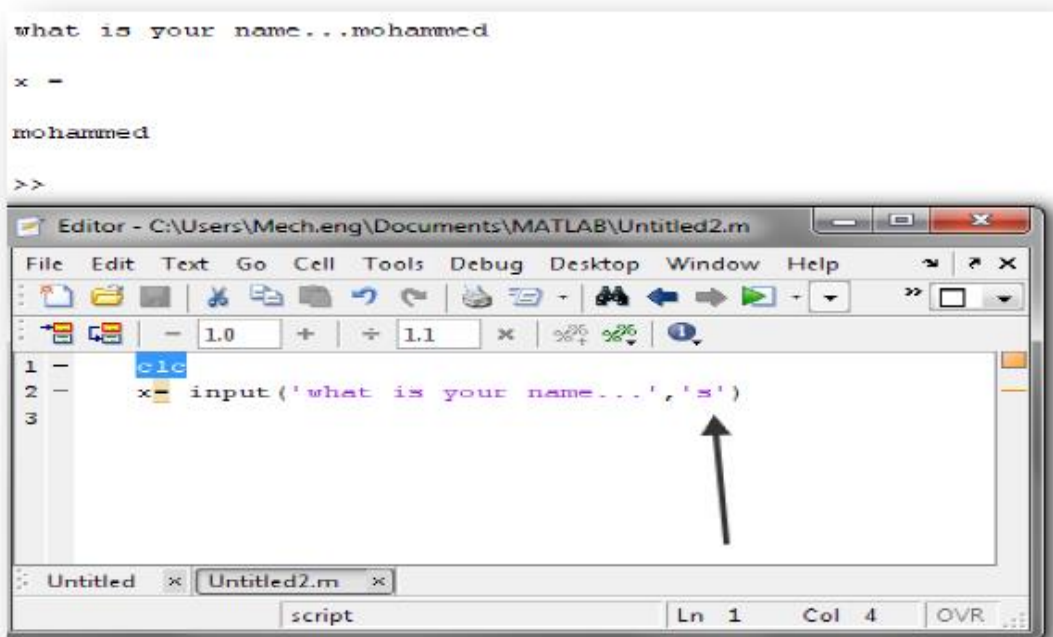
2
```



لاحظ انه طلب العمر وخرنه في المتغير
x



عندما نطلب لادخال اسم يجب بعد الكتاب ان نضع الرمز 's' انظر المثال للتوضيح اكثر



هناك نوعين من الأرقام الأول string والثاني number حيث الماتلاب يعامل الأول كأنه رمز وعندما يتم جمعه أو طرحه مع رقم ثاني سيعطي نتيجة خاطئ أما الثاني فهو رقم ويعامل معاملة الرقم

تحويل ال string الى رقم والعكس نستخدم احد الأمرين num2str or str2num سيأتي أهمية هذين الأمرين في جمل if و for لاحقاً .

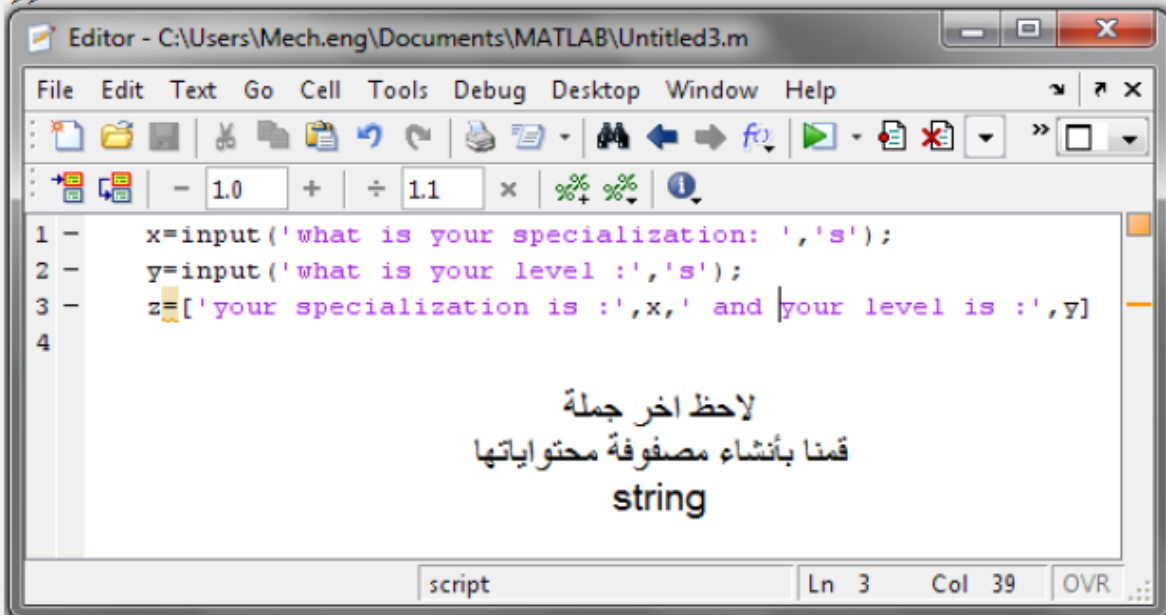
مثال: اكتب أمر يطلب من المستخدم ادخال تخصصه ومستواه ثم يظهر له عالاشاشه تخصصه ومستواه

```
what is your specialization: mechatronics
what is your level :third

z =

your specialization is :mechatronics and your level is :third

>>
```



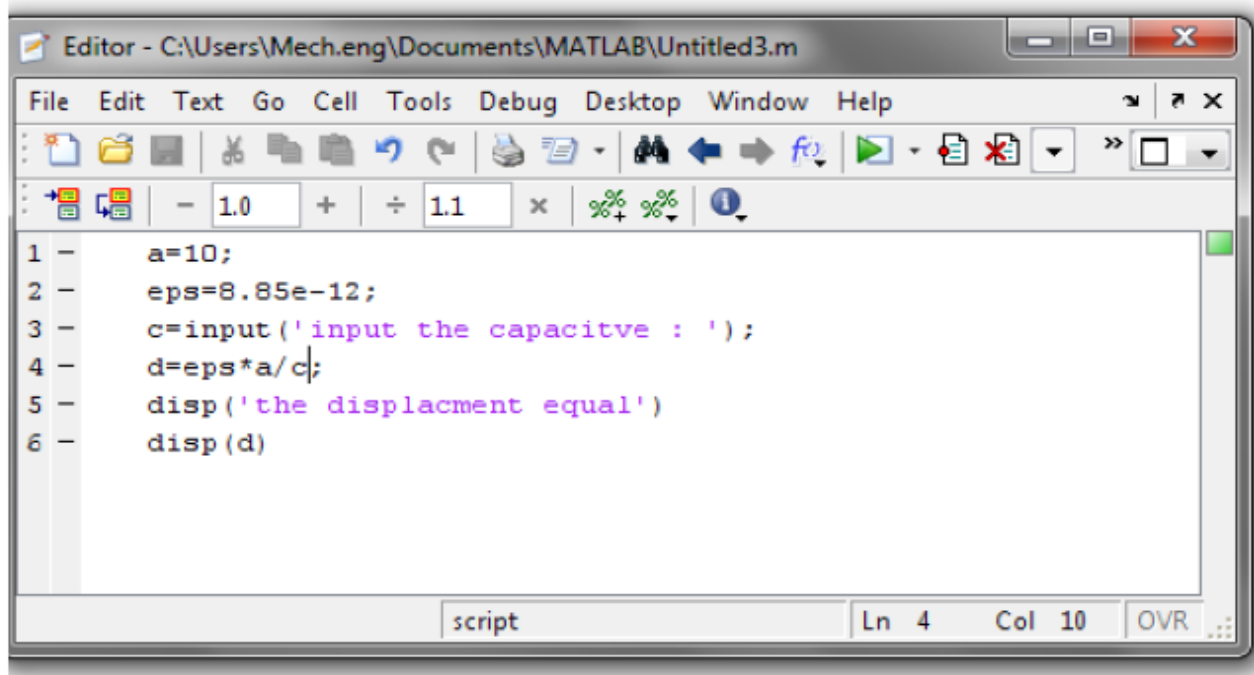
مثال : مواسعة المكثف تعطي بالعلاقة التالية :

$$C = \frac{\epsilon_0 \epsilon_r A}{d}$$

ويوجد لدينا اجهزة تعمل على قياس المواسعة c . ونحن نعلم مساحة سطح كل مكثف A وهي تساوي 10 متر تربيع وقيمة ايسلون معروفة لدينا تساوي $8.85e-12$ في الفراغ والمكثفات المستخدمة هي الفراغ

المطلوب: انشاء برنامج يطلب ادخال المواسعة . ويعطيك المسافه بين القطبين d

```
input the capacitve : 20e-12
the displacment equal
4.4250
```



Menu -3

الصيغه العامة

$k = \text{menu}(\text{'title'}, \text{'option1'}, \text{'option2'}, \dots \text{etc})$

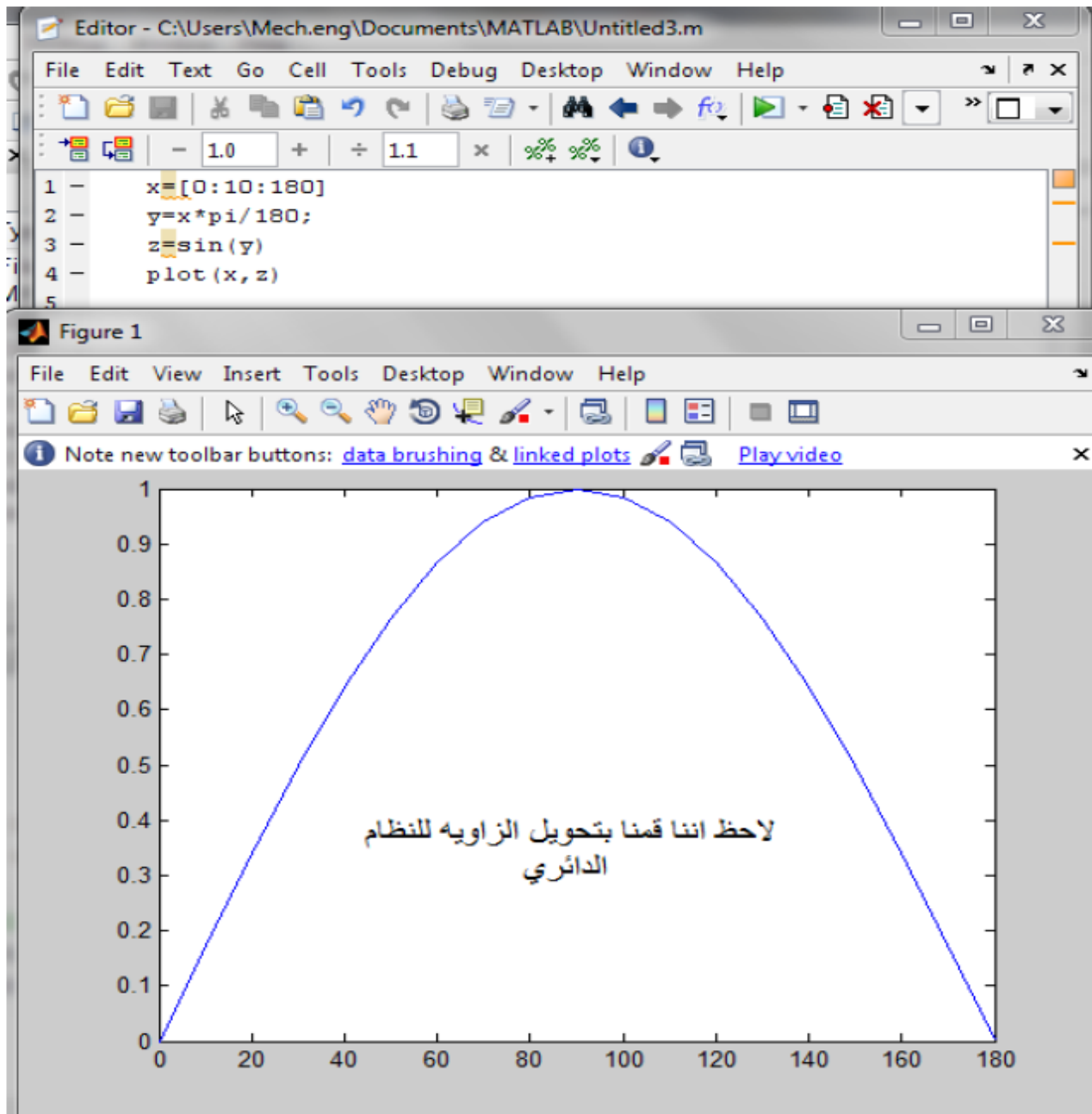
حاول تجريبه . سيتم استخدامه لاحقاً

plot -4

يمكنك استخدام الامر plot لرسم العلاقه بين متغيرين وسيتم توضيح لاحقاً بشكل تفصيلي لكن سنكتفي الآن بتطبيق سهل له وهو $\text{plot}(\text{var1}, \text{var2})$

مثال : اكتب أمراً لرسم الموجه الجيبية من 0-180 درجة

الحل : انظر الشكل



4.3 المقارنة

الرمز	الوظيفة
<	اقل
>	اكبر
<=	اقل او يساوي
>=	اكبر او يساوي

هل يساوي ؟	==
لا يساوي	~=
And	&
Or	!
Not	~

يمكن اجراء هذه العمليات على عناصر منفردة او مصفوفات

شاهد المثال

```
>> x=[3 4 5];
y=[3 9 1];
z=(x==y)
```

z =

```
1    0    0
```

```
>> z=x>y
```

z =

```
0    0    1
```

```
>> z= x<=y
```

z =

```
1    1    0
```

ياخذ كل عنصر على حدى
للمقارنه

```
x =
     2     3     4
     2     4     5
    6 6 5     6     4

>> y=[3 4 5 ; 6 3 4 ; 13 5 4 ]

y =
     3     4     5
     6     3     4
    13     5     4

>> z=x (y>x)
z =
     2
     2
     3
     4
```

معنى هذا الأمر هو إيجاد عناصر في x تحقق الشرط y>x

لاحظ: قمنا باستخدام الأمر var(var1 compare var) وهذا يعني اوجد في المتغير var عناصر تحقق الشرط var1 اقل من var

لاحظ هنا يعطيك العناصر وليس مكانها المصفوفة ولايجاد مكانها في المصفوفة نستخدم الأمر find

انظر المثال

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

x =
     1     2     3
     4     5     6

>> y= [ 3 2 1 ; 10 11 2]

y =
     3     2     1
    10    11     2

>> z=y (x>y)

z =
     1
     2
```

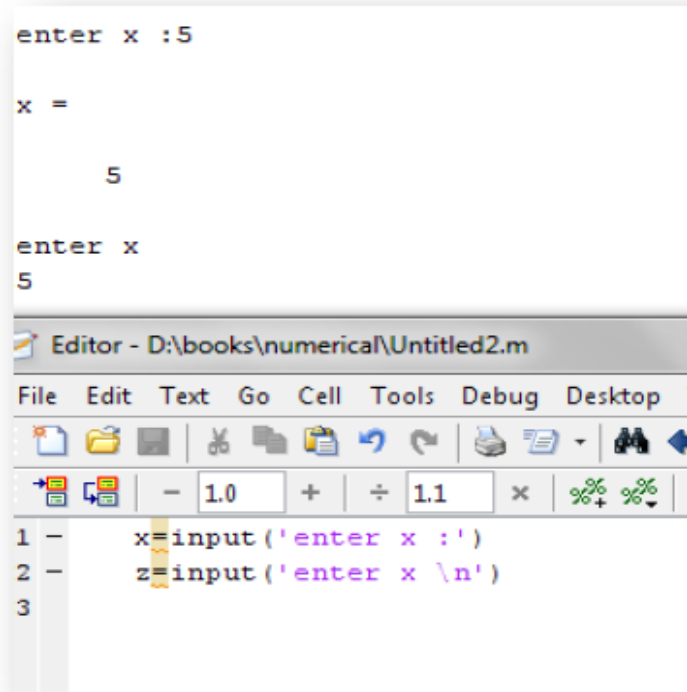
ايجاد العناصر التي تحقق الشرط

```
>> find(x>y)

ans =
     5
     6
```

ايجاد مكان العنصر الذي يحقق الشرط
لاحظ انه يعطي مكان ببعد واحد

ملاحظة: تستطيع وضع اخر جملة طلب الادخال الرمز \n حتى يتم طلب الادخال بنزول
سطر انظر المثال



The screenshot shows the MATLAB Editor window with the file 'Untitled2.m' open. The code in the editor is:

```
1 - x=input('enter x :')
2 - z=input('enter x \n')
3
```

The command window above the editor shows the execution of these commands. The first command 'enter x :5' prompts the user to enter a value for x, and the user enters '5'. The second command 'enter x' prompts the user to enter a value for z, and the user enters '5'. The command window also shows the variable assignments: 'x = 5' and 'z = 5'.

دون استخدام \n قام بطلب قيمة x بنفس السطر .. اما بوجودها طلبها بسطر اسفل .
هكذا نكون قد انتهينا من ال M-file وأوامره لكن لم ننتهي من تطبيقاته لأننا لم نبدأ بها بعد .
في الفصل القادم سيتم توضيح الأمور الخاصة بال Loops التي لا يخلو برنامج مهم منها .
لكن حاول الآن ان تحل المسائل القادمة حتى تبدأ الفصل القادم وانت فاهم لما تقدم .

problems

4.1 اكتب برنامج تقوم به بأدخال معادلات بعدد المجاهيل ويقوم بإيجاد المجاهيل

الجواب:

```
z=input(' the left of equal matrix: ');
w=input(' the right of equal matrix: ');
q= inv(z);
s= q*w
```

بدون مقدمات لهذا الفصل فالجمل الدورانية او الشرطية فهي مشابهة تماما لما في برنامج ال
C++ مع اختلاف بسيط وسيتم توضيح جميعها في هذا الفصل

if statement 5.1

ثلاث انواع لجمله if

simple if statement
nested if statement
if-else statement

simple if statement-1

الشكل العام

```
if expression 1  
    statement group 1  
end
```

nested if statement-2

الشكل العام

```
if expression 1  
    statement group 1  
    if expression 2  
        statement group 2  
    end  
    statement group 3  
end  
statement group 4
```

if-else statement-3

الشكل العام

```
if expression 1  
statement group 1  
else  
statement group 2  
end
```

لاحظ جميع الأوامر تنتهي ب **end**

الفرق بين الانظمة الثلاث .

النظام الأول يكون فيه جملة if واحده فهو نظام بسيط

اما النظام الثاني يكون فيه اكثر من جملة if داخل بعضها (لا ينظر بتحقيق الشرط الثاني الا اذا تحقق الأول)

اما النظام الثالث فهو نظام ينتقل عند عدم تحقيق الشرط الى else ويحققها

امثلة على الانظمة الثلاث

مثال 1 :

اكتب برنامج يقوم بحساب الجذر للأرقام الموجبة فقط

الحل: هنا يوجد شرط واحد فقط . اذا لم يتحقق لن يتم تحقيق اي شيء

```
enter number: -15
enter number: 25
```

```
w =
```

```
the sqrt of 25 = 5
```

```
>>
```

لاحظ:

عند ادخال رقم سالب لا يقوم بعمل شيء
استخدام الامر num2str في السطر الرابع

```
Editor - C:\Users\Mech.eng\Documents\MATLAB\Untitled2.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 - x=input('enter number: ');
2 - if x>=0
3 -     z=sqrt(x);
4 -     w=['the sqrt of ',num2str(x),' = ',num2str(z)]
5 - end
6
```

لاحظ اننا هنا استخدمنا الأمر num2str لانه كما ورد سابق المصفوفه يجب ان تكون كلها string او كلها numbers والمصفوفه W في السطر الرابع في مثالنا نريدها ان تكون مصفوفه رموز strings فحولنا x & z الى حروف .

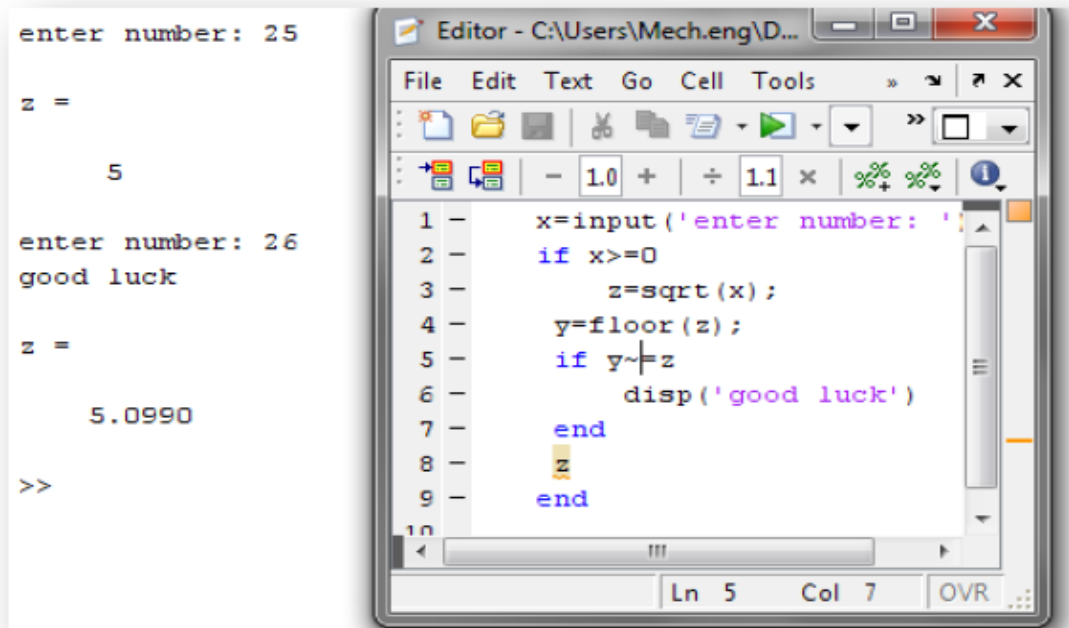
ملاحظه لو قمنا بتحويل x من البداية الى حروف ك ان نكتب هذا الأمر

x=input('enter number','s') سوف يتم تعريف x على اساس انها رمز فعند ايجاد جذرها لن يتم ايجاده بالطريقة الصحيحه . حاول التجريب

مثال 2 :

اكتب برنامج يقوم بايجاد الجذر للارقام الموجبة فقط واذا كان جوابه كسراً يقوم بكتابة good luck واذا كان الجواب ليس كسر يقوم بأخراج الجواب .

الحل : هنا يوجد شرطان . الشرط الثاني لا يتحقق الا بتحقق الشرط الأول وفي هذه الحالة يتم استخدام النمط الثاني من جمل if



The image shows a MATLAB Editor window with a script and its execution results. The script is as follows:

```
1 - x=input('enter number: ');
2 - if x>=0
3 -     z=sqrt(x);
4 -     y=floor(z);
5 -     if y~=z
6 -         disp('good luck')
7 -     end
8 -     z
9 - end
```

The execution results on the left show:

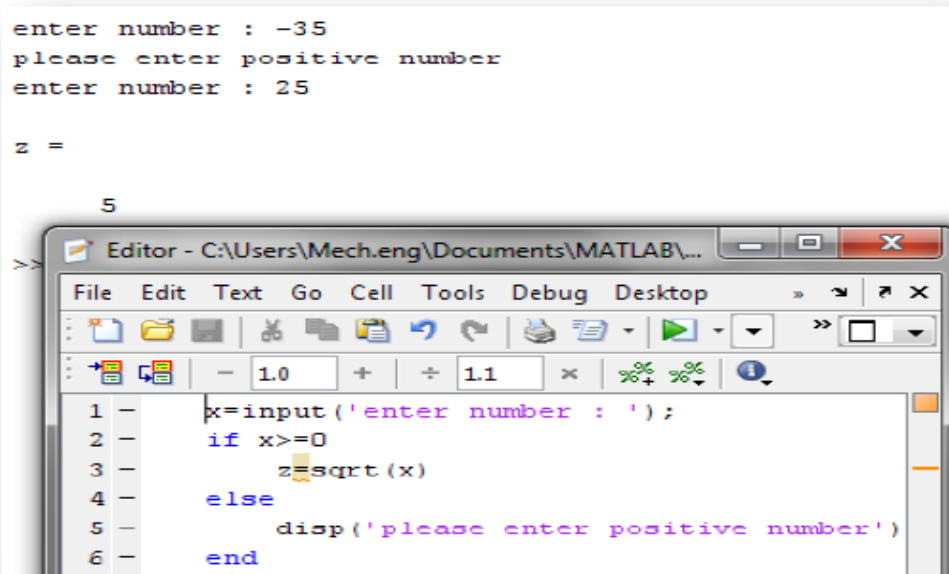
```
enter number: 25
z =
    5
enter number: 26
good luck
z =
    5.0990
>>
```

لاحظ استخدام الأمر floor الذي يعني اكبر عدد صحيح .

مثال 3 : اكتب برنامج يقوم بايجاد جذر الارقام الموجبة واذا كان الرقم سالب يقوم بكتابة

please enter positive number

الحل : هنا يوجد شرطان اذا لم يتحقق الأول سيتم تحقيق الثاني اذا نستخدم النظام الثالث



The image shows a MATLAB Editor window with a script and its execution results. The script is as follows:

```
1 - x=input('enter number : ');
2 - if x>=0
3 -     z=sqrt(x)
4 - else
5 -     disp('please enter positive number')
6 - end
```

The execution results on the left show:

```
enter number : -35
please enter positive number
enter number : 25
z =
    5
>>
```


for statement 5.2

الصيغة العامة

```
for i:s:f  
statement group X  
end
```

جملة for هي احد الجملـة الدورانية المهمة في عالم الماتلاب

i : تعني القيمة الابتدائية ويمكن ان تكون متغير

s: تعني الخطوه ويمكن ان تكون متغير وهي ليست اجبارية كتابتها اذا لم يتم كتابتها سيأخذ الخطوه تساوي 1

f : القيمة النهائية

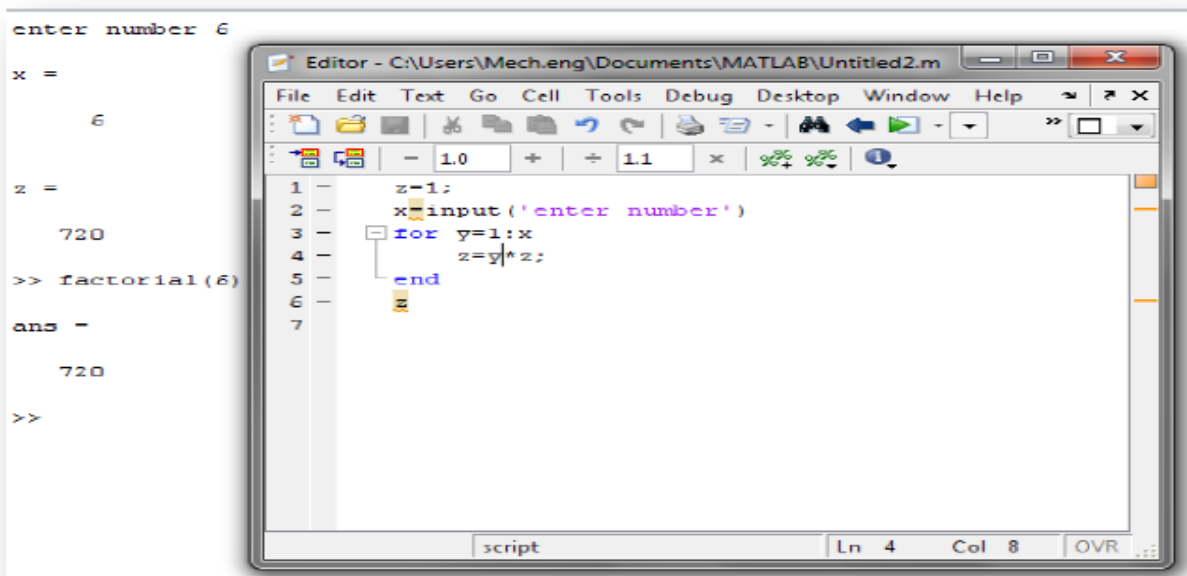
لطالما جملة for بين القيمة الابتدائية والنهائية سيتم تنفيذ البرنامج عندما تخرج عن النطاق سيذهب الى end

انظر الامثلة للتوضيح اكثر

مثال : اكتب برنامج يعرف ال factorial (المضروب)

ملاحظه: لايجاد المضروب بالماتلاب نستخدم الأمر التالي factorial(var or num)

الحل :



الأمريـن Break & continue

يستخدم هذين الأمرين للتخطي أو الأيقاف فالأمر break عند تحقق الشرط يقوم بإيقاف ال Loop والأمر continue عند تحقق الشرط يقوم بالتخطي والعودة لجملة if "اعلم أنك لم تفهم جيداً " انظر المثال للتوضيح

```
0
1
3
4
>>

Editor - C:\Users\Mech.eng\Documents\MA
File Edit Text Go Cell Tools Debug
- 1.0 + ÷ 1.1 x
1 - for y=0:4
2 -     if y==2
3 -         continue
4 -     end
5 -     disp(y)
6 - end
    لم يتم بتنفيذ جملة
    عندما for
    y=2
```

وتوضيح للأمر break

```
0
1
>>

1 - for y=0:4
2 -     if y==2
3 -         break
4 -     end
5 -     disp(y)
6 - end
```

عمل الأمر break على توقيف جملة for

while statement 5.3

الصيغة العامة

```
while expression 1  
statement group 1  
end  
statement group 2
```

وهذه الجملة تعني أنه لطالما الشرط expression 1 متحقق سيتم تحقيق statement group 1 وعندما لا يتحقق سوف يتم تحقيق مجموعة الشروط 2 .

واليك الأمثلة التالية:

مثال (1):

```
>> x = 10;  
>> if x == 10  
    msgbox ('ok', 'result');
```

مثال (2):

```
>> if x == 10  
    msgbox ('ok', 'result');  
else  
    msgbox ('no', 'result');  
end;
```

مثال (3):

```
>> x = 11;  
>> if x == 1  
    disp ('1');  
elseif x == 2  
    disp ('2');  
else  
    disp ('3');  
end;
```

الإخراج

الصيغة SWITCH-CASE

عندما يتوجب علينا تنفيذ أوامر اعتماداً على استخدام متكرر لاختيار كمي لوسط ما، عندها من السهل استخدام الصيغة switch-case التي لها الصيغة العامة التالية:

```
switch expression
case test-expression1
    (commands1)
case test-expression2
    (commands2)
otherwise
    (commands3)
end
```

يجب أن يكون expression هنا إما عدداً مفرداً أو سلسلة رمزية. يقارن التعبير expression الموجود في الصيغة السابقة بالتعبير test-expression1 الموجود في عبارة case الأولى. وإذا تساوى التعبيران، سيتم تنفيذ الأوامر (commands1) وتخطي التعليمات الواقعة بعدها حتى العبارة end. أما إذا لم يتحقق الشرط الأول، فسيختبر الشرط الثاني، حيث سيقارن expression في المثال السابق مع العبارات test-exoression2 الموجودة في عبارة case الثانية. وإذا تساوى التعبيران، سيتم تنفيذ (commands2) وتهمل بقية العبارات حتى عبارة end. إذا لم تحقق أي عبارة case المساواة مع التعبير expression، عندها ستنفذ الأوامر (commands3) التي تلي العبارة otherwise. لاحظ من الشرح الذي أوردناه عن صيغة switch-case بأنه سيتم تنفيذ إحدى مجموعات الأوامر المكونة للصيغة switch-case واليك الأمثلة التالية:

مثال (1):

```
x = 1;
switch x
case {1, 2, 3, 4, 5}
    disp ('1..5');
case {9, 10}
    disp ('9..10');
otherwise
    disp ('this is impossible');
end;
```

الإخراج 1..5

مثال (1):

```
>> x = 1;
```

```
>> while x < 25
```

```
    disp (x);
```

x

```
= x + 1;
```

```
end;
```

الإخراج

1

2

3

.

.

24