كلية العلوم

القسم : **الرياضيات**

السنة : **الرابعة**

المادة : برمجة غرضة التوجة

المحاضرة : الرابعة /عملي/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

كلية العلوم ، كلية الصيدلة ، الهندسة التقنية

٦

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر (What's app-Telegram) على الرقم 0931497960

جامعة طرطوس
كلية العلوم
قسم الرياضيات
السنة الرابعة

برمجة غرضية التوجه

المحاضرة الرابعة

القسم العملي

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 =
```

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)              %display only variable value
   5
>> display(x)          %display variable name and value
x =
   5
```

3. الامر (File Print Format) fprintf:-

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 fprinf('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 .

---

ftnirpf لا تستخدم لطباعة قيمة عددية

```
>> a =١ ;
fprintf(a);
Error using fprintf
No format string
```

لاستخدام ftnirpf لطباعة قيمة عددية يجب
استخدام احدى الصيغ تنسيق البيانات مثال:

```
;  a =١
 ( fprintf ('the  value of a is  %d\n ', a

the value of a is ١
```

تعابير التحكم بالتدفق

يشتمل MATLAB على عدد كبير من التعابير المستخدمة في التحكم بتدفق المعطيات كالتعابير الشرطية بالإضافة إلى عبارات تسمح بتكرار مجموعة من التعليمات وهو ما يعرف بالحلقات Loop
نعرض فيما يلي بعض العبارات (أو التعابير) الخاصة بالتحكم بالتدفق ضمن MATLAB

If/ elseif/else	تنفيذ مجموعة من العبارات إذا كان الشرط محقق
For	تنفيذ مجموعة من العبارات عدد محدد من المرات
while	تنفيذ مجموعة من العبارات بشكل متكرر طالما أن الشرط محقق
end	إنهاء مجموعة من التعليمات أو الإشارة لنهاية تعليمة ما
break	إنهاء تنفيذ حلقة for أو while
switch/case/otherwise	التبديل بين عدة حالات cases اعتماداً على تعبير ما

عبارة if : تتألف هذه العبارة من تعليمة if يليها تعبير منطقي يليه مجموعة من التعليمات أو العبارات في حال تحقق الشرط الموجود ضمن التعبير يتم تنفيذ البرنامج أما في حال عدم تحقق الشرط لا يتم تنفيذ أي تعليمة ويتم الانتقال إلى التعليمات اللاحقة بعد end

عبارة if else : تتألف هذه العبارة من تعليمة if يليها تعبير منطقي يليه مجموعة من التعليمات أو العبارات في حال تحقق الشرط الموجود ضمن التعبير يتم تنفيذ البرنامج أما في حال عدم تحقق الشرط يتم تنفيذ مجموعة التعليمات التي تلي التعليمة else

مثال:

البرنامج التالي يقوم بتوليد عدد طبيعي عشوائي ثم يقوم بعدة اختبارات وهي:

- إذا كان العدد أصغر تماماً من 30 تظهر رسالة بكلمة small
- إذا كان العدد أصغر تماماً من 80 وأكبر أو يساوي 30 تظهر رسالة بكلمة medium
- إذا كان العدد أكبر أو يساوي 80 تظهر رسالة بكلمة large

```
a = randi(100,1)
if a<30
    disp('small')
elseif a<80
    disp('medium')
else
    disp('large')
end
```

```
a =

    82

large
>> Untitled8
a =

    91

large
>> Untitled8
a =

    13

small
```

عبارة switch : تقوم كتلة تعليمات switch بتنفيذ إحدى المجموعات من التعليمات والعبارات بشكل شرطي من ضمن عدة خيارات من مجموعات من التعليمات والعبارات كل خيار يدعى حالة case

مثال:

Untitled9*

```
grade = 'B';
switch(grade)
    case'A'
        fprintf('excellent \n');
    case'B'
        fprintf('well done \n');
    case 'C'
        fprintf('not bad \n');
    case 'D'
        fprintf('you passed \n');
    case'F'
        fprintf('better try again \n');
    otherwise
        fprintf('invalid grade\n');
end
```

خرج البرنامج:

Command Window

```
>> Untitled9
well done
```

حلقة while : طالما أن الشرط الموجود لا زال محققاً فإن الحلقة تستمر بتنفيذ التعليمات داخل الحلقة ومن الضروري عدم نسيان تعليمة end في نهاية حلقة while لأنه في حالة وضع شرط صحيح دوماً تدخل البرمجية في حلقة لا نهائية ، من أجل الخروج من هذه الحلقة يمكن استخدام ctrl + c

مثال:

Untitled10*

```
a=5;
b=5;
while(a<15 && b<30)
    fprintf('value of a: %d\n',a);
    fprintf('value of b: %d\n',b);
    a = a+1;
    b = b+6;
end
```

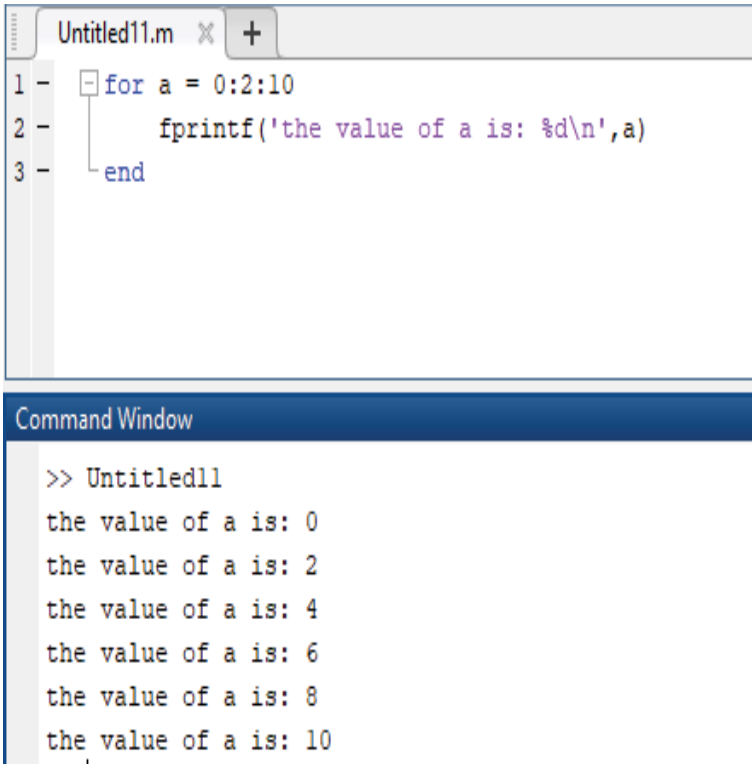
```
>> Untitled10
value of a: 5
value of b: 5
value of a: 6
value of b: 11
value of a: 7
value of b: 17
value of a: 8
value of b: 23
value of a: 9
value of b: 29
```

يقوم هذا البرنامج بطباعة قيمة المتحولين a و b طالما أن قيمة المتحول a أصغر من 15 وقيمة المتحول b أصغر من 30 أي أنه يكفي أن يصبح أحد المتحولات أكبر من القيمة المحددة فقط للخروج من الحلقة

الشرط (a<15 && b<30) يتطلب تحقيق الحالتين a و b وفي حال عدم تحقق أحدهما هذا يعني أن الشرط غير محقق
بينما عند (a<15 || b<30) يقبل أحد الحالتين أو كلاهما وهذا يعني أنه في حال عدم تحقق أحد الشرطين لا يتم الخروج من الحلقة وإنما يستمر حتى يتم اختلال الشرط الآخر

حلقة for : يتم استعمال الحلقة for عندما نحتاج لتكرار تنفيذ مجموعة من التعليمات عدد محدد ومعروف مسبقاً من المرات

مثال:



دوال ملفات M
M- File Function

الدوال function: الدوال هي مجموعة من الأوامر أو الجمل البرمجية المكتوبة في ملفات M-Files والتي تستخدم لتؤدي وظيفة معينة ولها اسم مميز يعبر عن وظيفتها لتؤدي أمر أو مجموعة من الأوامر الجاهزة.

أنواع الدوال function type:

1- يحتوي برنامج MATLAB على مئات من الدوال الداخلية الجاهزة مبنية في بيئة البرنامج بدلاً من كتابة أو برمجة هذه الدوال في كل مرة مثل:
sum, prod, mean, inv, det, size, length, angle, abs, ... وغيرها من الدوال التي تم برمجتها مسبقاً في برنامج MATLAB
2- دوال يتم إنشاءها من قبل المستخدم تكتب بصيغة برنامج MATLAB ويسمى هذا النوع من البرمجة بدوال ملفات M
تضاف هذه الدوال بعد إنشائها إلى مكتبة الدوال الداخلية الجاهزة
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 variable

OR
Function function_name(list of input variable)

شروط تسمية وإنشاء دوال ملفات M (M-File Function) :

1- يسمى اسم الملف بنفس اسم الدالة التي تم كتابتها في البرنامج
2- السطر الأول من البرنامج يجب أن يبدأ بالأمر function
3- يتبع اسم الدالة شروط تسمية المتغيرات في برنامج MATLAB
4- لا يجوز استخدام اسم الدالة من أسماء مشابهة لأسماء المتغيرات المستخدمة في البرنامج

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

1- برنامج الدوال يبدأ بالأمر function بينما برنامج ال script يبدأ بالبرنامج مباشرة
2- يتطلب كتابة برنامج الدوال function تعريف المخرجات والمدخلات مع اسم الدالة بينما في البرمجة النصية script لا يحتاج إلى ذلك
3- تستطيع في برنامج الدوال function عند تنفيذه تغيير قيم المدخلات بينما في البرمجة النصية ال script لا يمكن ذلك
4- المتغيرات التي تحسب داخل ال function لا تحفظ في work space بينما في ال script تحفظ في work space وتسمى متغيرات محلية.

مثال 1 : اكتب برنامج يحل معادلة من الدرجة الثانية باستخدام Function M-File
وإذا كانت قيمة الجذر أقل من الصفر اطبع العبارة (the root is complex)

quadratic_equation.m

```
1   function quadratic_equation(a,b,c)
2 -   delta = b^2 - 4*a*c
3 -   if delta >0
4 -   x1 = (-b + sqrt(delta)) / (2*a)
5 -   x2 =(-b - sqrt(delta)) / (2*a)
6 -   elseif delta <0
7 -   disp ('the root is complex')
8 -   else
9 -   x1_2 = (-b / (2*a))
10 -  end
```

نتائج تنفيذ البرنامج

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

delta =

     4

x1 =

   -0.5000

x2 =

    -1
```

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

```
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
```

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

max =

    18
```

نتائج تنفيذ البرنامج

مثال 3: اكتب برنامج بلغة MATLAB لإدخال رقم واختبار هل هو زوجي أم فردي :

test_number.m

```
function n = test_number(a)
if mod(a,2)==0
    n = 'even number';
else
    n = 'odd number';
end
```

```
>> test_number(4)

ans =

even number
```

مثال 4: اكتب برنامج بلغة MATLAB لإيجاد معامل عدد :

fact.m

```
function c =fact(n )

v = 1;
for i=1:n
    v = v * i;
end
v

end
```

```
>> fact(5)

v =

   120
```

مثال 5: اكتب برنامج بلغة MATLAB لإظهار جدول الضرب لأي رقم :

production.m

```
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 إذا كان الرقم المدخل هو ليس عنصر واحد أو قيم ليست عددية فيطبع العبارة التصحيحية لتبين عدم إدخال صحيح
2- يعمل الأمر disp على طبع مصفوفة مكونة من 5 عناصر يجب تحويل كل عنصر من عناصرها إلى نصوص (سلسلة حرفية string) عن طريق num2str حيث لا يمكن استخدام مصفوفة مكونة من أرقام وحروف، يجب أن تكون المصفوفة إما حروف لوحدها أو أرقام لوحدها

نتائج تنفيذ البرنامج السابق

```
>> production(4)
4X1=4
4X2=8
4X3=12
4X4=16
4X5=20
4X6=24
4X7=28
4X8=32
4X9=36
4X10=40
```

```
>> production([1 2 ])
error: please enter one number
>> production('m')
error: please enter one number
```

فيما يلي جدول يوضح معاملات المقارنة والمعاملات المنطقية المستخدمة في كتابة البرامج :

المعامل	الوصف
<	أصغر من
<=	أصغر أو يساوي
>	أكبر من
>=	أكبر أو يساوي
==	إشارة المساواة
~=	إشارة عدم المساواة
&	AND (وَ)
|	OR (أو)
~	NOT(نفي)

تدريب 1 :

اكتب برنامج في MATLAB يقوم بحساب مجوع الأعداد الزوجية المحصورة بين 0 و 20 وطباعة الاعداد الزوجية ثم أخيراً طباعة المجموع على الشاشة

تدريب 2 :

اكتب برنامج في MATLAB لحساب مكعب أي عدد يتم إدخاله

تدريب 3 :

اكتب برنامج في MATLAB يقوم بحساب مجموع ثلاثة أعداد مدخلة وإظهاره ثم اختبار ناتج الجمع إن كان عدد زوجي ام فردي وعرض نتيجة الاختبار على الشاشة

تدريب 4:

اكتب برنامج MATLAB يقوم بحساب المتوسط الحسابي لمجموعة أعداد مدخلة

اشرح عمل التابع التالي واكتب الخرج الذي سيقوم بإظهاره عند استدعاؤه :

to_meter.m

```
function to_meter(n, unit)
switch unit
    case('km')
        n = n*1000;
    case('hm')
        n = n*100;
    case ('dam')
        n = n*10;
    case ('dm')
        n = n/10;
    case('cm')
        n = n/100;
    case('mm')
        n = n/1000;
    case('m')
        n = n;
    otherwise
        disp('unvalid unit');
end
fprintf('the meter value is %f\n',n);
end
```

انتهت المحاضرة