

10. الطرق التكرارية او الحلقات Loops

تعتبر ايعازات السيطرة والتحكم من اكثر الأيعازات التي تستخدم في برمجة الماتلاب كما هو الحال في جميع لغات البرمجة الأخرى ، كما انها تعتبر من الأيعازات التي تساعد في بناء البرامج و هيكلية الخوارزميات لأستخدامها الواسع و لسهولة تنفيذ البرامج حيث يتم من خلالها تطبيق العمليات على المصفوفات و تكرار العمليات من خلال عملية تنفيذ لإيعاز سيطرة معينة بايعاز سيطرة اخرى أي يمكن ان تتداخل ايعازات التحكم و السيطرة ، كأداة if الشرطية مع ايعاز التحكم بالتكرار for كما تتميز بانها سلسلة الأستخدام والتراكيب و مرونة ربطها مع ايعازات تحكم اخرى تتعدد ايعازات التحكم و السيطرة لتستخدم حسب الحاجة المطلوبة لها في المكان المناسب ضمن النطاق المحدد لهيكلية البرنامج و هي كالآتي :

1- ايعازات التحكم المشروط

يقسم هذا النوع من التحكم المشروط الى نوعان هما :

أ) ايعاز الشرط if : و هو ايعاز رئيسي يتم من خلالها وضع الشرط الخاص أو المطلوب ، حيث يتم هنا وضع الشرط الذي يقوم بترشيح العملية المناسبة أو الحالة الخاصة المراد تطبيقها على الحالة العامة ، وعادةً ما تمثل العبارة أو الجملة الشرطية عملية رياضية منطقية مثل اكبر من أو اصغر من تكوين نتيجة بالأعتماد على ما مدخل معين او سلسلة من المدخلات المتتالية.

ب) ايعاز الشروط المنفصلة switch : حيث ان هذه الأداة واسعة الأستعمال في البرمجة لانه يمكننا من خلالها تنفيذ مجموعة من العمليات المطوقة بعدة شروط ضمن حالة خاصة.

2- ايعازات التحكم بالتكرار

وتقسم الى اربعة اصناف و هي :

أ) ايعاز التحكم المفتوح for : بواسطة هذه الأداة يكون باستطاعتنا تكرار تنفيذ سلسلة من العمليات ضمن الجزء المخصص لها في البرنامج و يكون باستطاعتنا تكرار عملية معينة حسب الحاجة للوصول الى الناتج أو النتائج المطلوب استخراجها

ب) ايعاز التحكم المشروط while : حيث يمكن بهذه الأداة تنفيذ مجموعة من العمليات و لكن ينتهي عمل هذه الأداة بتحقق الشرط المحدد الذي تم وضعه داخل الأداة

ج) ايعاز الأجتياز و العبور continue : و هو ايعاز يعمل على ايقاف تنفيذ عملية معينة أو العبور عنها و ضمن سلسلة من العمليات التي سيجري تنفيذها.

د) ايعاز توقيف التنفيذ break : و يعمل هذا الأيعاز على وقف عمل اداة التكرار عند تحقق شرط معين ضمن هيكلية البرنامج

3- ايعاز التحكم بالعمليات الخاطئة

وتقسم هذه الأيعازات الى
أ) ايعاز التنفيذ الخاطئ try : حيث من الممكن ان يكون هناك عملية معينة قد يحدث فيها
تنفيذ خاطئ ضمن سلسلة من العمليات الخاطئة ، حيث توضع العملية أو العمليات التي من
الممكن ان يحدث فيها خطأ داخل هذه الأداة

ب) ايعاز التأثير على الخطأ catch : حيث تربط هذا الأيعاز بالأيعاز السابق try حيث
بهذا الأيعاز سيتم الأسارة بعبارة معينة أو عملية خاصة تقوم بالأشارة عند وقوع الخطأ
داخل الجزء الذي وقع فيه خطأ

4- ايعاز العودة الى الحالة الرئيسية return

حيث تستخدم هذا الأيعاز في حالة ما يكون فيها البرنامج خال من الأخطاء و نريد العودة
الى الحالة الرئيسية للبرنامج أو يكون التنفيذ أو ناتج التنفيذ لهذا ابرنامج ليس بالشكل
الصحيح أو بسبب عدم اكتمال عمليات الأذخال

5- ايعازات الأستثناء و تقسم هذه الأيعازات الى :

أ) ايعاز الأستثناء المفرد else : وهي عبارة تقوم بتطبيق عملية معالجة معينة و وحيدة
على حالات لا تنطبق مع الشرط المحدد الموضوع في داخل ايعاز الشرط if

ب) ايعاز الأستثناء الشاملة otherwise : فهنا سيقوم البرنامج بمعالجة حالات لا تنطبق
مع عدة شروط و التي تكون داخل عدة ايعازات الشرط if و التي لا تتواءم و الشروط التي
تعالج الحالات الخاصة و عادةً ما توضع في داخل دالة الشروط المتعددة switch
لأحتوائها اكثر من حالة تعالج عملية مشروطة تعالج حالات خاصة

ج) ايعاز الشرط المنفصل else if :

و هي ايعاز تستخدم في حالة وجود عملية معينة لا تنطبق أو ليست مرادفة تماماً مع
الشروط السابقة لها و التي يتطلب كتابة شرط معين جديد

و جميع هذه الأيعازات يجب ان تنتهي مع ايعاز انهاء end

1- جملة اذا الشرطية if statement :

تستخدم هذه الجملة للمقارنة بين كائنين حسب الشرط المعين و المحدد.

توجد مجموعة من الطرق لتنفيذ "إذا الشرطية" وهي كما يلي :

الطريقة الأولى :

```
If logical expression  
Statement 1  
else  
Statement 2  
end
```

Logical expression : شرط جملة إذا الشرطية

Statement : جملة بلغة ماتلاب

إذا كان الشرط صحيح فإنه ينفذ الجملة 1 statement وبعكسه ينفذ الجملة statement2

مثال 1:

** يجب تحديد قيمة a أولا كي يعمل البرنامج (for example a=2)

```
if (a ==4)  
    b=5;  
else  
    b=8;  
end
```

هنا إذا كانت قيمة a=4 فإن قيمة B=5 وبعكسه فإن قيمة B=8

مثال 2 : * للتنفيذ // 1. نكتب النص البرمجي في m-file

2. نحدد قيمة a, b في command window

3. نضغط على * Run

```
if (a>b)  
    disp (a)  
else  
    disp(' a is not greater than b')  
end
```

الطريقة الثانية :

إذا المتداخلة nested if

```
if logical expression1  
Statement 1
```

else if logical expression 2

Statement 2

else if expression 3

Statement 3

else

Statement 4

end

end

end

* بعدد مرات التكرار يتم تكرارها

إذا كان الشرط الأول صحيح فإنه سيقوم بتنفيذ الجملة الأولى ، و إذا كان الشرط الأول خطأ فإنه سينفذ الشرط الثاني وإذا كان الشرط الثاني خطأ فإنه سينفذ الشرط الثالث و إذا كانت الجملة الثالثة خطأ فإنه سينفذ جملة التي بعد else وهكذا.

مثال 3:

```
if a==1
b=1
elseif a==2
b=4
elseif a==3
b=9
else
b=25
end
end
```

مثال 4 : اكتب برنامج يقوم بفحص قيمة المتغيران a , b فاذا كانا اكبر من صفر اوجد حاصل ضربهما وبعكسه اطبع ان القيم ليست اكبر من صفر ؟

الحل :

```
clear; clc
a=input('a=') ; b=input('b=');
if (a>0 && b>0)
disp (' a and b values greater than zero ');
C=a * b
else
disp ( ' a and b values not greater than zero ');
end
```

س || اكتب برنامج يعمل على ايجاد حل المعادلة: $ax^2+bx+c=0$ ؟
تحل بطريقة الدستور

2- ايعاز التكرار for :

لغرض تنفيذ جملة بلغة ماتلاب اكثر من مرة نستخدم ايعاز التكرار for والصيغة العامة هي :

For variable =initial value :increment :final value
Statements
End

Variable : متغير بلغة ماتلاب
Initial value :القيمة الأولية التي سوف يبدأ بها المتغير
Increment :مقدار الزيادة التي يتم اضافتها الى القيمة الأولية
Final value : مقدار القيمة النهائية التي يصل اليها المتغير

تعمل جملة التكرار على اخذ القيم الأولية ووضعها في المتغير ثم ينفذ الجمل التي تلي ايعاز التكرار ولحين الوصول الى جملة الأنهاء end يقوم بأضافة مقدار الزيادة الى المتغير ليحص على قيمة جديدة ويتحقق ان كانت قيمة المتغير هي اصغر من اويساوي القيمة النهائية فاذا كان الجواب نعم يذهب لينفذ من الأعلى كل الجمل التي تلي ايعاز التكرار اما اذا كان الجواب كلا فانه يخرج من التكرار.

مثال 1:

```
for i=1:5  
disp ( i )  
end
```

1

عند التنفيذ سيكون الناتج كما يلي:

2

3

4

5

مثال 2 : اطبع الأعداد التي تقبل القسمة على 3 للقيم من 3 و لغاية 16 .

```
for i= 3:3 :16
```

```
disp( i )  
end
```

عند التنفيذ سيكون الناتج كما يلي:

```
3  
6  
9  
12  
15
```

مثال 3 : اطبع الأعداد تنازلياً ممن 20 ولغاية 17 ؟

```
for i=20 : -1 : 17  
disp( i )  
end
```

عند التنفيذ سيكون الناتج كما يلي:

```
20  
19  
18  
17
```

مثال 4 : اطبع الأعداد التي تقبل القسمة على 3 (من 16 ولغاية 3) ؟

```
for i= 16 : -1 : 3  
If rem(i , 3)==0  
Disp( i )  
End  
End
```

```
15  
12  
9  
6  
3
```

مثال 5 : اوجد مجموع الأعداد من 1 و لغاية 5 ؟

```
s=0;  
for i=1 : 5  
s=s+i ;
```

```
end
disp( s )
```

15

مثال 6 : اجمع الأعداد الزوجية الواقعة بين 2 و 15؟

```
s=0;
for i= 2:2 :15
s=s+i;
end
disp(s)
```

56

التكرار المتداخل nested loop

يتكون التكرار المتداخل من اكثر من تكرار و يجب ان يكون غير متقاطع ، اما عمله فانه يأخذ اول قيمة من المتغير الأول التابع الى التكرار الأول و ينفذ التكرار الثاني ابتداءً من القمة الأولية ولحين الوصول الى اخر قيمة من التكرار الثاني ثم ياخذ القيمة الثانية من التكرار الأول و ينفذ التكرار الثاني ابتداءً من القمة الأولية الى اخر قيمة وهكذا حتى ينفذ اخر قيمة من التكرار الأول مع التكرار الثاني وصولاً الى اخر قيمة من التكرار الثاني والصيغة العامة هي:

```
For i =var 1: increment : final value 1
For j= var 2 : increment :final value 2
Statements
End
End
```

مثال 7: اكتب برنامج يعمل على تكوين مصفوفة ويطلب ابعاد المصفوفة قبل تكوينها :

```
clear; clc
n=input( 'n=' ); m= input( 'm=' );
for i=1:n
for j=1:m
    A(i,j)=input( 'a=' )
end
end
```

عند التنفيذ سيطلب قيم n, m التي تمثل ابعاد المصفوفة وقيم a التي تمثل عناصر المصفوفة. فمثلا عند وضع $n=2, m=2$ سوف يطلب منا ادخال اربع قيم ل a فلو وضعنا قيم a كما يلي :

$$a=1$$

$$a=2$$

$$a=3$$

$$a=4$$

سيصبح شكل المصفوفة كالتالي:

$$A=$$

$$1 \quad 2$$

$$3 \quad 4$$

س || اكتب برنامج لأيجاد جدول الضرب لأي قيمة معطاة ؟