البنية البرمجية

يقصد بالبنية البرمجية الأدوات المستخدمة في توجيه سير البرنامج حيث أن البرنامج ينفذ التعليمات الموجودة ضمنه سطراً سطراً وباستخدام الأدوات الموجهة لسير البرنامج يمكن تجاوز بعض السطور أوالعودة إلى الوراء سطراً أوأكثر وذلك وفق الحاجة البرمجية وهذه الأدوات تتضمن أدوات الشرط وأدوات اختيار الحالة والحلقات بشكل أساسي وسوف نستعرض فيمايلي هذه الأدوات مع الأمثلة التوضيحية.

أداة الشرط If statement

تستخدم أداة الشرط If statement لتوجيه سير البرنامج لينفذ كتلة برمجية ما في حال تحقق أحد الشروط أوكتلة برمجية أخرى في حال تحقق شرط آخر ويمكن أن يحتوي الشرط على عبارة منطقية أوأكثر تستخدم فيما بينها أدوات الربط المنطقية and أو or أو not وتستخدم الرموز المبينة في الجدولين التاليين للدلالة على العلاقات المنطقية وعلاقات الربط:

يساوي	= =
لا يساوي	~ =
أصغر من أويساوي	< =
أكبر من أويساوي	> =
أصغر من	<
أكبر من	>

not	~
and	&
or	|

كذلك يمكن أن تحتوي أداة الشرط IF على عدد من الشروط يختلف حسب الحاجة لدى من يكتب البرنامج.
تأخذ هذه الأداة البنية التالية:

IF expression
Statement
ELSEIF expression
Statements
ELSE
Statements
END

وسوف نستعرض فيمايلي بعض الأمثلة عن أداة الشرط IF مع الشرح:

مثال ١:

```
user_age = input('Enter your
age':)
if user_age >= 150 | user_age < 0
msgbox('Invalid age value'!!)
end
```

هذا المثال البسيط يحوي شرطاً مركباً ففي حال أدخل المستخدم عدداً دالاً على عمره لا ينطبق مع الشرط المذكور فإن البرنامج سوف ينتقل إلى end ويتابع سيره أما إذا أدخل عدداً ينطبق مع الشرط المذكور فإن البرنامج سوف يظهر رسالة الخطأ التالية:

Invalid age value!!

مثال ٢:

```
if x(i)<0
   y(i)=x(i)-3
elseif x(i)==0
   y(i)=0
elseif x(i)>0
   y(i)=2*x(i)+1
end
```

الكتلة البرمجية السابقة تقوم بحساب التابع y = f(x) بثلاث طرق حيث يأخذ هذا التابع قواعد ربط مختلفة وفقاً لقيم المتحول x كما يلي:

$$y = f(x) = \begin{cases} x-3 & ehere \quad x<0 \\ 0 & where \quad x=0 \\ 2x+1 & where \quad x>0 \end{cases}$$

وهذا ما يدعى بالتابع ذي القطع (أوالتابع المعرف على مجالات).
يمكن أيضاً الأخذ بعين الإعتبار أن المتحول x لا يمكنه أن يأخذ قيماً أخرى غير القيم المذكورة في الشروط الثلاثة السابقة وعلى هذا يمكننا إعادة كتابة الكتلة البرمجية السابقة كمايلي:

```
 if x(i)<0
    y(i)=x(i)-3
elseif x(i)==0
    y(i)=0
else
    y(i)=2*x(i)+1
end
```

وهي تقوم بنفس العمل تماماً حيث ينفذ الأمر:

```
y(i)=2*x(i)+1
```

عند عدم تحقق الشرطين المذكورين:

```
x(i)==0 و x(i)<0
```

الحلقة For

تقوم هذه الحلقة بتنفيذ كتلة برمجية ما عدداً من المرات يتحدد في السطر الأول منها ثم تنتهي بعبارة END وتكون البنية العامة لهذه الحلقة كمايلي:

For variable = expression
Statements
END

ولنأخذ كمثال الحلقة التالية:

```
for x=1:25
disp(x)
end
```

تقوم هذه الحلقة البرمجية بتوليد عمود من القيم للمتحول x بدءاً من الواحد وانتهاءاً بالـ 25 ثم تظهر هذه القيم على نافذة الأوامر لماتلاب.

مثال آخر (تعشيش الحلقات):

```
for i=1:10
   for j=1:5
      x=i+j;
```

```
        disp(x)
    end
end
```

هذه الحلقة تقوم بحساب قيمة المتحول x على أنها مجموع قيمتي المتحولين i وj وتظهر القيم على نافذة أوامر ماتلاب وتكون الحلقة الداخلية هنا هي الكتلة البرمجية المتضمنة في الحلقة الخارجية ولا تبدأ الحلقة الخارجية بأخذ قيمة جديدة وتكرار جديد حتى تنهي الحلقة الداخلية دورتها بشكل كامل.

الحلقة while

في هذه الحلقة يتم تنفيذ كتلة برمجية ما عدداً غير محدد مسبقاً من المرات وذلك طالما أن شرط الحلقة محقق حيث يتم اختباره في كل مرة وعندما يكون غير محقق ينتقل البرنامج إلى نهاية الحلقة end ويتابع سيره فيما بعدها.
وتكون البنية العامة لهذه الحلقة كمايلي:

While expression
Statements
End

ولنأخذ الحلقة التالية مثالاً:

```
y=0;ii=0
while y<1000
```

```
    ii=ii+1;
    x(ii)=ii;
    y(ii)=x(ii)^2
end
```

في هذه الحلقة أعطينا التابع y قيمة إبتدائية هي الصفر وذلك قبل البدء بتنفيذ الحلقة ثم يتم اختبار الشرط y<1000 وبالطبع سيكون هذا الشرط محققاً في البداية ويتم حساب كل من المتحولين x وy في كل مرة ثم يختبر الشرط ثانية وهكذا تتكرر هذه الحلقة حتى يصبح الشرط غير محقق فينتقل البرنامج عند ذلك إلى نهاية الحلقة end دون المرور بمحتوى الحلقة ويتابع سيره فيما بعدها.

استخدام الأمر Break

يستخدم الأمر Break لإيقاف تشغيل حلقة for أو حلقة while قبل الوصول إلى نهايتها المفروضة وذلك عند تحقق شرط ما كما في المثال التالي:

بفرض لدينا الحلقة التالية التي تحسب قيمة تابع العاملي !x كمايلي:

```
x_fac=1;
ix=input('Enter the value:');
for k=1:ix
   x_fac=x_fac*k;
end
```

```
disp(x_fac)
```

الآن إذا أردنا من البرنامج أن يقوم بحساب تابع آخر كمايلي:

$$x\_fac\_new = \begin{cases} x! \quad where \quad x! \le 5040 \\ 5040 \quad where x! > 5040 \end{cases}$$

فإننا نقوم بإجراء التعديل التالي على الحلقة السابقة لتصبح كمايلي:

```
x_fac_new=1;
ix=input('Enter the value:');
for k=1:ix
   x_fac_new=x_fac_new*k;
   if x_fac_new>5040
      x_fac_new=5040;
      break
   end
end
disp(x_fac_new)
```

ملاحظة:
في حال وجود عدة حلقات ضمن بعضها البعض (حالة التعشيش) فإن الأمر Break يوقف تنفيذ الحلقة الأقرب إليه من الداخل فقط وتتابع بقية الحلقات سيرها كما في المثال التالي:

```
for i=1:10
   for j=1:15
      for k=1:5
         x=i*j*k;
         y=i+j+k;
         if x>y
            break
         end
      end
   end
end
```

في هذه الحلقة سوف يقوم البرنامج بالخطوات المبينة أدناه:

I=	J=	K=	X=	Y=
1	1	1	1	3
		2	2	4
		3	3	5
		4	4	6
		5	5	7
	2	1	2	4
		2	4	5
		3	6	6
		4	8	7

هنا سوف يقوم البرنامج بتنفيذ الأمر break ويخرج من حلقة k دون أن يعطي لـ k قيمة جديدة k=5 كما هومفروض في سير الحلقة ولكنه لا يخرج من حلقة j بل يعود إليها ويعطي لـ j قيمة جديدة j=3 ويتابع كمايلي:

I=	J=	K=	X=	Y=
1	3	1	3	5
		2	6	6
		3	9	7

وهنا سوف يقوم بتنفيذ break ثانية ويعطي لـ j قيمة جديدة j=4 وهكذا دواليك.
أي أن الأمر Break يعتبر محلياً بالنسبة لحلقته ولا يتدخل في بقية الحلقات.

أداة الاختيار Switch

تستخدم هذه الأداة الهامة جداً لجعل البرنامج يقوم بعمل مختلف كلما أخذ متحول ما قيمة مختلفة وهي تغني عن أداة الشرط IF في حال أن الشروط كلها مطبقة على نفس المتحول؛ وتكون البنية البرمجية لهذه الأداة كمايلي:

```
Switch switch_expression
Case case_expr
Statements
```

Case case_expr
Statements

Otherwise
END

ولتوضيح هذه ا لأداة سوف نأخذ المثال التالي:

ليكن لدينا التابع الذي قاعدة ربطه هي:

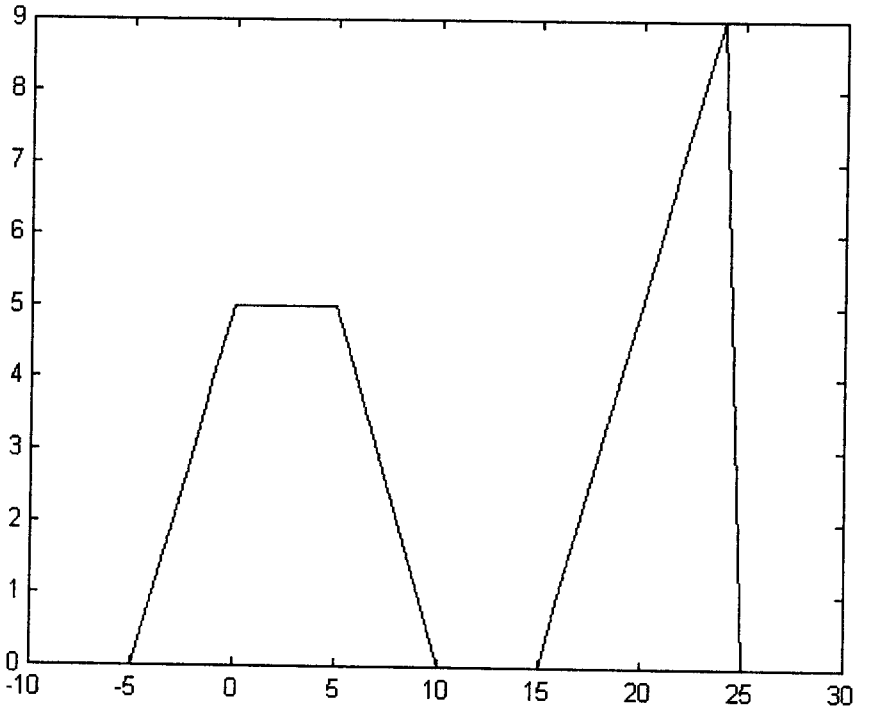
$$y=\begin{cases} 0 & where \quad x<-5 \\ x & where \quad -5\le x<0 \\ 10 & where \quad 0\le x<5 \\ -x+10 & where \quad 5\le x<10 \\ 0 & where \quad 10\le x<15 \\ x-15 & where \quad 15\le x<25 \\ 0 & where \quad x\ge 25 \end{cases}$$

يمكن تعريف هذا التابع في ماتلاب باستخدام أداة الاختيار switch كمايلي:

```
x=[-10:30];
```

```
for k=1:length(x)
switch x(k)
case {-5,-4,-3,-2,-1}
   y(k)=x(k)+5
case{0,1,2,3,4}
   y(k)=5
case {5,6,7,8,9,10}
   y(k)=-x(k)+10
case
{15,16,17,18,19,20,21,22,23,24}
   y(k)=x(k)-15
otherwise
   y(k)=0
end
end
plot(x,y)
```

وعند رسم هذا التابع نحصل على الشكل التالي:



مثال محلول:

بفرض لدينا المصفوفة:

$$X = [5 \quad -2 \quad 1 \quad 0 \quad 3 \quad -4 \quad 2 \quad 7 \quad -1]$$

١. أكتب برنامجاً يقوم بترتيب عناصر هذه المصفوفة تصاعدياً.
٢. طور البرنامج ليقوم بترتيب أي مصفوفة سطرية يدخلها المستخدم.

الحل:

```
x=[5,-2,1,0,3,-4,2,7,-1];
for k=1:length(x)
   for l=2:length(x)
      if x(l-1)>x(l)
         x_temp=x(l-1);
         x(l-1)=x(l);
         x(l)=x_temp;
      end
   end
end
disp(x)
```

يتم التطوير المطلوب بتعديل السطر الأول فقط ليصبح:

```
x=input('Enter the matrix you
want to rearrange:');
```

تنويه:

في ماتلاب يمكن ترتيب عناصر المصفوفة مباشرة بواسطة التابع sort ولكن المثال هنا للتدريب فقط.