كلية العلوم

القسم : **الفيزياء**

السنة : **الثانية**

المادة : لغات البرمجة١

المحاضرة : ٥+٦ /ن+ع/دكتور

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

كلية العلوم ، كلية الصيدلة ، الهندسة التقنية

2026

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

بنى التحكم التكرارية control structures

1. البنية التكرارية while (حلقة while)
2. البنية التكرارية do \ while
3. البنية التكرارية for

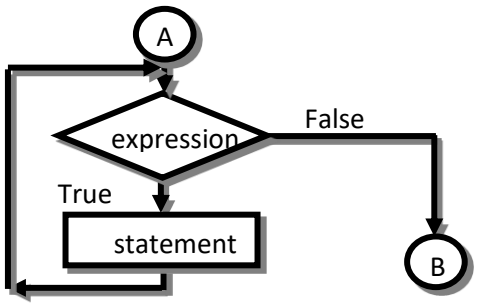
1.**البنية التكرارية while (حلقة while) :** تسمى بالبنية التكرارية ذات الشرط المسبق.

تملك الصيغة العامة التالية :

```
while ( expression ) statement ;
```

تقدر قيمة expression فإذا كانت true فيتم تنفيذ statement والتي قد تكون تعليمة مفردة أو مركبة، ثم يعاد تقدير قيمة expression مرة أخرى، فإذا كانت true تنفذ statement مرة أخرى وهكذا تتكرر هاتان الخطوتان حتى تصبح قيمة expression مساوية false، بمعنى أنه يتم تكرار تنفيذ statement طالما أن قيمة expression مساوية true وعندما تصبح تلك القيمة false يتم الخروج من البنية while.

يتم إختبار صحة الشرط قبل تنفيذ الحلقة، ومن هنا أتت تسميتها بالبنية ذات الشرط المسبق. ولا بد أن يحتوي جسم الحلقة while على تعليمة تجعل الشرط غير محقق حتى يتم الخروج من هذه الحلقة أي بمعنى منع تكرار حلقة لا نهائية. ويبين الشكل المخطط الصندوقي للبنية while .



مثال 1 :

```
using namespace std;
#include<iostream >
main()   {int x;
      cout<<"enter number:”;
      cin>>x;
      while(x>5) { cout<<'*' ;
                   x--;}
      cout<<endl;
      cout<<"++"<<endl; }
```

إذا تم إدخال `x = 8` نجد أن شرط الحلقة `while` محقق(`8>5`) وبالتالي ستنفذ التعليمة بعد الشرط (يطبع *) ثم يتم إنقاص `x` بمقدار `1` فيصبح `x = 7` ويعاد اختبار الشرط (`7 > 5`) وباعتباره محققاً تتم طباعة * وإنقاص قيمة `x` بمقدار `1` وهكذا حتى يصبح `x = 5` فيختل الشرط ويتم الخروج من حلقة `while` وتتم طباعة ++ :

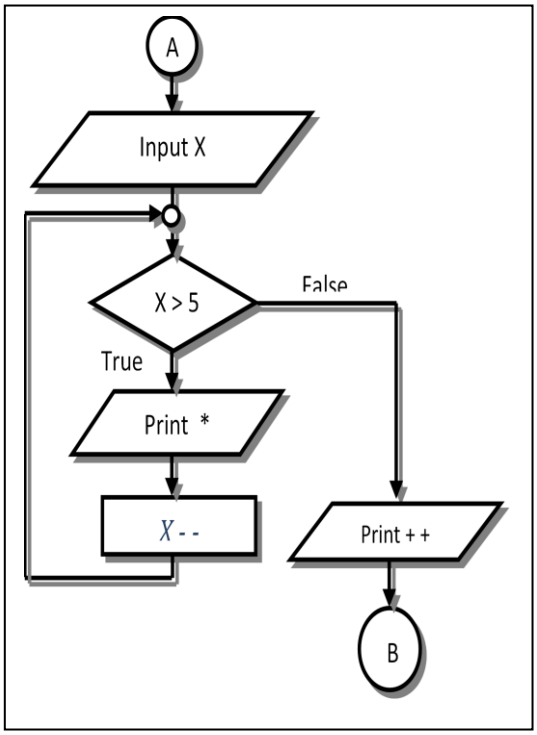
بينما لو تم إدخال `x = 2` سيتم اختبار الشرط ويكون غير محقق وبالتالي لن تنفذ التعليمة بعد الشرط ويكون خرج البرنامج :

```
enter number: 8
***
++
```

```
enter number: 2

++
```

ويبين المخطط الانسيابي خوارزمية التنفيذ للمثال السابق:



مثال 2 :

```
using namespace std;
#include<iostream >
      main(){int pro=2;
      while(pro<=60)
            pro=2*pro;
      cout<<pro<<endl; }
```

في هذا البرنامج سيختل الشرط عندما يصبح `pro = 64` وسيتم الخروج من `while` وطباعة القيمة 64.

Pro Old	Pro New
2	4
4	8
8	16
16	32
32	64

خرج البرنامج:

```
64

Press any key to
```

مثال 3 :

بفرض أننا نرغب بطباعة الأعداد المحصورة بين العدد المدخل والعدد 10 مفترضين أن العدد المدخل أقل من 10.

```
using namespace std;
#include<iostream >
main(){
      int counter;
      cout<<"enter number :";
      cin >> counter;
      cout<<"the number between "<<counter
            <<" and 10 are :"<<endl;
      while (counter <= 10) {
                  cout << counter <<"  " ;
            ++counter; }
      cout<<endl; }
```

خرج البرنامج:

```
enter number :4

the number between 4 and 10 are
:

4 5 6 7 8 9 10
```

هذا البرنامج يعطي الأعداد المحصورة بين العدد المدخل والعدد 10 إذا كان العدد المدخل أصغر أو يساوي العدد 10.

مثال 4: تطوير المثال السابق ليعطي خرجاً مهما يكن العدد المدخل، فهو يعالج الحالتين أي إذا كان ذلك العدد أكبر من 10 أو أصغر منه.

```
using namespace std;
#include <iostream >
main(){
      int counter;
      cout<<"enter number : ";
      cin >> counter;
      cout<<"the number between "<<counter
            <<" and 10 are :"<<endl;
      if (counter <= 10)
            while (counter <= 10)
                  {
                  cout << counter <<"  "; ++counter ;
                  }
      else
            while(counter >= 10)
                  {
                  cout << counter <<"  ";  --counter ;
                  }
cout<<endl; }
```

إذا كان العدد المدخل (2<10) سيكون خرج البرنامج:

```
enter number : 2
 the number between 2 and 10 are:
2 3 4 5 6 7 8 9 10
```

إذا كان العدد المدخل(18 >10) سيكون خرج البرنامج:

```
enter number : 18
the number between 18 and 10 are:
18 17 16 15 14 13 12 11 10
```

مثال 5:

ليكن المطلوب حساب متوسط علامات الطلاب الناجحين في مقرر ما وعدد الطلاب الراسبين في ذلك المقرر. سيكون البرنامج بالشكل التالي :

```
using namespace std;
#include <iostream >

main(){float average; int counter،grade،total،count1; total=0;counter=0;count1=0;

      cout << "Enter grade،-1 to end";
      cin >> grade;
            while (grade != -1) {
                  if (grade >= 60) {
                              total = total + grade;
                        counter = counter + 1; }
                  else
                        { count1 = count1+1; }

                  cout << "Enter grade، -1 to end ";

            cin >> grade; }
            if (counter != 0) {
            average = (float) total / counter;
            cout << "Class average is " <<average << endl; }
      else
            cout << "No grades were entered GE 48" << endl;
            if (count1 != 0) {
            cout << "the number of the failed student is " ;
            cout<<count1<< endl; }
      else
cout << "No grades were entered LT 48 " << endl; }
```

```
Enter grade,-1 to end :80
Enter grade,-1 to end :50
Enter grade,-1 to end :43
Enter grade,-1 to end :23
Enter grade,-1 to end :79
Enter grade,-1 to end :65
Enter grade,-1 to end :-1
Class average is 68.5
```

إذا كانت جميع القيم المدخلة أصغر من 48 سيطبع البرنامج رسالة تدل على ذلك، كذلك الأمر فيما لو كانت جميع القيم المدخلة أكبر من 48.

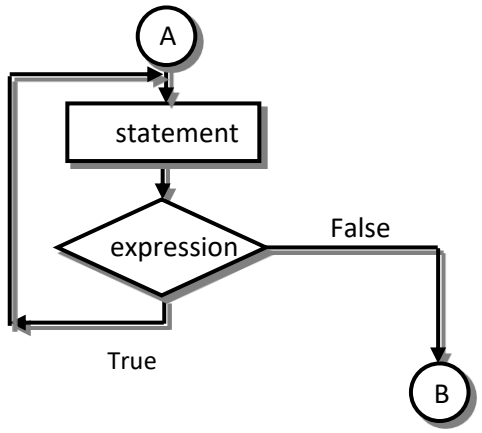
2- البنية التكرارية do \ while

و هي تسمى بالبنية التكرارية ذات الشرط اللاحق، و هي تملك الصيغة العامة التالية:

```
do statement while ( expression ) ;
```

تنفذ التعليمة statement (مفردة أو مركبة) ثم يتم اختبار الشرط expression فإذا كانت نتيجة هذا الاختبار true يعاد تنفيذ التعليمة statement ، ثم اختبار الشرط وهكذا دواليك، بمعنى أنه يتكرر تنفيذ التعليمة statement حتى يختل الشرط expression.

الشكل يبين المخطط الصندوقي للبنية do\while .



المخطط الصندوقي للبنية do\while

تجدر الإشارة أنه في البنية do \ while تنفذ التعليمة statement قبل اختبار الشرط ومن هنا أتت تسميتها بالبنية ذات الشرط الملحق.

مثال 6:

اكتب برنامجاً يسمح بحساب معدل عشر قيم يتم إدخالها من لوحة المفاتيح معتمداً على البنية do\while.

```
using namespace std;
#include <iostream >
      main(){int counter,grade,total,average; total = 0;
      counter = 1;
      do
                  { cout<<"Enter grade:";
                  cin>>grade;
                  total=total+grade;
                  counter = counter + 1; }
      while(counter <= 10);
            average = total / 10;
      cout<<"Class average is "<<average<<endl; }
```

مثال 7–:

اكتب برنامجاً يسمح بحساب معدل الطلبة الناجحين، وعدد الطلبة الراسبين في مادة ما بالإعتماد على البنية do\while.

ويتم التحكم بعدد القيم المدخلة بإدخال الرقم 1– .

```
using namespace std;
#include<iostream >
main(){

      float average;

      int counter،grade،total،count1;

      total=0;counter=0;count1=0;

      cout<<"Enter grade،-1 to end ";


      cin >> grade;
      do
            {
                  if (grade >= 60)
                        {
                              total = total + grade;
                              counter = counter + 1;}
                  else
                        {
                              count1 = count1+1;
                        }

                  cout << "Enter grade، -1 to end ";

                  cin >> grade;
            }while (grade != -1) ;
      if (counter != 0)
            {
            average = (float) total / counter;
            cout << "Class average is " <<average << endl;}
      else
            cout << "No grades were entered GE 60" << endl;
      if (count1 != 0){
            cout << "the number of the failed student is";
            cout<<count1<< endl;
      }
      else
       cout << "No grades were entered LT 48 " << endl; }
```

3- البنية التكرارية for :

تسمى بالبنية ذات العدد المعروف من مرات التكرار.

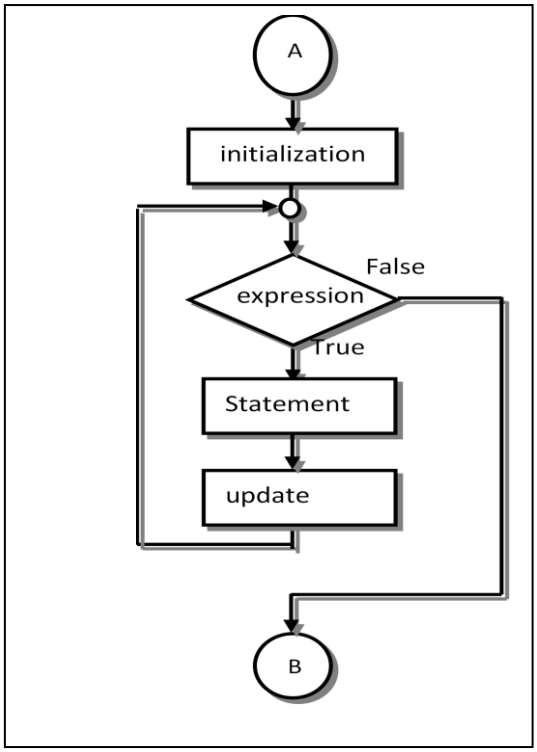
تملك الصيغة العامة التالية :

```
for( initialization ; continuation condition ;update ) statement ;
```

يتم التحكم في هذه البنية التكرارية بثلاثة أجزاء منفصلة:

- القيمة الابتدائية initialization.
- شرط الاستمرار continuation condition.
- القيمة الجديدة update.

آلية تنفيذ الحلقة :



يأخذ متحول الحلقة قيمة هي القيمة الابتدائية initialization ويتم اختبار شرط استمرار الحلقة، فإن كان محققاً تنفذ التعليمة statement مفردة كانت أم مركبة، ثم يعطى متحول الحلقة قيمة جديدة ويعاد اختبار شرط استمرار الحلقة فإن كان محققاً تنفذ statement، ثم يعاد إعطاء متحول الحلقة قيمة جديدة وهكذا يستمر تنفيذ statement حتى يختل شرط استمرار الحلقة.

القيمة الابتدائية وشرط استمرار الحلقة والقيمة الجديدة يمكن أن يكونوا فارغين بدون أية قيم.

يبين الشكل المخطط الصندوقي للبنية for.

جميع البرامج التي تستخدم فيها حلقة while يمكن أن تستخدم فيها حلقة for بدلاً من while فحلقتي for وwhile التاليتان متكافئتان :

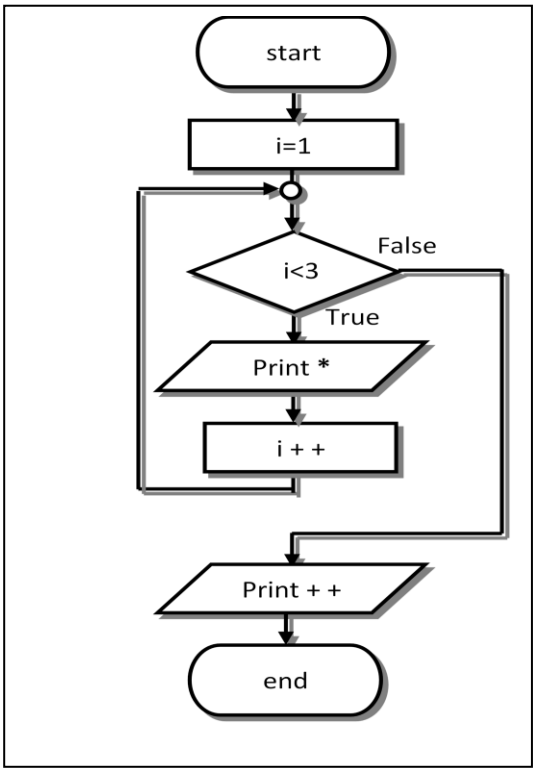
```
for(int i=1;i<=10;i++) statement;
==========================
int i=1;
while(i<=10) { statement; i=i+1; }
```

مثال 8 :

```
#include<iostream.h>
void main()
{
      for(int i=1 ; i<3  ; i++ )
            cout<<"*" ;
      cout<<endl;
      cout<<"++"<<endl;
}
```

```
**

++
```



ففي هذا المقطع البرمجي يأخذ المتحول الصحيح للحلقة قيمة ابتدائية مساوية للعدد 1 وتكون نتيجة اختبار شرط استمرار الحلقة (1<3) true لذلك ستنفذ التعليمة cout<<'*'; ويتم طباعة المحرف * ، ثم تزاد بعد ذلك قيمة المتحول i وقيمة الزيادة هنا خطوة واحدة (يمكن أن تزاد بأي قيمة صحيحة) ، ثم يعاد اختبار شرط استمرار الحلقة (2 < 3) والذي تكون قيمته true ويطبع * ، ، ثم تزاد قيمة المتحول i فتصبح 3 وهنا يختل شرط استمرار الحلقة ويتم الخروج من for. ويكون خرج البرنامج:

مثال 9 :

```
using namespace std;
#include<iostream >
main()
{
      for(int i=2;i>0;i--)
            cout<<"*";
      cout<<endl<<"++"<<endl;
}
```

في هذا المثال القيمة الابتدائية للحلقة أكبر من القيمة النهائية والعداد i يتم إنقاصه حتى يصل إلى القيمة النهائية.

سيعطي هذا البرنامج خرجاً كالبرنامج السابق.

مثال 10:

البرنامج التالي يدخل متتالية من الأعداد الصحيحة تنتهي بالعدد الصحيح صفر ويطبع أكبر الأعداد وأصغرها في المتتالية وذلك بالاعتماد على البنية for.

وسنكتب البرنامج أولاً بالاعتماد على البنية while.

```
using namespace std;
#include<iostream >
main(){int n,min,max;

      cout<<"enter integers،";

      cout<<" terminate input with 0:\n";
      cin>>n;
      min=max=n;
      while(n!=0)
            {
                  if(n<min)
                        min=n;
                  else
                  if(n>max)
                        max=n;
            cin>>n;
            }
      cout<<"min="<<min;
      cout<<"  "<<"max="<<max<<endl; }
```

```
enter positive integers, terminate input with 0:

2 4 7 1 9 4 0

min=1  max=9
```

في البرنامج السابق أعطي للمتغيرين max،min قيمة ابتدائية هي قيمة المتغير n، ذات القيمة الابتدائية أعطيت للمتغيرين في حلقة for من البرنامج التالي.

تنفذ حلقة while في البرنامج الأول حتى يدخل الرقم صفر.

وشرط استمرار حلقة for في البرنامج الثاني أن يدخل عدد مغاير للصفر.

مثال 11:

اكتب برنامجاً يطبع مكعبات الأعداد المدخلة إلى أن يدخل العدد صفر بالاعتماد على البنية for.

```
using namespace std;
#include<iostream >
main(){int n;

      cout<<"Enter a positive integer،";
      cout<<"0 to end:";
      cin>>n;
      for(  ; n!=0 ; )
      {     cout<<n<<" cubed is:";
            cout<<n*n*n<<endl;

            cout<<"Enter a positive integer،";
            cout<<" 0 to end:";
      cin>>n; }}
```

خرج البرنامج:

```
Enter a positive integer,0 to end :5

5 cubed is:125

Enter a positive integer,0 to end :6

6 cubed is:216

Enter a positive integer,0 to end :2

2 cubed is:8

Enter a positive integer,0 to end :3

3 cubed is:27
```

مثال 12 :

اكتب برنامجاً يطبع القيم بين العدد المدخل والعدد 10 سواء كان ذلك العدد أكبر أو أصغر من العدد 10؟

```
using namespace std;
#include<iostream >
main(){int counter;

      cin >> counter;
      if (counter <= 10)
            for(counter=counter;counter<=10;counter++)
                  cout<<counter<<endl;
      else
            for(counter=counter;counter>=10;counter--)
      cout << counter << endl; }
```

مثال 13:

اكتب برنامجاً يقوم بما يلي :

1- طباعة مجموع الأعداد المحصورة بين العدد 1 والعدد 100.

2- طباعة مجموع الأعداد الزوجية المحصورة بين العدد 1 والعدد100.

3- طباعة مجموع الأعداد الزوجية التي تقبل القسمة على العدد 5 والمحصورة بين 1و100.

4- طباعة مجموع الأعداد المحصورة بين عدد مدخل والعدد 10.

```
using namespace std;
#include <iostream >
main(){int sum=0,sum1=0, sum2=0,sum3=0; int sum0=0,sum4=0;
      cout<<"the sum of the number between 1 and 100 is:";
      for(int number=1;number<=100;number +=1)
            sum += number;
      cout <<sum<<endl;
      cout<<"the sum of the even number between 1 and 100 is:";
      for(int number2=2;number2<=100; number2 +=2)
            sum2 += number2;
      cout<<sum2<<endl;
      cout<<"the sum of the even number"
            <<" between 1 and 100 that divide 5 is:";
      for(int number1=2;number1<=100;number1 +=2)
            if(number1%5 == 0)
                  sum1+=number1;
      cout<<sum1<<endl;
      cout<<" enter number: ";
      cin >>sum0;
      cout<<" the sum of the numbers between "
              <<sum0<<"and 10 is:";
```

```
if(sum0>=10)
{
for( number2=sum0;number2>=10;number2 -=1)
      sum4 += number2;
      cout<<sum4<<endl; }
else {
      for(number2=sum0;number2<=10;number2 +=1)
            sum4 += number2; cout<<sum4<<endl; } }
            ************************************
```

انتهت المحاضرة