

الحلقات التكرارية (جمل التكرار) Iteration Statements

تستخدم الحلقة التكرارية في تكرار تنفيذ تعليمة أو مجموعة من تعليمات البرنامج أكثر من مرة وهي نوعان :

1. الحلقات المحددة (غير المشروطة)

في هذا النوع يتم تحديد عدد مرات التكرار بواسطة رقم صحيح ويبنى هذا النوع على مفهوم العداد counter وهو عبارة عن متغير يأخذ قيمة ابتدائية Initial Value تتغير باستمرار بمعدل معين بالزيادة أو النقصان إلى أن تصل إلى القيمة النهائية Final value ولتصميم هذه البنية نستخدم الجملة for()

الصيغة العامة لحلقة for

for(counter=initial value; condition ; step)

حيث:

الشرط condition :- هو عبارة عن جملة تقارن بين قيمة العداد الحالية والقيمة النهائية .

Step :- تعبير رياضي يوضح التغير في العداد في كل دورة.

والشكل العام للحلقة

```
for()  
statement;
```

أو

```
for()  
{  
    Statement 1;  
    .  
    .  
    Statement n;  
}
```

أمثلة على حلقة for

1 اكتب برنامج لطباعة الأرقام من 1 إلى 10 ومربعاتها

```
#include<iostream.h>  
int main()  
{  
    int i;  
    for(i=1;i<=10;i++)  
        cout<<i<<"\t"<<i*i<<"\n";  
    return 0;  
}
```

2 اكتب برنامج لحساب متوسط القيم من 123_____477

```
#include<iostream.h>
int main()
{
int i,n=0,sum=0;
float avg;
for(i=123;i<=477;i++)
{
n++;
sum+=i;
}
avg=sum/n;
cout<<"average="<<avg;
return 0;
}
```

3 اكتب برنامج لحساب مضروب n حيث $n!=1 \times 2 \times 3 \times 4 \times \dots \times n$ أو $n!=n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$

أ- الطريقة الأولى :

```
#include<iostream.h>
int main()
{
int n,f=1,i;
cout<<"enter n =";
cin>>n;
for (i=1;i<=n;i++)
f*=i;
cout<<"factorial="<<f;
return 0;
}
```

ب- الطريقة الثانية

```
#include<iostream.h>
int main()
{
int n , f ,i ;
cout<<"enter n:" ;
```

```
cin>>n ;
f=1;
for(i=n ; i>=1 ; i--)
f*=i ;
cout<<"factorial ="<<f ;
return 0;
}
```

4 أكتب برنامج لحساب a^b (باستخدام عملية *)

```
#include<iostream.h>
int main()
{
int a , b , z , i ;
cout<<"enter a , b : " ;
cin>>a>>b;
z=1;
for(i=1 ; i<=b ; i++)
z*=a ;
cout <<"a^b="<<z;
return 0;
}
```

5 أكتب برنامج لإيجاد جمع المتسلسلة التالية :

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

حيث n يتم إدخالها من قبل المستخدم

```
#include<iostream.h>
int main()
{
float n,i, sum=0;
cout<<"enter n:";
cin>>n ;
for (i=1 ; i<=n ; i++)
sum+=1/i;
cout<<"sum="<<sum ;
return 0;
}
```

6 - أكتب برنامج لطباعة الشكل التالي :

```
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

باستخدام الحلقات المتداخلة

```
#include<iostream.h>
int main()
{
    int i , j ;
    for (i=1 ; i<=10 ; i++)
    {
        for ( j=1 ; j<=i ; j++)
        {
            cout<<"*" ;
        }
        cout<<"\n";
    }
    return 0;
}
```

7 - أكتب برنامجاً يطبع جدول ضرب الأعداد من 1 حتى 10 باستخدام الحلقات المتداخلة :

```
#include<iostream.h>
int main()
{
    int i , j ;
    for (i=1 ; i<=10 ; i++)
    {
        cout<<"\n";
        for ( j=1 ; j<=10 ; j++)
        cout<<i<<"*"<<j<<"="<<i*j<<"\t" ;
    }
    return 0;
}
```

Jump Statements **جمل القفز**

break **جملة (1)**

تستخدم لإيقاف الحلقة التكرارية عندما يتحقق شرط معين

مثال :-

```
// break loop example
#include<iostream.h>
int main()
{
    int n ;
    for (n=10 ; n>0 ; n--)
    {
        cout<<n<<" , " ;
        if (n==3)
        {
            cout<<"countdown aborted ";
            break ;
        }
    }
    return 0;
}
```

المخرج :

10 , 9 , 8 , 7 , 6 , 5 , 4 , 3
countdown aborted

حيث أن الحلقة تتوقف عندما (n=3)

continue **جملة (2)**

تستخدم لاستبعاد شرط معين من الحلقة التكرارية

مثال :-

```
// continue loop example
#include<iostream.h>
int main()
{
    for(int n=10 ; n>0 ; n--)
    {
```

```
if (n==5)
continue ;
cout <<n<<" , " ;
}
return 0;
}
```

المخرج :

10 , 9 , 8 , 7 , 6 , 4 , 3 , 2 , 1

حيث أنه تم استبعاد القيمة 5 (الشرط) من المخرج

(3) جملة goto

تستخدم لتحويل سير تنفيذ البرنامج إما تحت شرط معين أو بدون شرط ومهمتها توجيه تنفيذ البرنامج والانتقال إلى جملة أخرى

مثال 1 - : جملة goto المشروطة

```
// goto example
#include<iostream.h>
int main()
{
int n=10 ;
loop :
cout<<n<<" , " ;
n-- ;
if(n>0) goto loop ;
return 0;
}
```

المخرج :

10 , 9 , 8 , 7 , 6 , 5 , 4 , 3 , 2 , 1

مثال 2 - : جملة goto غير المشروطة

```
#include<iostream.h>
int main()
{
int a , b , c ;
a=1 ;
b=2 ;
```

```
first : c=a*b ;  
cout<<a<<"\t"<<b<<"\t"<<c<<"\n";  
a++ ;  
goto first ;  
return 0;  
}
```

يطبع هذا البرنامج قيم المتغيرات :

a=1 b=2 c=2

a=2 b=2 c=4

ويستمر في التنفيذ إلى ما لانهاية

2. الحلقات المشروطة (غير المحددة)

هذا النوع غير محدد الدورات وينتهي تنفيذ الحلقة عندما يتحقق شرط معين ، أو بالأصح تستمر

الحلقة في التنفيذ طالما الشرط صحيح

(a) حلقة while()

ويمكن أن تؤدي دور الجملة for() .

الصيغة العامة لجملة while

```
while (condition)  
{  
statement 1 ;  
statement 2 ;  
.  
.  
.  
statement n ;  
}
```

مثال :- حوّل جملة : for (i=1 ; i<=10 ; i++) إلى جملة while

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

يتم استخدام while() كدالة تكرارية مفتوحة لتعريف متغير يأخذ قيمة ابتدائية معينة ثم بعد ذلك وضع الشرط طالما قيمة المتغير ثابتة استمر . ويجب إعطاء المستخدم إمكانية تغيير هذه القيمة بعد كل دورة بواسطة دالة الإدخال

أمثلة على حلقة while ()

1 - أكتب برنامج لطباعة c++ language أكثر من مرة :

```
#include<iostream.h>
int main()
{
char ok='y' ;
while (ok=='y')
{
cout<<"\n c++ language";
cout<<"\n continue [y / n]" ;
cin>>ok ;
}
return 0;
}
```

2 - أكتب برنامج لطباعة الأعداد من 1 إلى n بشكل تنازلي :

```
#include<iostream.h>
int main()
{
int n ;
cout<<"enter the starting number : " ;
cin>>n;
while (n>0)
{
cout<<n<<"\n" ;
-- n ;
}
return 0;
}
```

3 - أكتب برنامج لإيجاد مضروب الأعداد من 1 إلى 10 حيث مضروب العدد n يحسب كالآتي

$$n! = n(n-1)(n-2) \dots \times 3 \times 2 \times 1$$

```
#include<iostream.h>
int main()
{
long fact ;
int i , k ;
cout<<" i factorial\n";
```



```
i=1;
while (i<=10)
{
fact=1 ;
k=1 ;
while(k<=i)
{
fact*=k;
k++ ;
}
cout<<i<<"\t" <<fact<<"\n";
i++ ;
}
return 0;
}
```

4 أكتب برنامج لإيجاد متوسط n من القيم الصحيحة المدخلة من قبل المستخدم

```
#include<iostream.h>
int main()
{
int n , i , x, sum ;
float avg ;
sum=0;
i=1;
cout<<"enter n:";
cin>>n;
while (i<=n)
{
// enter 10th numbers
cout<<"enter x"<<i<<"\n";
cin>>x ;
sum+=x;
i++ ;
}
avg=sum/n;
cout<<"The average ="<<avg<<"\n";
return 0;
}
```

(b) حلقة do-while :

هذه الجملة مشابهة لجملة while وتختلف عنها في أمرين :

○ تبدأ بالعملية أولاً ويتم التحقق من الشرط في أسفل الجملة.

○ لا بد من تنفيذ الجمل الموجودة بين do و while مرة واحدة على الأقل حتى ولو كان الشرط غير متحقق.

الصيغة العامة لها :

```
do
{
statement 1;

.

.
statement n ;
}while(condition) ;
```

أمثلة على حلقة do-while

1 - اكتب برنامج يقوم بإيجاد مربعات الأعداد من 1 إلى 10

```
# include < iostream.h >
main( )
{
int i;
i=1; // i is a counter
do
{
cout << i*i<<"\n";
i++;
}while ( i<=10);
return 0;
}
```

2 - اكتب برنامج لحساب حاصل ضرب جميع الأعداد الزوجية من 2 إلى 10

```
# include < iostream.h >
int main( )
{
int i, product;
product=1;
```

```
i=2;
do
{
product * = i;
i += 2;
} while (i<=10);
cout<<" product = " << product;
return 0;
}
```

3 - أعد كتابة الآتي باستخدام جملة do-while

a-

```
int i;
for (i=1;i<=10;i++)
cout <<"i="<<i<<" \n";
```

الحل

```
int i;
i=1;
do
{
cout<<" i="<<i<<"\n";
i++;
} while (i<=10);
```

b-

```
short a;
a=1;
for ( ; a++<=10;)
cout<<a<<" \n ";
```

الحل

```
short a;
a=1;
do
{
cout<<a<<" \n ";
a++;
} while (a<=100);
```