* عبارات أو جمل التكرار Loops

نستخدم الـ Loops لتكرار مجموعة معينة من الجمل عدد معلوم من المرات ونستخدم لهذا التكرار العداد counter

1) حلقة for ؛

الصيغة العامة:

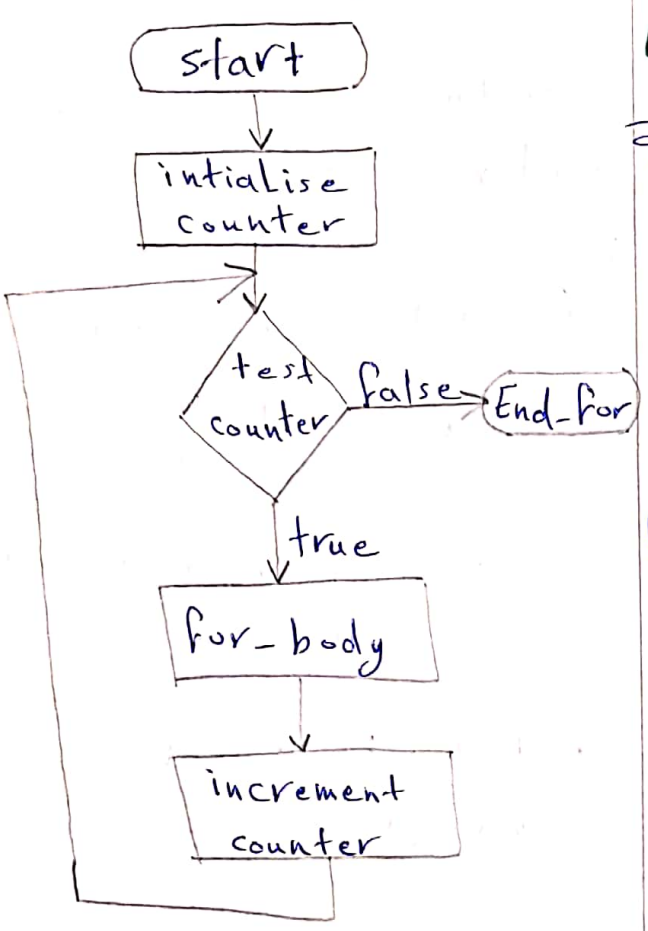
```
for(intialiaLise counter ; test counter ; increment counter)
{
   do this;
   and this;
   ;
}
```

intialise counter: تعطي تعريف العداد وتعطيه قيمة ابتدائية

test counter: اختبار العداد

increment counter: زيادة (i) وتقصان العداد

الخطط التدفقي:



* مثال: اكتب برنامج يقوم بطباعة أرقام صحيحة من 0-7

```
#include<iostream.h>
void main()
{
 for(int i=0;i<=7;i++)
 cout<<"i="<<i<<endl;
}
```

* اكتب برنامج لطباعة الأعداد الزوجية من 4-40

```
#include<iostream.h>
void main()
{
  for(int i=4;i<=40;i+=2)
cout<<i<<endl; }
```

* اكتب برنامج يطبع ! وكان العدد زوجي أم فردي من 0 → 10

```
#include<iostream.h>
void main()
{
for(int i=0;i<=10;i++)
{ if(i%2==0)
cout<<i<<" is even"<<endl;
else
cout<<i<<"is odd"<<endl;}}
```

* اكتب برنامج يقرأ عشرة أرقام ويجمع الأعداد الزوجية فقط.

```
#include<iostream.h>
void main()
{
int x, sum=0;
for(int i=1;i<=10;i++)
{ cin>>x;
 if(x>0)
 sum=sum+x;}
cout<<"sum="<<sum;}
```

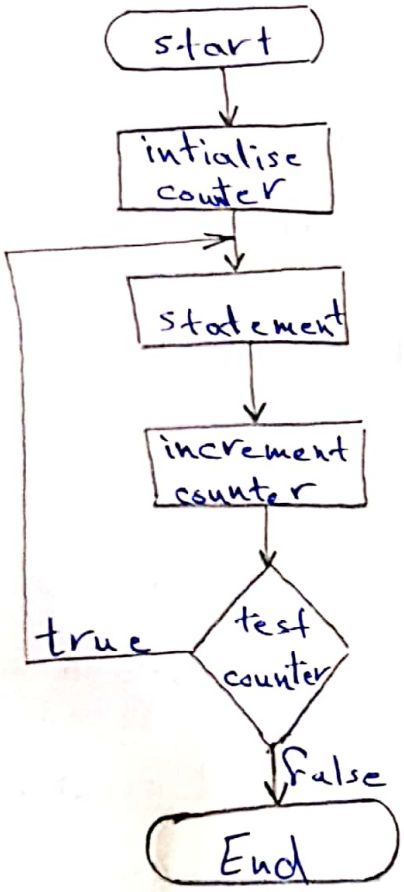
* اكتب برنامج يدخل درجات 8 مواد ويقوم بحساب المعدل الكلي وطباعته.

```
#include<iostream.h>
void main()
{
int degree, sum=0, average=0;
for(int i=1; i<=8; i++)
{
  cin>>degree;
  sum=sum+degree;
}
 average=sum/8;
 cout<<"averag is"<<average;
}
```

التفريع عنا:

```
initialise counter;
do {
 statement;
 increment counter;
} while (test_counter);
```

المخطط التدفقي:



* اكتب برنامج لطباعة مربع الأعداد من 1 → 5

```
#include<iostream.h>
void main()
{
 int i=1;
 while(i<=5)
 { cout<<i<<"\t"<<i*i<<endl;
  i++; }
}
```

3) عبارة التكرارية (Do-while)

هي عبارة تكرارية تستخدم لتكرار مجموعة محصورة بين قوسيه لعدد من المرات وتكون مشابهة لعبارتي التكرار (for) و (while) ولكنها تختلف عنهم في أنها ستنفذ ما بين قوسيه مرة واحدة على الأقل حتى لو كان الشرط غير محقق بعكس عبارتي (for) و (while) اللتان لن تنفذا أي شيء في حالة عدم تحقق الشرط.

س - اكتب برنامج يقوم بطباعة الاعداد العكسية من 4 → 0

```
#include<iostream.h>
 Void main()
{
 int i=4;
 while(i>=0)
{ cout<<i<<"\t";
  i--;}
}
```

* اكتب برنامج يقوم بجمع الاعداد الفردية بين 1-100 وطباعة المجموع.

```
#include<iostream.h>
 Void main()
{
  int i=1, sum=0;
  while(i<100)
  { sum=sum+i;
   i=i+2;}
cout<<"sum="<<sum;}
```

س - اكتب البرنامج 1 - محاضرة 9

الطباعة التكرارية الثانية من عبارات التكرار - وهي:

2) عبارة (while) التكرارية:

هي عبارة تكرارية تستخدم لتكرار الخطوات البرمجية المحددة لعدد من المرات وتكون مشابهة تماماً لعبارة التكرار (for) وتختلف عنها في طريقة التصريح كالتالي:

```
intialise counter;
while(test_counter)
{
 statement;
increment counter;}
```

حيث كما نلاحظ أن intialise counter هي القيمة الابتدائية التي سيبدأ العداد العمل

* test counter: هو اختبار العداد وعندما تصبح قيمته غير محققة لشرط التوقف (while) سوف تخرج من عبارة

* statement: هي الخطوات البرمجية التي ستنفذ عدد من المرات مادام الشرط محقق

* increment: مقدار الزيادة أو النقصان في قيمة العداد counter في كل دورة

* اكتب برنامج لايجاد جدول الضرب لاي عدد تدخله ثم طباعة هذا الجدول

```
#include<iostream.h>
void main()
{
 int i=1, x;
cin>>x;
do{
cout<<i*x<<endl;
}while(i<=10);
}
```

---

* اكتب برنامج يقوم بجمع الاعداد المدخلة ومن ثم يتوقف عن العمل ويقوم بطباعة مجموع ادخال الرقم صفر.

```
#include<iostream.h>
void main()
{
 int x, sum=0;
```

```
do{
 cin>>x;
 sum=sum+x;
}while(x!=0);
cout<<"sum="<<sum;}
```

س١: اكتب برنامج لـ محاكاة ١٠ * استخدام عبارتي Break, continue في الحلقات التكرارية:

عبارة Break: تستخدم هذه الدالة للتوقف عن العبارات التكرارية عند شرط معين والانتقال إلى ما بعد العبارة التكرارية لتنفيذ الخطوات البرمجية التي تليها.

مثال استخدامها في الحلقة for:

```
for( intialise counter; test counter; increment counter)
{
  if (condition)
  break;
  statement;
}
```

* اكتب برنامج يقوم بقياس درجة حرارة مريض بقارئ درجة واحدة في كل مرة وعند وصول درجة الحرارة 39 يكتب البرنامج اخراج رسالة تحذير وعند السلام ان المريض سيقيس كل درجة حرارة تبدأ من 36 الى 40 تحدث فقط.

```
#include<iostream.h>
void main()
{
for(int t=36; t<=40; t++)
{
  if(t==39)
  {
   cout<<"warning";
   break;
  }
cout<<"temperature is"<<t;
}
}
```

* عبارة continue :

تستخدم هذه الدالة لإيقاف الخطوات البرمجية التي تليها والانتقال إلى عبارة التكرار وعدة جديدة حسب شرط عبارة التكرار أي انه فقط يتجاهل statement وينتقل لعبارة التكرار ليبدأ عدة جديدة.

```
for(intialise couter ; test counter ; increment counter)
{
  if(condition)
  continue;
  statement;}
```

* اكتب برنامج لعبارة تكرارية تعد من 1 → 5 وعندما يصل العداد إلى 3 يتجاهل الخطوات البرمجية التي تليه (أي لا يطبع 3) ويعود لعبارة التكرار.

```
#include<iostream.h>
void main()
{
for(int i=1; i<=5; i++)
{
  if(i==3)
  continue;
  cout<<i<<"\t";}
}
```