الجامعة العربية الخاصة للعلوم والتكنولوجيا AUST

كلية الهندسة المعلوماتية Engineering Faculty

قسم تقانة المعلومات IT

- المقرر: مدخل إلى البرمجة (١)
- المحاضرة : الرابعة

Dr. Mohammed AL-Mohammed

Chapter 4

المؤثرات
Operators
في لغة c++

المؤثر هو عبارة عن رمز رياضي يقوم بإجراء عملية معينة على معاملاته والمعامل هو القيمة التي يجري عليها المؤثر العملية.

ويوجد في لغة C++ عدد من المؤثرات هي :

١- المؤثرات الحسابية Arithmetic Operators :

هناك ستة مؤثرات حسابية في لغة C++هي

المؤثر	معناه
+	الجمع Addition
-	الطرح Subtraction
*	الضرب Multiplication
/	القسمة Division
%	باقي القسمة Modulation
^	الرفع لقوى Exponential

أمثلة: الجدول التالي يوضح بعض العمليات الحسابية الجبرية وما يقابلها بلغة C++

جبرياً	بلغة C++
$m = \frac{a+b+c+d}{4}$	m=(a+b+c+d)/4
$m = a+b+c+\frac{d}{4}$	m=a+b+c+d/4
$m = ab+\frac{w}{x}-y$	m=a*b+w/x-y
$m = a+\frac{b+c}{8}+5d-\frac{a}{b}$	m=a+(b+c)/8+5*d-(a/b)

يجب ان نأخذ في الاعتبار في حالة القسمة (/) يكون الناتج عدداً صحيحاً إذا كانت عناصر البيانات المستعملة أعداد صحيحة وكذلك بالنسبة للمؤثر (%) يجب أن تكون عناصر البيانات قيم صحيحة وإلا فالنتيجة تكون خاطئة.

مثال : يوضح العلاقة بين عناصر البيانات من النوع الصحيح والمؤثرات الحسابية.

```
# include < iostream.h >
main( )
{
int a=18,b=5;
cout<<a+b<<"\n";
cout<<a–b<<"\n";
cout<<a*b<<"\n";
cout<<a/b<<"\n";
cout<<a%b<<"\n";
}
```

مثال : يمكن استخدام المؤثرات مع المتغيرات الصحيحة والحقيقة كما يلي :

```
# include < iostream.h >
main()
{
int a,b;
float x,y;
a=100;
b=5;
x=9–6;
y=0–3;
cout<<x+y<<"\n";
cout<<a–b<<"\n";
cout<<a*b<<"\n";
cout<<x/y<<"\n";
}
```

اولوية العمليات الحسابية

أي تعبير حسابي يجب ان ينفذ بالترتيب التالي :

١. الأقواس إن وجدت ( ) .
٢. الرفع للقوى ^ .
٣. الضرب (*) و القسمة (/) وباقي القسمة (%) من اليسار إلى اليمين .
٤. الجمع (+) والطرح (-) من اليسار إلى اليمين .

مثال : ما هو مخرج البرنامج التالي :

```
# include < iostream.h >
main()
{
int p,q,r,x;
p=2;
q=6;
r=p + q / p;
x=( p + q ) / p;
cout<<"r = "<<r<<"\n x = "<<x<<"\n";
}
```

الناتج

r = 5

x = 4

مثال : ما هي نتيجة العمليات الحسابية التالية :

1) x=7+3*6/2-1; x =15

2) y=2%2+2*2-2/2; y =3

3) Z=(3*9*(3+(9*3/(3)))); z =324

a) y=2*5/2+6-12/3

b) z=(5-6+8/2)+6*9/3-2

c) w=10-8%3*2+9

٢. المؤثرات العلائقية Relational Operators :

توجد ست مؤثرات علائقية نستطيع استخدامها على أي زوج من العناصر ويكون ناتجها إما صحيحاً True أو خاطئاً False .وهي كما يلي :

المؤثر	معناه	
= =	Equal to	يساوي
! =	Not Equal to	لا يساوي
>	Greater Than	أكبر من
<	Less Than	أصغر من
> =	Greater Than or Equal to	أكبر من أو يساوي
< =	Less Than or Equal to	أصغر من أو يساوي

مثال : افرض أن المتغيرين I وJ من النوع الصحيح ، وحدد لهما القيمتين 5 ، 3-على التوالي وفيما يلي مجموعة من التعبيرات والنتائج المناظرة لها .

التعبير	القيمة
I != J	TRUE
J * 2 = = I * 2	FALSE
I – J < I + J	FALSE
I – 3 = = J + 5	TRUE
J < I > I + J	TRUE

٣- المؤثرات المنطقية logical operators

وهي تستخدم لربط أكثر من تعبير منطقي بسيط مع بعضها البعض وهذه المؤثرات هي :

المؤثر	المؤثر في لغة C ++
AND"و"	&&
OR "أو"	||
NOT "لا"	!

وأولوية تنفيذها من اليسار إلى اليمين كما يلي

NOT ⟶ AND ⟶ OR

مثال: جدول الحقيقة والصواب للمؤثرات المنطقية

A	B	A&&B	A||B	!A
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

مثال: ما هو ناتج التعابير التالية :

١- ((5==5)&&(3>6))

True && false

False

٢-((5==5)||(3>6))

True || false

True

٣- F && F || F || F && F

F && F || F || F && F

F || F || F

F || F

F

المؤثر الشرطي (: ?) Conditional Operator

يستخدم لاختبار شرط معين ويرجع قيمة إذا تحقق الشرط ويرجع قيمة أخرى إذا لم يتحقق الشرط .

والصيغة العامة له هي :

(condition) ? result1 : result2

إذن المؤثر الشرطي يقبل ثلاث معاملات هي

- الشرط المنطقي (condition)
- القيمة الراجعة إذا تحقق الشرط result1
- القيمة الراجعة إذا لم يتحقق الشرط result2

مثال ١ : برنامج لإيجاد القيمة الأكبر من بين قيمتين باستخدام المؤثر الشرطي

```
#include <iostream.h>
int main ( )
{
int a,b,c;
a=2;
b=7;
c=(a>b)?a:b;
cout<<"max = "<<c;
return o;
}
```

مثال ٢ : عدل البرنامج السابق لإدخال ثلاثة قيم و إيجاد القيمة الأكبر

```
#include <iostream.h>
int main ( )
{
int a,b,c,d,e;
cin>>a>>b>>c;
d=(a>b)?a:b;
e=(c>d)?c:d;
cout<<"max = "<<e;
return o;
}
```

5- مؤثرات التعيين المركبة compound assignation operators

(+=, -=, *=, /=, %=)

يتوفر في لغة C ++ عدداً من عمليات الإسناد. فبالإضافة لعملية الإسناد العادية التي تستخدم فيها الإشارة = يوجد طرق مختصرة أخرى كما في الأمثلة التالية:

- عملية الإسناد b=b+c تكافئ عملية الإسناد المختصرة b+=c
- عملية الإسناد b=b-c تكافئ عملية الإسناد المختصرة b- =c
- عملية الإسناد b=b*c تكافئ عملية الإسناد المختصرة b*=c
- عملية الإسناد b=b/c تكافئ عملية الإسناد المختصرة b/=c
- عملية الإسناد b=b%c تكافئ عملية الإسناد المختصرة b%=c

وبطريقة أخرى يمكن كتابة عمليات الإسناد والتخصيص كما:

v+=r	تجمع r إلى v وتضع النتيجة في v
v-= r	تطرح r من v وتضع النتيجة في v
v*= r	تضرب r بـــ v وتضع النتيجة في v
v/= r	تقسم vعلى r وتضع النتيجة في v
v%= r	تجد باقي قسمة v على r وتضع النتيجة في v

مثال 1:

التعبير المكافئ	التعبير
Value=value + increase	value+=increase
a=a-5	a-=5
a=a/b	a/=b
price=price*(units + 1)	price*=units+1

مثال ٢

```
//compound assignation
#include <iostream.h>
int main ( )
{
int a,b=3;
a=b;
a+=2; //equivalent to a=a+2
cout<<a;
return 0;
}
```

مثال ٣:- ما هو ناتج البرنامج التالي :

```
#include <iostream.h>
int main ( )
{
int p,q,r;
p=2;
q=3;
r=10;
r/=p+q;
cout<<"r="<<r;
return 0;
}
```

٦-مؤثرات التزايد و التناقص Increase and Decrease Operators

مؤثر التزايد هو ++ ويعني زيادة قيمة المتغير بمقدار واحد

ومؤثر التناقص هو – ويعني إنقاص قيمة المتغير بمقدار واحد

مثال ١

العبارة C++

تعني زيادة قيمة المتغير C بمقدار واحد وهي مكافئة للتعابير التالية

```
C+=1
C=C+1
```

وينطبق هذا أيضا في حالة النقصان (--)

والتزايد والتناقص يمكن أن يكون قبلي أو بعدي, أي يمكن وضع++ أو-- في بداية المتغير أو في نهايته

ولكن هناك فرق كبير بينهما .

فالتعبير i ++ يعني إضافة واحد لقيمة المتغير i قبل احتساب التعبير

في حين أن التعبير i++ يعني استخدام القيمة الحالية للمتغير i ثم إضافة واحد إلى المتغير

أي أن عند استعمال ++i نزيد قيمة i قبل احتساب التخصيص وعند استعمال i++ يحسب التخصيص قبل زيادةi

وينطبق هذا أيضا في حالة النقصان --

وباختصار يمكن توضيح التزايد والتناقص القبلي والبعدي كما يلي

- ++x تزيد واحداً إلى x وبعدئذ تستعمل العبارة محتوى x الجديد.
- x++ يستعمل محتوى x أولاً وبعدئذ يزاد إليه واحد.
- --y تنقص واحداً من y وبعدئذ تستعمل العبارة محتوى y الجديد.
- y-- يستعمل أولاً محتوى y وبعدئذ ينقص منه واحد.

مثال ٢ **ما هي نتيجة كل من a ، b بعد كل عملية من العمليات الحسابية التالية :**

1. في البداية a = 6 b = 6

a =+ +b	a=7 b=7
a =b ++	a=7 b=8
++ b	a=7 b=9
a = - -b	a=8 b=8
a = b- -	a=8 b=7

2. إذا كان في البدء

$a = 2,\ b = 4,\ c = 6$

ما القيمة التي سيأخذها a, b,c بعد كل من العبارات التالية :

1. a + = 5 + a
2. a * = b+ + -c
3. a - = -b+ +*++c

الحل: 1-

a + =5+a = 5+2 =7

a = a +7= 2+7 = 9

القيم الجديدة: a=9, b=4, c=6

2 -

a* = b+ + -c = 4-6 = -2

a = a*-2 = 9 *-2 = - 18

القيم الجديدة: a=-18 , b = 5 , c = 6

3 -

a- = -b+ +*++c

a- = -5*7=-35

a = a-(-35) =-18 + 35= 17

القيم الجديدة: a =17, b =6, c=7

مثال:- ما هو ناتج البرنامج التالي :

```
#include <iostream.h>
int main ( )
{
int i,j,k;
i=j=2;
k=++i;
cout<<"i="<<i<<"k="<<k<<"\n";
k=j++;
cout<<"j="<<j<<"k="<<k<<"\n";
return 0;
}
i
```

جمل التحكم 1-:الجمل الشرطية

هي عبارة عن جمل تتحكم في كيفية تنفيذ خطوات البرنامج وتنقسم إلى نوعين:

١- جمل شرطية Conditional Statements

٢- جمل تكرار (حلقات تكرارية) Loops Statements

الجمل الشرطية: تتحكم في تنفيذ خطوات البرنامج من خلال شرط أو مجموعة شروط ومن هذه الجمل :

جملة IF والتي تأخذ ثلاثة أشكال

أ–

1– if (condition)
    statement;

مثال ١

```
if (x==100)
    cout<<"x is 100";
```

2– if(condition)

```
    {
      statement 1;
      statement 2;
      statement n;
    }
```

وهذا الشكل يعني أنه إذا تحقق الشرط فان جملة أو مجموعة من الجمل تنفذ

مثال ٢

```
if(x==100)
{
cout<<"x is ";
cout<<x;
}
```

ب-

```
if(condition)
statement1;
else
statement2;
```

وهذا الشكل يعني انه إذا تحقق الشرط فان جملة أو مجموعة من الجمل تنفذ وإذا لم يتحقق الشرط فان جملة
أو مجموعة من الجمل تنفذ أيضاً

مثال 3

```
    if(x==100)
    cout<<"x is 100";
else
    cout <<"x is not 100";
```

ج-

```
if  (condition1)
   statement1;
else if(condition2)
   statement2;
else if(condition3)
   stataement3;
    :
     else
   statement n;
```

```
if(x>0)
  cout<<"x is positive";
else if (x<0)
  cout<<"x is negative";
else
  cout<<"x is 0";
```

أمثلة على جملة if:-

١. اكتب برنامج يقوم بإدخال القيمة الصحيحة xوإذا كانت اكبر من أو تساوي ١٠٠ اطبع large value

```
#include<iostream.h>
main( )
{
int x;
cout<<"enter x:";
cin>>x;
if ( x>=100)
cout<<"large value";
}
```

٢. اكتب برنامج لإدخال القيمتين الصحيحتين xو y ثم طباعة القيمة الأكبر

```
#include<iostream.h>
 main( )
{
 int x,y;
cout<<"enter x and y \n";
cin>>x>>y;
if (x>y)
cout<<"max ="<<x;
else
cout<<"max="<<y;
}
```

٣. اكتب برنامج لإدخال ثلاث قيم صحيحة ثم طباعة القيمة الأكبر

الطريقة الأولى :

```
#include<iostream.h>
main( )
 {
 int x,y,z;
cout<<"enter x,y,z";
cin>>x>>y>>z;
if(x>y && x>z)
cout<<"max="<<x;
else if (y>x && y>z)
cout<<"max="<<y;
else
cout<<"max="<<z;
}
```

الطريقة الثانية :

```
#include<iostream.h>
main()
{
int x,y,z,m;
cout<<"enter x,y,z";
cin>>x>>y>>z;
m=x;
if(y>m)
m=y;
 if (z>m)
m=z;
cout<<"max="<<m;
}
```

٤. اكتب برنامج لإدخال عدد ثم تحديد فيما إذا كان العدد فردي أم زوجي

```
#include<iostream.h>
main()
{
int number;
cout<<"enter number :";
cin>>number;
if (number % 2 ==0)
cout<<"the"<<number<<"is even number ";
else
cout<<"the"<<number<<"is odd number";
}
```

٥. أكتب برنامج لإدخال عددين صحيحين وفحص فيما إذا كان الثاني قاسم للأول ام لا .

```
# include<iostream.h>
main ( )
{
      int  a , b ;
      cout<<"enter a=";
      cin>>a;
      cout<<"enter b=“;
      cin>>b;
      if ( (b!= 0) && (a % b == 0) )
          cout << a << " is divisible by " <<b ;
      else
          cout <<a<<" is not divisible by " << b ;
      }
```

جملة Switch

تستخدم هذه الجملة الشرط المغلق close condition أي الشرط الذي يستخدم قيمة محددة (رقمية أو حرفية)

ولهذا يمكن القول بأن هذه الجملة تتعامل مع متغير واحد يمكن ان يأخذ في كل مرة قيمة محددة من خلال مجموعة من القيم

الصيغة العامة للجملة :

```
switch(variable)
{
case value 1: statement(s);break;
case value 2: statement(s);break;
:
:
case value n: statement(s);break;
default :statement(s);
```

هذه الصيغة تمثل الاحتمالات التي يمكن أن يأخذها المتغير variable من [1-----n]

كل احتمال يتم وضعه داخل جملة باستخدام العبارة case ثم بعد ذلك كتابة الحدث الذي يمكن أن يكون تعليمة أو مجموعة تعليمات ولا بد أن ينتهي الحدث بواسطة العبارة break التي تمثل نهاية الجملة أما الاحتمال default يمثل الحدث الملازم للحالة خارج النطاق من [1-------n]

مثال ١ :-

اكتب برنامج لإظهار القائمة التالية :

Main Menu
1-college
2-Department
3-address
Enter your choice[1----3]

وعلى المستخدم اختيار واحدا من الخيارات الثلاثة بإدخاله رقم الخيار المطلوب

حيث

(١) يطبع Teachers' college (٢)يطبع computer (٣) يطبع Riyadh

```
#include<iostream.h>
main()
{
int x;
cout<<"\t\t\t Main Menu";
cout<<"\n\t\t 1-college \n\t\t 2-Deparment ";
cout<<"\n\t\t 3-address ";
cout<<"\n\t\t Enter your choice [1-3]\n";
cin>>x;
switch(x)
{
case 1:cout<<"Teachers' college  "; break;
case 2:cout<<"Computer";  break;
case 3:cout<<"Riyadh"; break;
default:cout<<"out of range";
}
}
```

مثال٢ :أكتب برنامج لإعطاء اسم اليوم من أيام الأسبوع عند ادخال رقمه

```
 # include < iostream.h>
main ()
{
    int  d ;
    cout << "enter number : " ;
    cin >> d ;
    switch (d )
{
    case 1 : cout << " Saturday " ; break ;
    case 2 : cout << " Sunday " ; break ;
    case 3 : cout << " Monday " ; break ;
    case 4 :  cout << " Tuesday " ; break ;
    case 5 :  cout << " Wednesday " ; break ;
    case 6 :  cout << " Thursday " ; break ;
    case 7 :  cout << " Friday " ; break ;
    default :  cout << " The number out of range(1----7) " ;
}
}
```

مثال٣:اكتب برنامج لإدخال القيمتين X وy ثم اجراء العمليات الحسابية (الجمع،الطرح،الضرب،القسمة،باقي القسمة)على هاتين القيمتين.

جمل التحكم

الجزء الثاني :الحلقات التكرارية

ب- الحلقات التكرارية (جمل التكرار) Itreation Statements

تستخدم الحلقة التكرارية في تكرار تنفيذ تعليمة أو مجموعة من تعليمات

أكثر من مرة وهي نوعان :

البرنامج في هذا النوع يتم تحديد عدد مرات التكرار بواسطة رقم صحيح ويبين هذا النوع على مفهوم العداد counter وهو عبارة عن متغير يأخذ قيمة ابتدائيةInitial Value تتغير باستمرار بمعدل معين بالزيادة أو النقصان الى ان تصل الى القيمة النهائية Final value ولتصميم هذه البنية نستخدم الجملة for

الصيغة العامة لحلقة for

```
for(counter=initial_value ; condition ; step)
```

حيث:

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

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

الشكل العام للحلقة

```
for(       )
   statement;
```

أو

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

امثلة على حلقة for

١. اكتب برنامج لطباعة الارقام من ١ الى ١٠ ومربعاتها

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

٢. اكتب برنامج لحساب متوسط القيم من 123_____477

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

٣. اكتب برنامج لحساب مضروب n

حيث ni=1x2x3x4x……xn

أو n!=nx(n–1)x(n–2)x….x2x1

أ– الطريقة الاولى :

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

ب- الطريقة الثانية

```
#include<iostream.h>
main()
{
int n , f ,i ;
cout<<"Enter n:" ;
cin>>n ;
f=1;
for(i=n ; i>=1 ; i--)
f *=I ;
cout<<"factorial ="<<f ;
}
```

٤- أكتب برنامج لحساب ab (باستخدام عملية *)

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

٥-أكتب برنامج لإيجاد جمع المتسلسلة التالية :

حيث n يتم إدخالها من قبل المستخدم

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

٦. برنامج إيجاد قواسم عدد x

الحل :نختبر الأعداد التي قبل x بحيث إذا كان باقي القسمة عليها يساوي الصفر عندئذ يكون العدد قاسما للعدد x فإذا فرضنا أن العدد x=6 فان القواسم هي ٢ و٣

```
# include < iostream . h >
main ()
{
      int  x  ;
      cout << " enter number : " ;
      cin >> x ;
      for ( int i = 1 ; i <x ; i ++)
      if ( x % i = = 0 )
           cout << i << " \n";
}
```

٧. أكتب برنامج لطباعة :

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

باستخدام الحلقات المتداخلة

```
#include<iostream.h>
main()
{
int i , j ;
for (i=1 ; i<=5 ; i++){
for ( j=1 ; j<=i ; j++){
cout<<"*" ;
}
cout<<"\n";}}
```

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 وتستخدم الجملة while() ويمكن أن تؤدي دور الجملة for().

الصيغة العامة لجملة while

```
while (condition)
{
statement 1 ;
statement 2 ;

statement n ;
}
```

مثاّل : حوُل 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

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

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

٢- أكتب برنامج لطباعة الأعداد من n------ 1 بشكل تنازلي :

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

٣- أكتب برنامج لإيجاد متوسط n من القيم الصحيحة المدخلة من قبل المستخدم

```
#include<iostream.h>
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";
}
```

٢- حلقة : do-while

هذه الجملة مشابهة لجملة while وتختلف عنها في أمرين :

تبدأ بالعملية أولا ويتم التحقق من الشرط في أسفل الجملة.

لابد 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 :**

١. اكتب برنامج يقوم بإيجاد مربعات الأعداد من ١ إلى ١٠ ؟

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

٢. أكتب برنامج لحساب حاصل ضرب جميع الأعداد الزوجية من ٢ إلى ١٠ ؟

```
# include < iostream.h >
main( )
{
int i, product;
product=1;
i=2;
do
{
product * = i;
i + = 2;
} while (i<=10);
cout<<" product = " << product;
}
```

٣. اعد 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);
```

تدريب

١. باستخدام جملة التكرار while اكتب برنامج لإيجاد حاصل جمع الآتي :-

a- sum 1 =1+2+3+4+…n

b- sum 2 =1-2+3-4+…n

٢. اكتب برنامج لإيجاد حاصل جمع مربعات الأعداد الصحيحة الفردية الواقعة بين عددين صحيحين يقوم بإدخالهما المستخدم عن طريق لوحة المفاتيح ؟