Example3: ***Write C++ program to calculate the squared value of a number passed from main function. Use this function in a program to calculate the squares of numbers from 1 to 10:***

```
include<iostream.h>
int square ( int y )
{
        int z;
        z = y * y; return ( z
        );
}
void main( )
{
         int x;
         for ( x=1; x <= 10; x++ )
            cout << square ( x ) << endl;
}
```

Example4: ***Write C++ program using function to calculate the average of two numbers entered by the user in the main program:***

```
include<iostream.h>
float aver (int x1, int x2)
{
       float z;
       z = ( x1 + x2) / 2.0; return
       ( z);
}
void main( )
{
       float x;
       int num1,num2;
       cout << "Enter 2 positive number \n";
       cin >> num1 >> num2;
       x = aver (num1, num2); cout
       << x;
```

}

Example5: ***Write C++ program, using function, to find the summation of the following series:***

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2$$

```
#include<iostream.h> int
summation ( int x)
{
        int i = 1, sum = 0;
        while ( i <= x )
        {
                sum += i * i ; i++;
        }
        return (sum);
}
void main ( )
{
         int n ,s;
         cout << "enter positive number"; cin
         >> n;
         s = summation ( n );
         cout << "sum is: " << s << endl;
}
```

Example6:***Write a function to find the largest integer among three integers entered by the user in the main function.***

```
#include <iostream.h>
```

```
int max(int y1, int y2, int y3)
{
        int big; big=y1;
        if (y2>big) big=y2; if
        (y3>big) big=y3; return
        (big);
}
void main( )
{
        int largest,x1,x2,x3;
        cout<<"Enter 3 integer numbers:";
        cin>>x1>>x2>>x3;
        largest=max(x1,x2,x3); cout<<largest;
}
```

Example7: ***Write program in C++, using function, presentation for logic gates (AND, OR ,NAND, X-OR,NOT) by in A,B enter from user.***

```
#include<iostream.h>
void ANDF(int,int);
 void ORF(int,int);
 void XORF(int,int);
void NOTF(int);

void main()
{ char s;int a,b;
cout<<"Enter the value A,B :";
cin>>a>>b;
  cout<<"Enter the select value \n";
  cout<<"\ta--(AND gate)\n\to--(OR gate)\n\tx--(X-OR)\n\tn--(NOT gate)\n\te--(<<EXIT>>
      :";
      cin>>s;
      switch(s)
              {
              case 'a':ANDF(a,b);break;
               case 'o':ORF(a,b);break;
              case 'x':XORF(a,b);break;
              case 'n':NOTF(a); cout<<" ";NOTF(b);break;
              case 'e':break;
              deafult:cout<<":bad choose";
```

```
}

}
void ANDF(int a,int b)
 {
 cout<<(a&&b);
 }
 void ORF(int a,int b)
 {
 cout<<(a||b);
 }
 void XORF(int a,int b)
 {
 cout<<(a^b);
 }
 void NOTF(int a)
 {
 cout<<(!a);
 }
```

passing array to function

EX: Write a program to enter integer array named A[6] in main program and then use the function to find the sum of it and print the sum in main program

.

```
#include<iostream.h>
  int sum (int a[6])
 { int i,s;
 s=0;
     for(i=0;i<6; i++)
       s=s+a[i];
    return s;
}
main()
{ int i,k ,a[6];
 for(i=0 ; i<6 ; i++)
 {cout<<" enter the number"<< "\n";
 cin>>a[i];
```

الدالة والفرعية

بعد انتهاء تنفيذ s ستقوم باخراج قيمة ال returnنلاحظ ان جملة ال s الدالة زالفرعية وعودتها الى كلمة الاستدعاء وتقوم بوضع هذه القيمة في اسم الدالة في كلمة الاستدعاء في الدالة الرئيسية ثم تنقل الى المتغير k ثم نكمل تنفيذ الجمل التي تليها في الدالة الرئيسية

```
}
k=sum(a)
cout<<"the sum ="<<k<<"\n";
return 0;
}
```

كلمة استدعاء الدالة الفرعية

EX: write a program that use function to read integer array B[5] and find the sum of only even numbers and print the sum in main program .
: write a program that use function to read integer array B[5] and find the sum of only even numbers and print the sum in main program .

```
#include < iostream.h>
 int even( )
 {
int i, B[5] ;
  i=0;
while(i<=4)
{
cin>>B{i];
 if(B[i] %2= =0 )
s=s+B[i] ;
i=i+1;
 }
 return  s;
 }
main()
{int k;
   k=even( ) ;
 cout<<" the sum= " << k ;
return 0; }
```

EX:Write a program to enter the number and then test it if the number primary or not by using a function and print the message show if the number is primary or not in it .

```
 #include<iostream.h>
int primaryornot(int  x)
int i, k;    {
    k=1;
for(i=2 ;i<=x-1  ; i++)
  if (x % i = = 0 )
    {k=0;
```

جملة واحدة داخل ال forجملة ال if

```
    break;
    }
if(k==0)
    cout<< x<< "is not primary "<<"\n" ;
else
  cout<< x <<" is  primary "<<"\n" ;
return 0;
}

main ()
 { int d ;
  cin>> d ;
primaryornot(d) ;
return 0;
}
```

فكرة

الاعداد الاولية ان العدد الاولي هو العدد الذي يقبل القسمة على واحد وعلى نفسة فقط و بدون باقي بينما باقي الارقام ايضا تقبل القسمة على نفسها وعلى واحد وبدون باقي وايضا تقبل القسمة على بعض الارقام وبدون باقي فتسمى هذه الاعداد بالاعداد الغير اولية فلكي نعرف ان العدد اولي او لا يكون كالاتتي انه عندما ندخل رقم فاننا نعد من واحد الى هذا الرقم المدخل فعلى سبيل المثال عندما ندخل الرقم 5 فنعد من 1,2,3,4,5واحد الى خمسة اي ( ) ثم نترك الواحد والخمسة لان كل الارقام تقبل القسمة على واحد ووعلى 2,3,4وبدون باقي فيترك الواحد والخمسة في مثالنا فقط ناخذ (نفسها)ونقوم بقسمة العدد المدخل على 2,3,4كل منهم0 فقيمة العدد المدخل والذي هو في مثالنا العدد خمسة فاذا واحد من هذه الاعداد( )عندما 2,3,4نقسم العدد خمسة عليةيكون بدون باقي فان العدد خمسة هو عدد غير اولي اما اذا كل الاعداد ( ) 2,3,4عندما نقسم الخمسة عليهم يكون هناك باقي فان العدد هو اولي وهكذا بالنسبة الى بقية الارقام عندما نريد ان نعرف هل هي اولية او لا 0

EX: Write a program to enterr integer array a[3][3] in main program and then use function to find the square of element in main diagonal and then print the result array in it .

```
#include< iostream .h>
#include< math .h>
  int diagonal (int a[3][3])
 { int i , j ;
for(i=0:i<3 ; i++)
 for(j=0 ; j<3 ; j++)
 if(i==j)
 a[i][j]= pow(a[i][j], 2)
    for(i=0 ; i<3 ; i++ )
 {
for(j=0 ; j<3 ; j++)
```

```
 cout<< a[i][j] ;
  cout <<"\n" ;
 }
      return 0;
     }
     main()
      { int  i,j ,a[3][3] ;
for(i=0 ; i<3 ; i++)
          for(j=0 ; j<3 ; j++)
           cin>>  a[i][j] ;
     diagonal(a);
    return 0;
      }
```

ex:Writ e a program to enter integer array a[5] in main program and then use two function.The first function use to sort array ascending and print the result array in it . The second function use to sort the array descending and print the result array in it .

```
#include<iostream.h>
 int sortasc( int a[5] )
  { int i ,j,  t;
   fot(i=0 ; i<=3  ; i++)
  forj(j=i+1; j<=4 ;j++)
    if (a[i]>a[j] )
        {  t= a[i] ;
          a[i]= a[j] ;
          a[j] = t ;
        }
 for(i=0 ; i<=4 ; i++)
  cout<< a[i]<<"\n" ;
return 0;
}
 int sortdes( int a[5] )
  { int i ,j,  t;
   fot(i=0 ; i<=3  ; i++)
  forj(j=i+1; j<=4 ;j++)
    if (a[i]< a[j] )
        {  t= a[i] ;
          a[i]= a[j] ;
```

الدالة الاولى

الدالة الثانية

ملاحظة: لايمكن ان تكون ال value بعد ال كلمة ال return عبارة عن مصفوفة لان تكون قيمة واحدة والمصفوفة تحتو ي على عدة قيم مخزونة في مواقع return بعد ال لاخراج المصفوفة بعد تغييرقيمها من الدالة الفرعية الى return لذلك لا تستخدم ال كلمة استدعائها لان المصفوفة بمجرد تغيير قيمها مثلا في الدالة الفرعية والعودة الى كلمة استدعائها فان القيم القديمة تلغى ونحتفظ بالقيم الجديدة لذلك الدالة الرئيسية ستتعرف على القيم الجديده للمصفوفة بعد الخروج من الدالة الفرعية الى كلمة استدعائها في الدالة الرئيسية لتكملة تنفيذ الجمل وهذا فقط يحدث مع المصفوفات(اي ان لاخراج المصفوفة الناتجة فيها لانها ستكون معروفة return الدالة لا تستخدم ال مباشرة في الدالة الرئيسية)

```
        a[j] = t ;
      }
 for(i=0 ; i<=4 ; i++)
  cout<< a[i]<<"\n" ;
return 0;
}
main()
  { int a[5]
   for(i=0; i<=4 ; i++)
    cin >> a[i];
  sortasc( a) ;
  sortdes(a);
return0;
}
```

كلمة استدعاء الدالة الاولى

ان كلمة الاسدعاء sortasc(a) تقوم باستدعاء الدالة الاولى وسوف نرسل لها المصفوفة a الغير المرتبة والتي ادخلناها في الدالة الرئيسية حيث تقوم الدالة الاولى بيرتيب المصفوفة المرسلة لها ترتيب تصاعديا وعندما تنتهي تنفيذ الدالة الاولى ترجع الى كلمة استدعائها في الدالة الرئيسية لتكمل تنفيذ الجمل التي تلي كلمة الاستدعاء وبعد عودتها الى كلمة الاستدعاءنكون قد حصلنا على مصفوفة مرتبة تصاعديا والمصفوفة القديمة الغيت اي هنا تغيرت المصفوفة من مصفوفة غير مرتبة الى مصفوفة مرتبة تصاعيا فان الدالة الرئيسية ايضا سوف تتعامل مع هذه المصفوفة المرتبة لان المصفوفة القديمة تغيرت واصبحت مرتية تصاعديا لذلك الدالة الرئيسية دائما تاخذ التغير الجديد الذي يطرا على المصفوفة ثم بعد الانتقال الى تنفيذ الجملة التي تلي كلمة استدعاء الدالة الاولى والتي هي ايضا كلمة استدعاء ولكن للدالة الثانية عن طريق كلمة الاستدعاء sortdes(a) فهنا سوف نرسل لهذه الدالة المصفوفة a المرتبة تصاعديا وليست المصفوفة الاولى التي ادخلناها لان المصفوفة الاولى تغيرت بعد اخالها الى الدالة الاولى واصبحت مرتية تصاعديا فبعد رجوعها الى الدالة الرئيسية ناخذ التغير الجديد الذي طرا على المصفوفة ثم تقوم الدالة الثانية باعادة ترتيب المصفوفة التي ادخلت لها ترتيب تنازليا اي ان التفيير النهائي الذي سيحدث على المصفوفة انها ستكون مرتبة تنازليا لان بعد انتهاء تنفيذ الدالة الثانية وعودتها الى كلمة الاستدعاء في الدالة الرئيسية سنكون قد حصلنا على مصفوفة مرتبة تنازليا

4-write a program to enter 10 integer numbers and find the sum of positive numbers only in main program. Ues the function to test if the number is positive or not .

```
#include<iostream.h>
 int  positive(int x )
{ int  k;
if(x> 0)
   k=1;
    else
k=0;
 return k;
}
main ()
{int s,f,x,I;
s=0;
for(i=0 ; i<=9; i++)
{ cin>> x;
  f=positive(x)
if( f= =1)
s=s+x;
}
cout<<"the sum "<<s;
return 0; }
```

EX//Write a program to enter A[5] in main program and use a function to square the element in the array and print the result array in main program

```
#include<iostream.h>
#include<math.h>
Int sqrarray(int a[5])
{ int i;
  For(i=0  ;i<5  ;i++)
   a[i] = squre(a[i],2) ;
  Return 0 ;
}
Main()  int
{ int  i;
 For(i=0; i<5 ;i++)
  Cin >>a[i];
sqrarray(a);
For(i=0 ;i<5 ;i++)
 Cout<<a[i] ;
Return 0;
}
```

عندما نقوم الدالة الرئيسية باستدعاء الدالة الفرعية عن طريق كلمة سوف يتم ارسال المصفوفة التي تم ادخالها في sqrarray(a)الاستدعاء الدالة الرئيسية الى الدالة الفرعية بعد ذلك يتم تنفيذ الدالة الفرعية حيث تقوم هذه الدالة بتربيع قيم المصفوفة المدخلة اي اصبحت لدينا مصفوفة بقيم جديدة وهي المربعة فعند انتهاء تنفيذ الدالة الفرعية ورجوعها الى كلمة استدعائها فيى الدالة الرئيسية لتكمل تنفيذ الجمل التي تليها فان الدالة الرئيسية ستتعرف على المصفوفة الجديدة المربعة لذلك تقوم الدالة الرئيسية بطباعة المصفوفة المربعة ونلاحظ ان الدالة الرئيسية تعرفت كما ذكرنا ذلك سابقاreturnعلى المصفوفة الجديدة بدون اخراجها بال