

نظم معلومات ادارية المرحلة الثانية

ملاحظات مهمة

ملاحظة://

من الممكن استخدام اكثر من عبارة اعادة في الدالة الواحدة ولكن واحدة منها سوف تنفذ وتنتهي البرنامج والاخرى او الأخرى سوف تهمل مثال :

```
int max (int x ,int y)
{
    if (x > y)
        return x ;
    else
        return y ;
}
```

ملاحظة://

```
return (x > 5);
```

ستعيد صح او خطأ وليس قيمة المتغير (x) ، فاذا كانت العبارة صح ستعيد القيمة (1) اما اذا كانت خطأ ستعيد القيمة (0) .

ملاحظة://

إذا ما تم تنفيذ عبارة الأعادة (return)، فإنها ستكون اخر عبارة تنفذ في تلك الدالة ولا تنفذ اي عبارة بعدها وبذلك ينتهي تنفيذ الدالة.

ملاحظة://

اعادة قيمة في الدالة ممكن ان تتم باستخدام عبارة الاعادة كما بينا، او من الممكن ان تستخدم طريقة الاسناد وذلك باسناد قيمة الى اسم الدالة داخل جسم الدالة، في ادناه امثلة لاعادة قيم :

```
int square ( int x ،int y )  
    { int s = x * y ;  
      return s;    }
```

```
int square ( int x ،int y )  
    { return ( x * y ) }
```

جملة (return) لها وظيفتان:

1. تعد مخرجا طبيعيا في نهاية الدالة، وتعيد نتيجة الدالة الى العبارة التي استدعت الدالة في البرنامج.
2. تستعمل لعمليات حساب واستخراج قيم تعابير بداخلها.

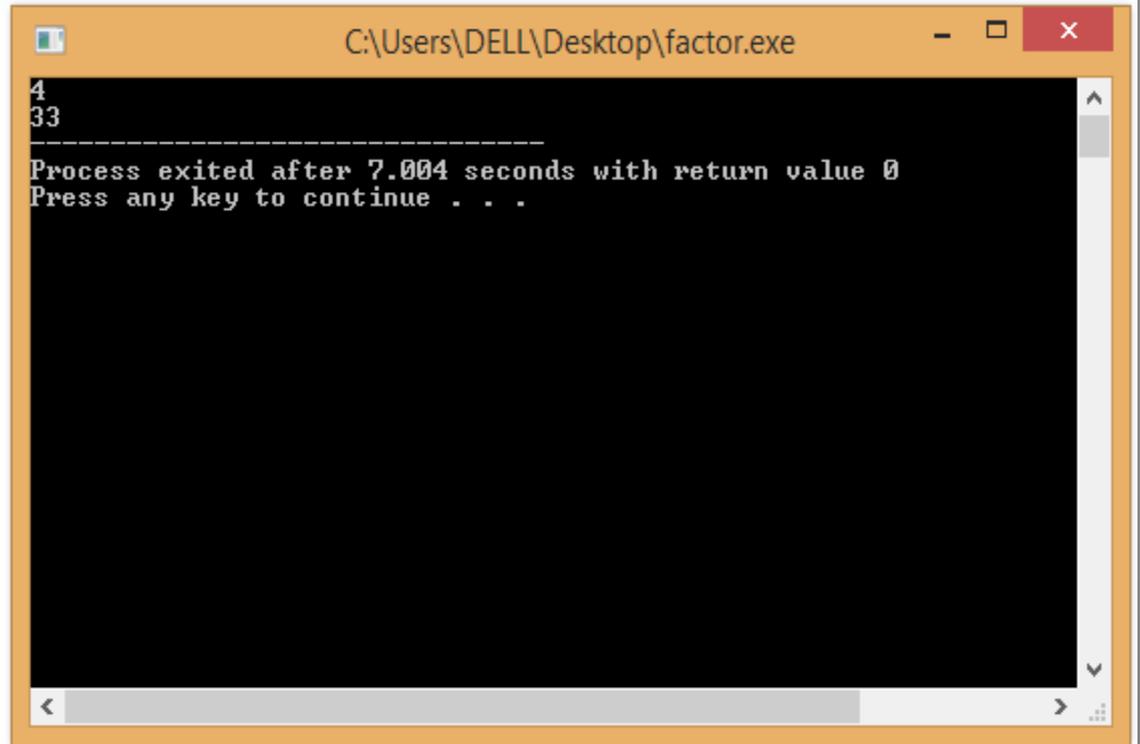
اكتب برنامج لحساب مجموع n من المتوالية التالية:

$$F=1!+2!+3!+4!+\dots+n!$$

استخدم دالة تحسب المفكوك.

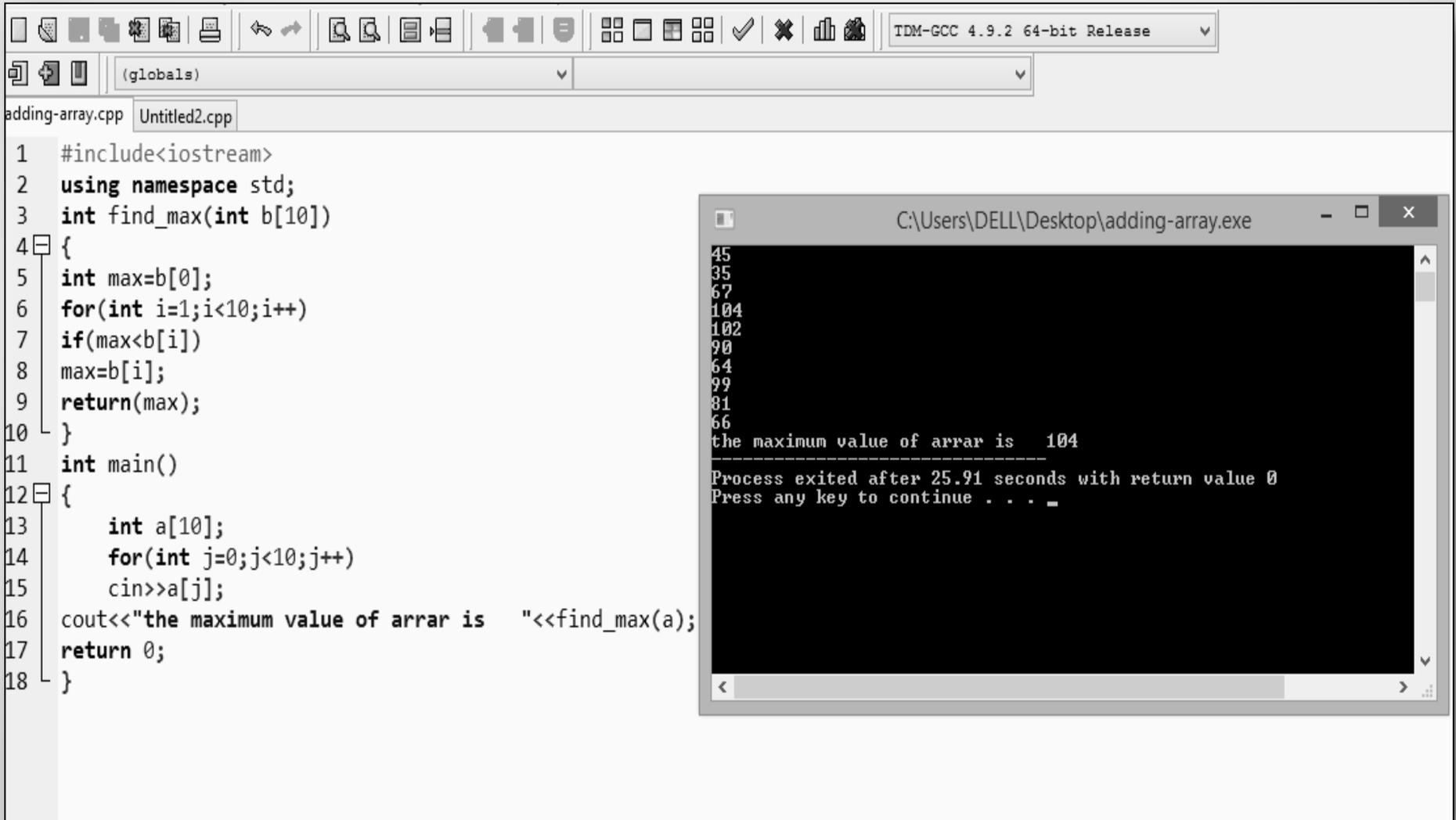
actor.cpp

```
1  #include<iostream>
2  using namespace std;
3  int fac(int a)
4  {
5      int fact=1;
6      for(int i=1;i<=a;i++)
7          fact*=i;
8      return(fact);
9  }
10 int main()
11 {
12     int f=0,n;
13     cin>>n;
14     for(int j=1;j<=n;j++)
15         f=f+fac(j);
16     cout<<f;
17     return 0;
18 }
19
20
```



```
C:\Users\DELL\Desktop\factor.exe
4
33
-----
Process exited after 7.004 seconds with return value 0
Press any key to continue . . .
```

اكتب برنامج لقراءة مصفوفة احادية البعد ثم استخدم دالة لايجاد اكبر قيمة في المصفوفة



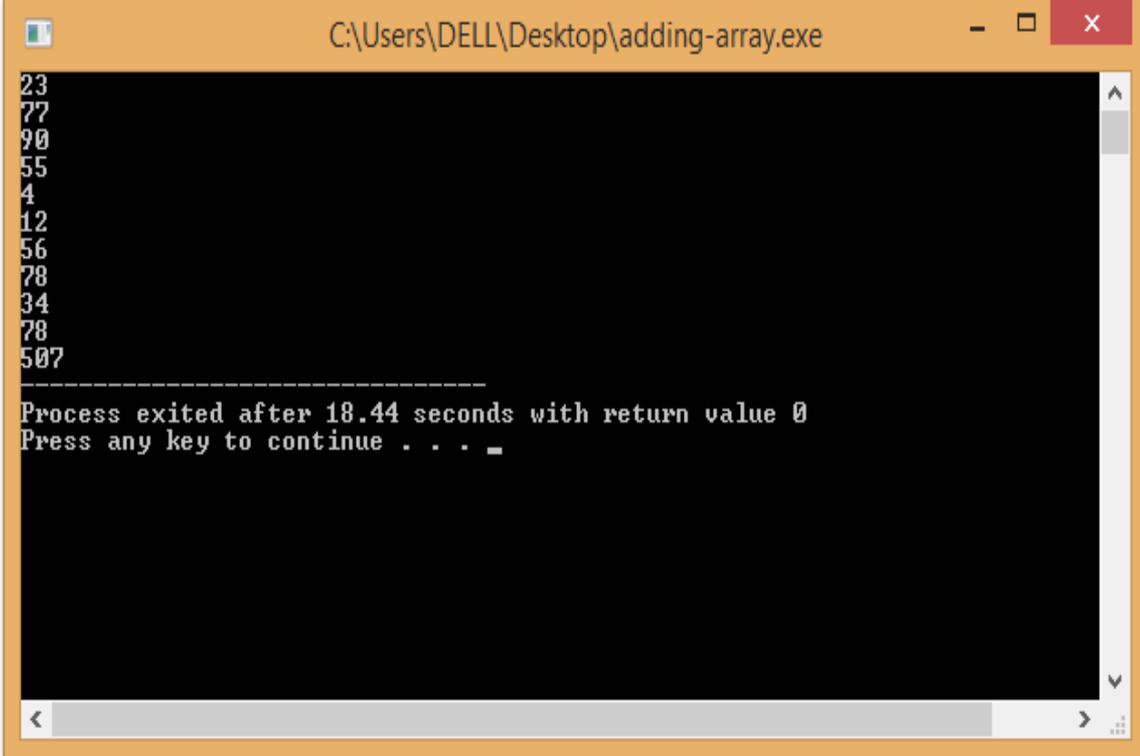
```
1 #include<iostream>
2 using namespace std;
3 int find_max(int b[10])
4 {
5     int max=b[0];
6     for(int i=1;i<10;i++)
7         if(max<b[i])
8             max=b[i];
9     return(max);
10 }
11 int main()
12 {
13     int a[10];
14     for(int j=0;j<10;j++)
15         cin>>a[j];
16     cout<<"the maximum value of arrar is  "<<find_max(a);
17     return 0;
18 }
```

C:\Users\DELL\Desktop\adding-array.exe

```
45
35
67
104
102
90
64
99
81
66
the maximum value of arrar is 104
-----
Process exited after 25.91 seconds with return value 0
Press any key to continue . . . _
```

اكتب برنامج يقرأ مصفوفة احادية ذات ١٠ قيم عددية ثم يحسب مجموع قيم المصفوفة باستخدام دالة

```
(globals)
vector.cpp adding-array.cpp
1 #include<iostream>
2 using namespace std;
3 int adding(int b[10])
4 {
5     int sum=0;
6     for(int i=0;i<10;i++)
7     sum+=b[i];
8     return(sum);
9 }
0 int main()
1 {
2     int a[10];
3     for(int j=0;j<10;j++)
4     cin>>a[j];
5     cout<<adding(a);
6     return 0;
7 }
```



```
C:\Users\DELL\Desktop\adding-array.exe
23
77
90
55
4
12
56
78
34
78
-----
Process exited after 18.44 seconds with return value 0
Press any key to continue . . . _
```

اكتب برنامج لقراءة ثلاث مصفوفات احادية البعد ثم يجد مجموع كل مصفوفة ،استخدم دالة لايجاد المجموع

```
tor.cpp adding-array.cpp
1  #include<iostream>
2  using namespace std;
3  int adding(int x[5])
4  {
5  int sum=0;
6  for(int i=1;i<=5;i++)
7  sum+=x[i];
8  return(sum);
9  }
10 int main()
11 {
12 int a[5],b[5],c[5];
13 for(int j=0;j<5;j++)
14 cin>>a[j];
15 for(int j=0;j<5;j++)
16 cin>>b[j];
17 for(int j=0;j<5;j++)
18 cin>>c[j];
19 cout<<"sum of a  "<<adding(a)<<endl;
20 cout<<"sum of b  "<<adding(b)<<endl;
21 cout<<"sum of c  "<<adding(c)<<endl;
22 return 0;
23 }
```

```
C:\Users\DELL\Desktop\adding-array.exe
20 30 40 50 60
40 60 80 100 110
120 140 160 180 200
sum of a 185
sum of b 349
sum of c 680

-----
Process exited after 35.5 seconds with return value 0
Press any key to continue . . .
```

تمرين: اكتب برنامج لقراءة ثلاث مصفوفات احادية البعد ثم ايجاد اكبر قيمة في كل مصفوفة
استخدم دالة لايجاد القيمة الاكبر .

Void-Function

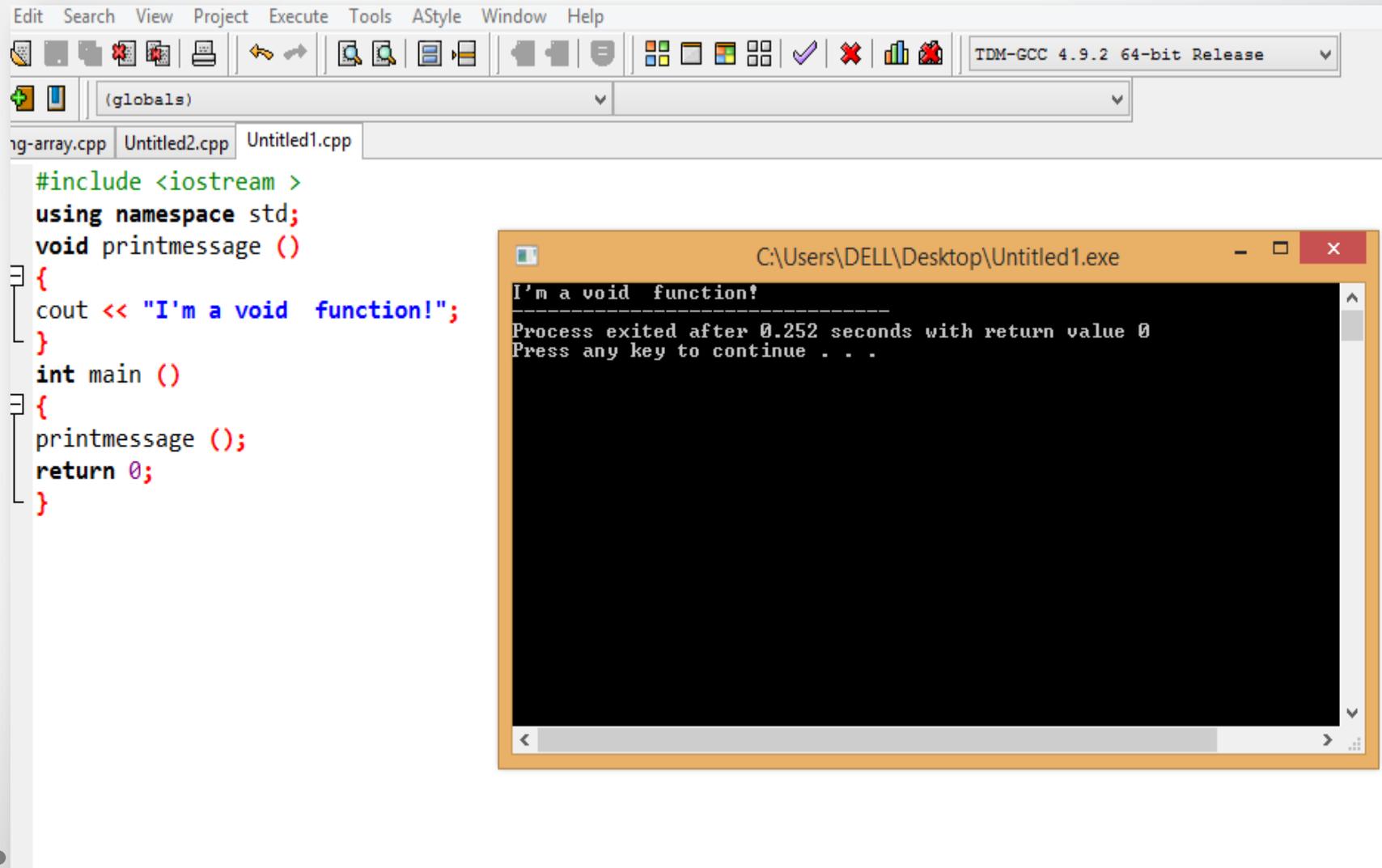
- Function definition syntax:

```
void functionName()  
{  
    statements  
}
```

هذا النوع من دالة void بدون :
- معلمات
- نوع بياني يسبق الدالة
- جملة return
- اي انها لاتاخذ معلمات وليس لها مخرجات

- void is a reserved word
- Function call syntax: `functionName();`

مثال:



The image shows a C++ IDE window with the following code in the editor:

```
#include <iostream >
using namespace std;
void printmessage ()
{
    cout << "I'm a void function!";
}
int main ()
{
    printmessage ();
    return 0;
}
```

The IDE toolbar includes menus for Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The compiler is identified as TDM-GCC 4.9.2 64-bit Release. The file tabs show 1g-array.cpp, Untitled2.cpp, and Untitled1.cpp.

A terminal window titled "C:\Users\DELL\Desktop\Untitled1.exe" displays the output of the program:

```
I'm a void function!
-----
Process exited after 0.252 seconds with return value 0
Press any key to continue . . .
```

void Functions With Parameters

- Function definition syntax:

void functionName(formal parameter list)

{

statements

}

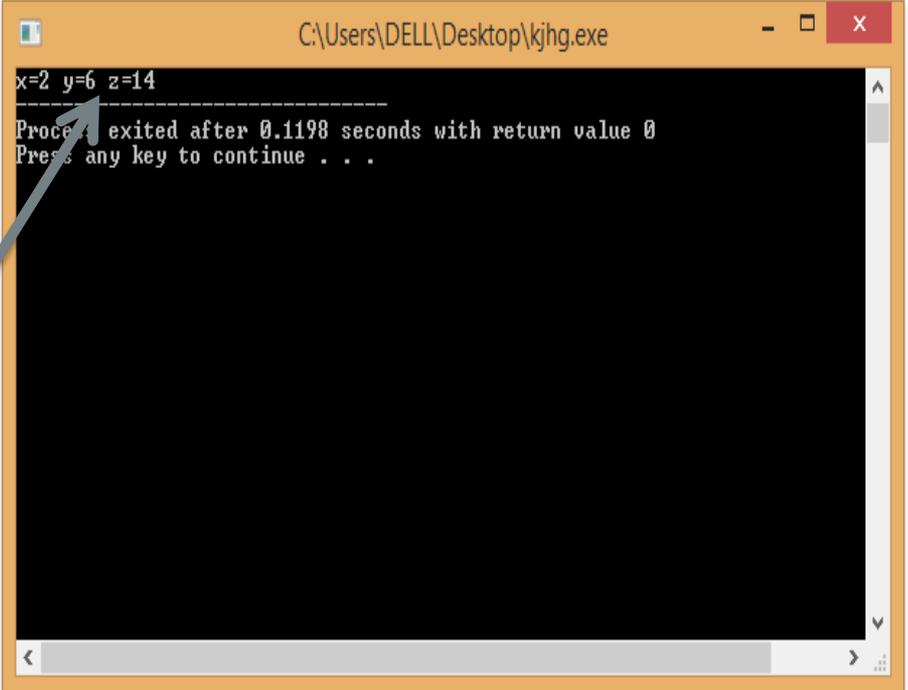


هذا النوع باستخدام المعلمات
وفائدة هذا النوع :

- يسمح باعادة اكثر من قيمة للبرنامج الرئيسي
- استبدال المعلمات الحقيقية
- عندما يتم تمرير العناوين للمعلمات فانه سيتم توفى الخزن والوقت

لارجاع اكثر من قيمة بواسطة الدالة نستخدم طريقة الدالة void:

```
adding-array.cpp  Untitled2.cpp  Untitled1.cpp  Untitled2.cpp  kjhg.cpp
1  #include <iostream>
2  using namespace std;
3  void duplicate (int &a, int &b, int &c)
4  {
5  a*=2;
6  b*=2;
7  c*=2;
8  }
9  int main ()
10 {
11 int x=1, y=3 ,z=7;
12 duplicate (x, y, z);
13 cout << "x=" << x << " y=" << y << " z=" << z;
14 return 0;
15 }
16
```



```
C:\Users\DELL\Desktop\kjhg.exe
x=2 y=6 z=14
-----
Process exited after 0.1198 seconds with return value 0
Press any key to continue . . .
```

تم ارجاع ثلاث قيم