

Funciones con parámetros



Por valor y referencia



Ejemplo 3 - Por valor y referencia

```
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void operaciones(int, int, int&, int&);
7
8  int main()
9  {
10     int numA=7, numB=15;
11     int suma=0, mult=0;
12
13     operaciones(numA, numB, suma, mult);
14
15     cout<<"La suma es: "<<suma<<endl;
16     cout<<"El producto es: "<<mult<<endl;
17
18     getch();
19     return 0;
20 }
21
22 void operaciones (int numA, int numB, int& suma, int& mult)
23 {
24     suma = numA + numB;
25     mult = numA * numB;
26 }
```

En el ejemplo se crea un programa que calcula la suma y el producto de dos valores, con una sola función que debe generar ambos resultados, esto solamente es posible a través de funciones con parámetros por referencia, ya que es importante recordar que las funciones con parámetros por valor retornan un único valor.

En este ejemplo también se ilustra la utilización de parámetros por valor y parámetros por referencia en la misma función.

Ejemplo 3 - Por valor y referencia

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void operaciones(int, int, int&, int&);
7
8  int main()
9  {
10     int numA=7, numB=15;
11     int suma=0, mult=0;
12
13     operaciones(numA, numB, suma, mult);
14
15     cout<<"La suma es: "<<suma<<endl;
16     cout<<"El producto es: "<<mult<<endl;
17
18     getch();
19     return 0;
20 }
21
22 void operaciones (int numA, int numB, int& suma, int& mult)
23 {
24     suma = numA + numB;
25     mult = numA * numB;
26 }

```

- Línea 6: se escribe el prototipo de la función indicando que sus parámetros serán dos por valor de tipo entero y dos por referencia de tipo entero, basta con indicar el tipo sin necesidad de incluir ningún nombre.
- Líneas 10 y 11: se definen las variables que contienen los números a operar y las que van a contener los resultados de las operaciones.
- Línea 13: se hace el llamado a la función con los cuatro parámetros (las variables de los dos números y las variables de los dos resultados).
- Líneas 15 y 16: se imprimen en pantalla los resultados de las variables
- Línea 22: se define la función *operaciones* indicando el tipo y nombre de los parámetros por valor y por referencia.
- Líneas 24 y 25: se realizan las operaciones que modificarán la información de los parámetros que fueron pasados por referencia.

Ejemplo 3 - Por valor y referencia

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void operaciones(int, int, int&, int&);
7
8  int main()
9  {
10     int numA=7, numB=15;
11     int suma=0, mult=0;
12
13     operaciones(numA, numB, suma, mult);
14
15     cout<<"La suma es: "<<suma<<endl;
16     cout<<"El producto es: "<<mult<<endl;
17
18     getch();
19     return 0;
20 }
21
22 void operaciones (int numA, int numB, int& suma, int& mult)
23 {
24     suma = numA + numB;
25     mult = numA * numB;
26 }

```

El orden de ejecución del programa anterior es el siguiente:

1. Se hace el llamado a la función operaciones con los valores 7, 15, 0, 0.
2. Se realiza para la variable *suma* la operación 7 + 15.
3. Se realiza para la variable *mult* la operación 7 * 15.
4. La función termina y entonces el programa regresa al punto en que ésta fue llamada.
5. Se imprime el nuevo resultado de la variable *suma* 22.
6. Se imprime el nuevo resultado de la variable *mult* 105.

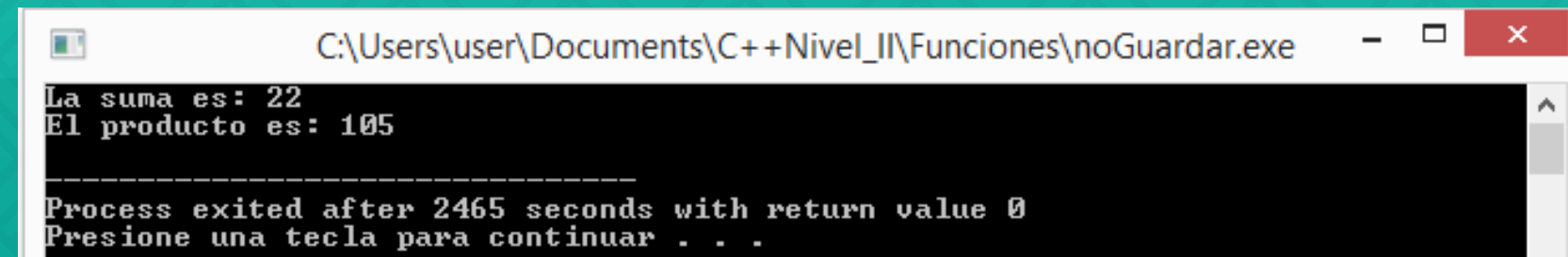
Ejemplo 3 - Por valor y referencia

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void operaciones(int, int, int&, int&);
7
8  int main()
9  {
10     int numA=7, numB=15;
11     int suma=0, mult=0;
12
13     operaciones(numA, numB, suma, mult);
14
15     cout<<"La suma es: "<<suma<<endl;
16     cout<<"El producto es: "<<mult<<endl;
17
18     getch();
19     return 0;
20 }
21
22 void operaciones (int numA, int numB, int& suma, int& mult)
23 {
24     suma = numA + numB;
25     mult = numA * numB;
26 }

```

Teniendo en cuenta este orden, el resultado devuelto al ejecutar el programa se observa a continuación:



```

C:\Users\user\Documents\C++Nivel_II\Funciones\noGuardar.exe
La suma es: 22
El producto es: 105

-----
Process exited after 2465 seconds with return value 0
Presione una tecla para continuar . . .

```


Ejemplo 3 - Por valor y referencia

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void operaciones(int, int, int&, int&);
7
8  int main()
9  {
10     int numA=7, numB=15;
11     int suma=0, mult=0;
12
13     operaciones(numA, numB, suma, mult);
14
15     cout<<"La suma es: "<<suma<<endl;
16     cout<<"El producto es: "<<mult<<endl;
17
18     getch();
19     return 0;
20 }
21
22 void operaciones (int numA, int numB, int& suma, int& mult)
23 {
24     suma = numA + numB;
25     mult = numA * numB;
26 }

```

Como se puede ver, los valores de los parámetros *suma* y *mult* no conservaron su valor original que era cero, sino que la función operaciones al tener la dirección de memoria de las variables *suma* y *mult* pudo modificarlas, por esta razón cuando el programa regresa al punto en que la función fue llamada, es posible imprimir en pantalla el nuevo resultado de dichas variables y se ha logrado que con una función se obtenga más de un valor de retorno.

En conclusión, en el paso de parámetros por valor lo que se realiza entonces es enviar a la función una copia de la variable original para que sea utilizada con el fin de llevar a cabo acciones u operaciones dentro de la función a la que se le envía, por esta razón esa variable original se mantiene intacta y las funciones no tienen la propiedad de modificarla.