

Funciones con parámetros



Por referencia



Ejemplo 2 - Por referencia

```
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }
```

Como se observa, el código es prácticamente igual, la diferencia radica en el prototipo de la función (Línea 6) y en el inicio de la declaración de la misma (Línea 19).

Después de indicar el tipo del parámetro, se debe agregar un signo ampersand (&) que le indica al programa que no debe pasar como parámetro una copia de la variable sino la dirección en memoria de la misma, ya que se requiere que el valor contenido en ella sea modificado después de ejecutada la función que es llamada.

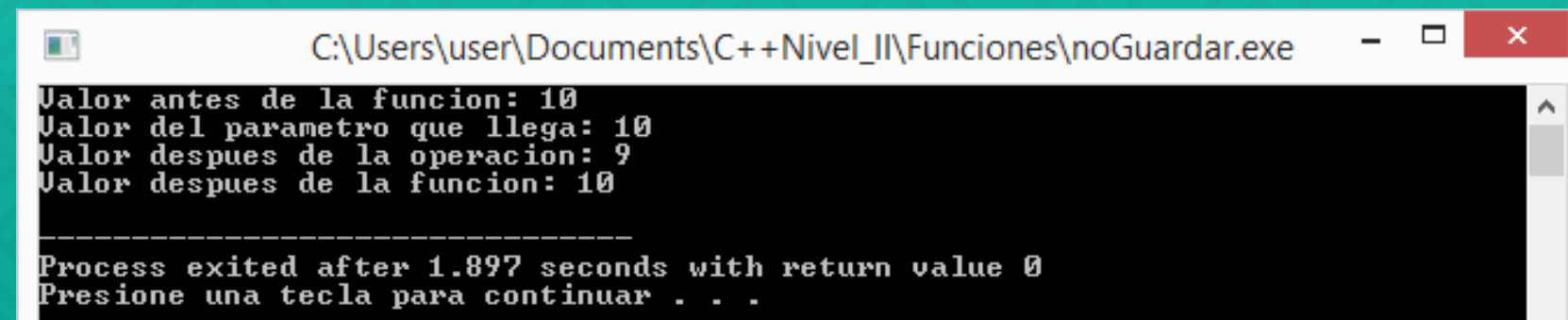
Ejemplo 2 - Por referencia

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }

```

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
Valor antes de la funcion: 10
Valor del parametro que llega: 10
Valor despues de la operacion: 9
Valor despues de la funcion: 10

-----
Process exited after 1.897 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejemplo 2 - Por referencia

```

1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void llamarFuncion(int valor);
7
8  int main()
9  {
10     int numero=10;
11
12     cout<<"Valor antes de la funcion: "<<numero<<endl;
13     llamarFuncion(numero);
14     cout<<"Valor despues de la funcion: "<<numero<<endl;
15
16     return 0;
17 }
18
19 void llamarFuncion (int valor)
20 {
21     cout<<"Valor del parametro que llega: "<<valor<<endl;
22     valor --;
23     cout<<"Valor despues de la operacion: "<<valor<<endl;
24 }

```

Como se puede ver, el valor del parámetro se mantiene en la función *main* antes y después de la ejecución de la función *llamarFuncion*, dicho parámetro se utiliza para realizar la operación y su valor inicial queda alterado según la operación realizada, por esta razón el valor después de la operación y el valor después de la función es 9.

En conclusión, en el paso de parámetros por referencia lo que se hace entonces es enviar a la función la dirección de la variable original para que sea empleada con el fin de realizar acciones u operaciones dentro de la función a la que se le envía y posteriormente que su valor sea modificado también en la función inicial.