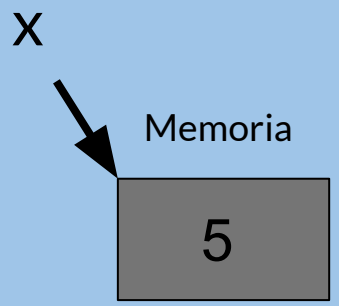
Pasaje de parámetros en C++

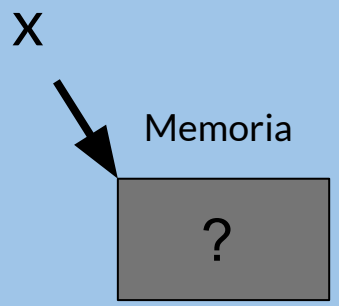
Porción de memoria de una variable

```
1 #include <iostream>
2
3 int main() {
4    int x = 5;
5
6    return 0;
7}
```



Porción de memoria de una variable

```
1 #include <iostream>
2
3 int main() {
4    int x;
5
6    return 0;
7}
```



Pasaje de parámetros por valor (o por copia)

¿Cuanto vale y luego de ejecutarse cambiarValor ?

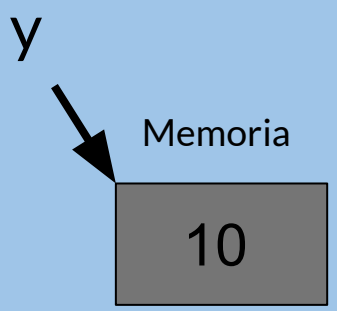
Ejemplo 1

```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```

¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 1

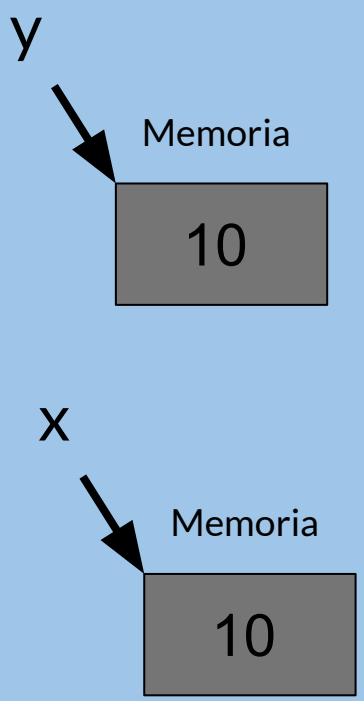
```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 1

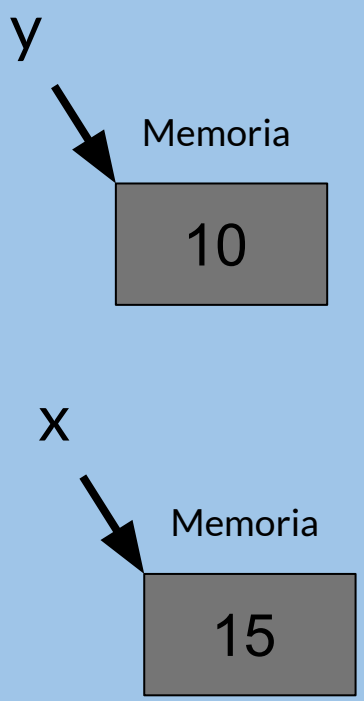
```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 1

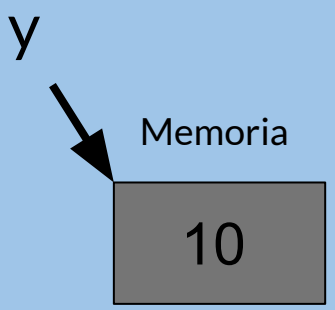
```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 1

```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cómo cambiamos el código si queremos que cambie el valor de y ?

¿Cómo cambiamos el código si queremos que cambie el valor de y ?

Ejemplo 1

```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4       x = 15;
5 }
6
7 int main(){
8      int y = 10;
9      cambiarValor(y);
10     std::cout <<"Y: "<< y <<endl;
11     return 0;
12}
```

Ejemplo 2

```
1 #include <iostream>
2
3 int cambiarValor(int x) {
4       x = 15;
5       return x;
6 }
7
8 int main(){
9      int y = 10;
10     y = cambiarValor(y);
11     std::cout <<"Y: "<< y <<endl;
12     return 0;
13}
```

¿Cuanto vale y luego de ejecutarse cambiarValor ?

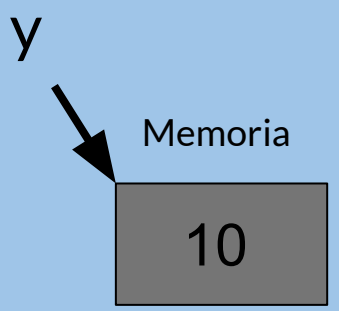
Ejemplo 2

```
1 #include <iostream>
2
3 int cambiarValor(int x) {
4        x = 15;
5        return x;
6 }
7
8 int main(){
9       int y = 10;
10      y = cambiarValor(y);
11      std::cout <<"Y: "<< y <<endl;
12      return 0;
13}
```

¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 2

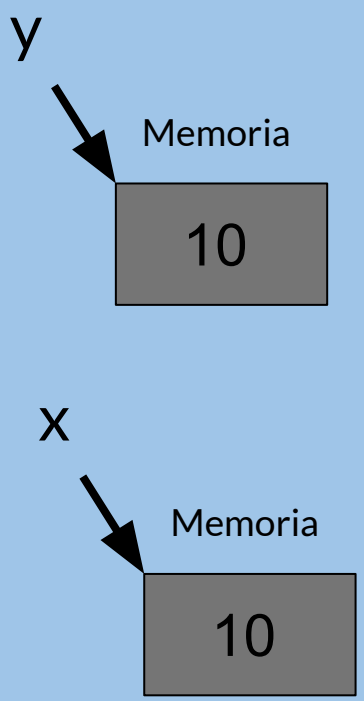
```
1 #include <iostream>
2
3 int cambiarValor(int x) {
4        x = 15;
5        return x;
6 }
7
8 int main(){
9       int y = 10;
10       y = cambiarValor(y);
11       std::cout <<"Y: "<< y <<endl;
12       return 0;
13}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 2

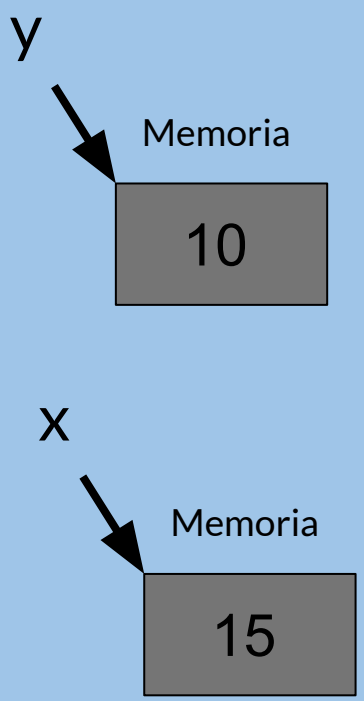
```
1 #include <iostream>
2
3 int cambiarValor(int x) {
4        x = 15;
5        return x;
6 }
7
8 int main(){
9       int y = 10;
10       y = cambiarValor(y);
11       std::cout <<"Y: "<< y <<endl;
12       return 0;
13}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 2

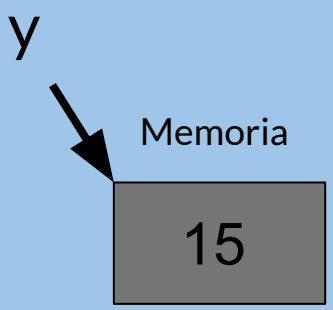
```
1 #include <iostream>
2
3 int cambiarValor(int x) {
4        x = 15;
5        return x;
6 }
7
8 int main(){
9       int y = 10;
10       y = cambiarValor(y);
11       std::cout <<"Y: "<< y <<endl;
12       return 0;
13}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 2

```
1 #include <iostream>
2
3 int cambiarValor(int x) {
4        x = 15;
5        return x;
6 }
7
8 int main(){
9       int y = 10;
10      y = cambiarValor(y);
11      std::cout <<"Y: "<< y <<endl;
12      return 0;
13}
```



Pasaje de parámetros por valor (o por copia)

- Coloca en la posición de memoria del argumento de entrada el valor de la expresión usada en la invocación.
- Si la función modifica el valor, no se cambian las variables en el llamador.
- Declaración de la función: int f(int b);
- Invocación de la función: f(x), o bien f(x+5) o bien f(5).
- Es el modo por defecto de pasaje de argumentos en C++.

Pasaje de parámetros por referencia

Pasaje de parámetros por referencia

Ejemplo 1

```
1 #include <iostream>
2
3 void cambiarValor(int x) {
4       x = 15;
5 }
6
7 int main(){
8      int y = 10;
9      cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```

Ejemplo 3

```
1 #include <iostream>
2
3 void cambiarValor(int &x) {
4       x = 15;
5 }
6
7 int main(){
8      int y = 10;
9      cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```

¿Cuanto vale y luego de ejecutarse cambiarValor ?

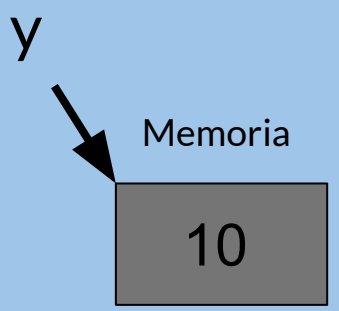
Ejemplo 3

```
1 #include <iostream>
2
3 void cambiarValor(int &x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```

¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 3

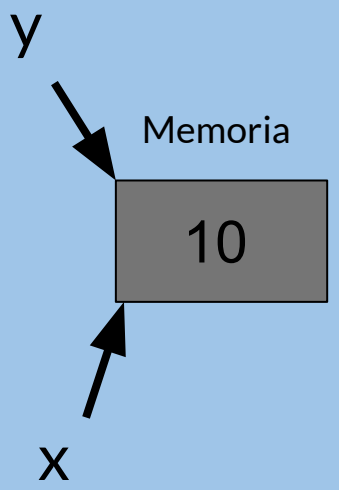
```
1 #include <iostream>
2
3 void cambiarValor(int &x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 3

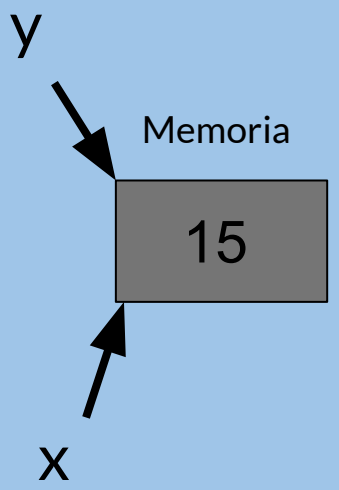
```
1 #include <iostream>
2
3 void cambiarValor(int &x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 3

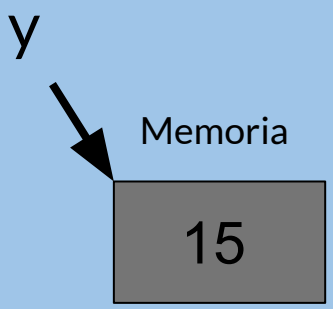
```
1 #include <iostream>
2
3 void cambiarValor(int &x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



¿Cuanto vale y luego de ejecutarse cambiarValor ?

Ejemplo 3

```
1 #include <iostream>
2
3 void cambiarValor(int &x) {
4        x = 15;
5 }
6
7 int main(){
8       int y = 10;
9       cambiarValor(y);
10      std::cout <<"Y: "<< y <<endl;
11      return 0;
12}
```



Aliasing

- Decimos una variable es un alias de otra variable si ambas apuntan a la misma porción de la memoria.

- El operador & permite indicar que usaremos la referencia en lugar de la copia de una variable.

Pasaje de parámetros por referencia

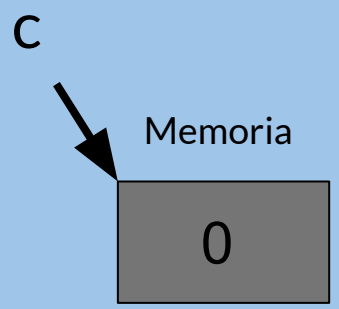
- La función recibe una dirección de memoria donde encontrar el argumento.
- La función puede leer esa posición de memoria pero también puede escribirla.
- Todas las asignaciones hechas dentro del cuerpo de la función cambian el contenido de la memoria del llamador.
- La expresión con la que se realiza la invocación debe ser necesariamente una variable.
- Invocación de la función: f(x), pero no f(x+5) ni f(5).
- Declaración de la función: int f(int &b);

Ejemplo: Aliasing

```
1 #include <iostream>
2
3 int cambiarValor(int &a, int &b) {
4       //cuanto vale a? cuanto vale b?
5       b = 3;
6       //cuanto vale a? cuanto vale b?
7       a = 4;
8       //cuanto vale a? cuanto vale b?
9       return 0;
10 }
11
12 int main(){
13      int c = 0;
14      cambiarValor(c, c);
15      return 0;
16}
```

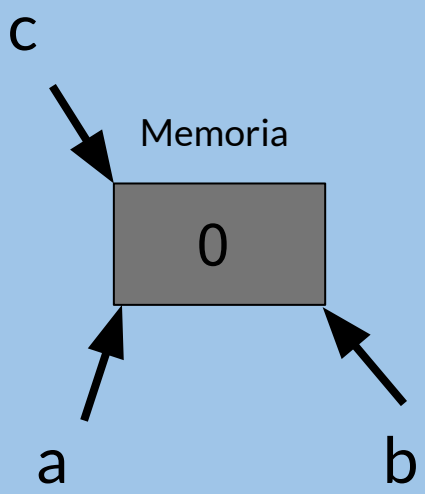
Ejemplo: Aliasing

```
1 #include <iostream>
2
3 int cambiarValor(int &a, int &b) {
4       //cuanto vale a? cuanto vale b?
5       b = 3;
6       //cuanto vale a? cuanto vale b?
7       a = 4;
8       //cuanto vale a? cuanto vale b?
9       return 0;
10 }
11
12 int main(){
13      int c = 0;
14      cambiarValor(c, c);
15      return 0;
16}
```



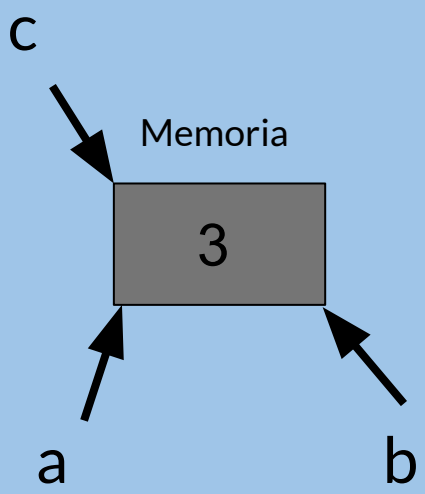
Ejercicio: Aliasing

```
1 #include <iostream>
2
3 int cambiarValor(int &a, int &b) {
4       //cuanto vale a? cuanto vale b?
5       b = 3;
6       //cuanto vale a? cuanto vale b?
7       a = 4;
8       //cuanto vale a? cuanto vale b?
9       return 0;
10 }
11
12 int main(){
13      int c = 0;
14      cambiarValor(c, c);
15      return 0;
16}
```



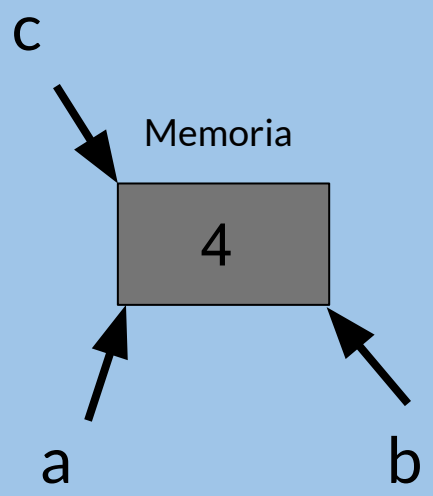
Ejercicio: Aliasing

```
1 #include <iostream>
2
3 int cambiarValor(int &a, int &b) {
4       //cuanto vale a? cuanto vale b?
5       b = 3;
6       //cuanto vale a? cuanto vale b?
7       a = 4;
8       //cuanto vale a? cuanto vale b?
9       return 0;
10 }
11
12 int main(){
13      int c = 0;
14      cambiarValor(c, c);
15      return 0;
16}
```



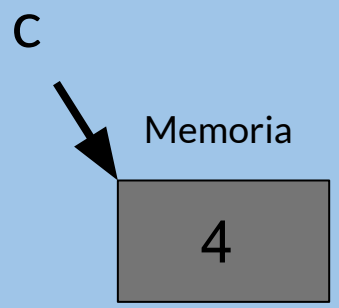
Ejercicio: Aliasing

```
1 #include <iostream>
2
3 int cambiarValor(int &a, int &b) {
4       //cuanto vale a? cuanto vale b?
5       b = 3;
6       //cuanto vale a? cuanto vale b?
7       a = 4;
8       //cuanto vale a? cuanto vale b?
9       return 0;
10 }
11
12 int main(){
13      int c = 0;
14      cambiarValor(c, c);
15      return 0;
16}
```



Ejercicio: Aliasing

```
1 #include <iostream>
2
3 int cambiarValor(int &a, int &b) {
4       //cuanto vale a? cuanto vale b?
5       b = 3;
6       //cuanto vale a? cuanto vale b?
7       a = 4;
8       //cuanto vale a? cuanto vale b?
9       return 0;
10 }
11
12 int main(){
13      int c = 0;
14      cambiarValor(c, c);
15      return 0;
16}
```



Parametros in, out, inout

- En nuestro lenguaje de especificación los parámetros de una función pueden ser in, out o inout.
  - in: parámetros de entrada
  - out: parámetros de salida
  - inout: parámetros de entrada y de salida
- ¿Que podemos usar en C++ para implementar cada uno de estos parámetros?
  - Para un parámetro in: un argumento que se pase por copia.
  - Para un parámetro inout: un argumento que se pase por referencia.
  - Para un parámetro out:
    - un argumento que se pase por referencia, o
    - el valor de retorno de la función