

## Ejercicio:

Construir una función que intercambie el valor de dos variables

```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char c1, char c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```

PILA

## Ejercicio:

Construir una función que intercambie el valor de dos variables

```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char c1, char c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```



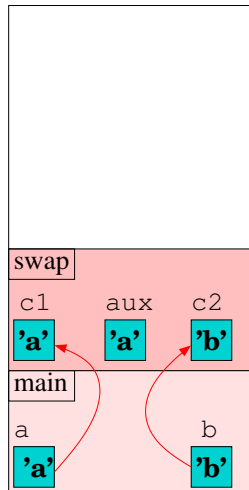
## Ejercicio:

Construir una función que intercambie el valor de dos variables

```

1  #include <iostream>
2  using namespace std;
3
4  void Swap(char c1, char c2){
5      char aux=c1;
6      c1=c2;
7      c2=aux;
8  }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15           << " y b=" << b << endl;
16 }

```



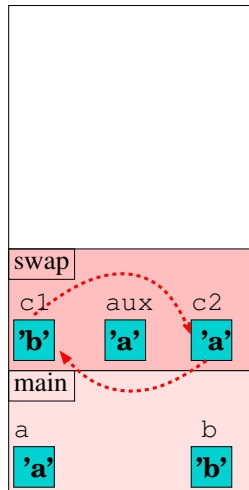
## Ejercicio:

Construir una función que intercambie el valor de dos variables

```

1  #include <iostream>
2  using namespace std;
3
4  void Swap(char c1, char c2){
5      char aux=c1;
6      c1=c2;
7      c2=aux;
8  }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15           << " y b=" << b << endl;
16 }

```



## Ejercicio:

Construir una función que intercambie el valor de dos variables

```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char c1, char c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15         << " y b=" << b << endl;
16 }
```



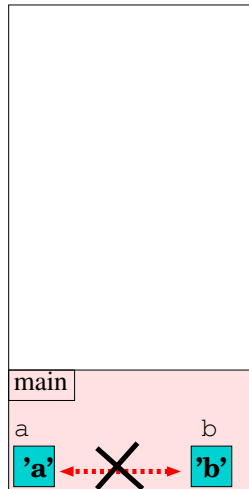
## Ejercicio:

Construir una función que intercambie el valor de dos variables

```

1  #include <iostream>
2  using namespace std;
3
4  void Swap(char c1, char c2){
5      char aux=c1;
6      c1=c2;
7      c2=aux;
8  }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15         << " y b=" << b << endl;
16 }

```



## Ejercicio:

Construir una función que intercambie el valor de dos variables

## Análisis

- Los valores de las variables a y b no se han modificado.

## Ejercicio:

Construir una función que intercambie el valor de dos variables

## Análisis

- Los valores de las variables a y b no se han modificado.
- Los que se intercambiaron fueron sus copias c1 y c2.



## Ejercicio:

Construir una función que intercambie el valor de dos variables

## Análisis

- Los valores de las variables  $a$  y  $b$  no se han modificado.
- Los que se intercambiaron fueron sus copias  $c1$  y  $c2$ .
- El problema es que se necesita extender el ámbito de  $a$  y  $b$  para que sean manipulables en el entorno de Swap.

## Ejercicio:

Construir una función que intercambie el valor de dos variables

## Análisis

- Los valores de las variables a y b no se han modificado.
- Los que se intercambiaron fueron sus copias c1 y c2.
- El problema es que se necesita extender el ámbito de a y b para que sean manipulables en el entorno de Swap.

## Solución

Paso de parámetros por referencia

## Ejercicio:

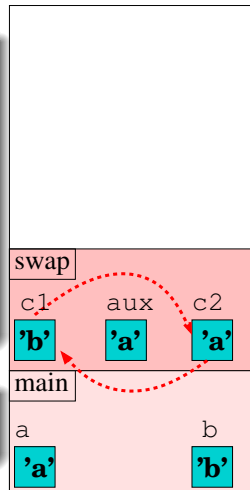
Construir una función que intercambie el valor de dos variables

## Análisis

- Los valores de las variables a y b no se han modificado.
- Los que se intercambiaron fueron sus copias c1 y c2.
- El problema es que se necesita extender el ámbito de a y b para que sean manipulables en el entorno de Swap.

## Solución

Paso de parámetros por referencia



## Paso de parámetros

### Por valor o copia

- Es el paso de argumentos por defecto.

## Paso de parámetros

### Por valor o copia

- Es el paso de argumentos por defecto.
- Durante la llamada se realiza una copia del parámetro actual en el parámetro formal.

## Paso de parámetros

### Por valor o copia

- Es el paso de argumentos por defecto.
- Durante la llamada se realiza una copia del parámetro actual en el parámetro formal.
- De esta forma, el módulo invocado trabaja con una copia y no con el valor original.

## Paso de parámetros

### Por valor o copia

- Es el paso de argumentos por defecto.
- Durante la llamada se realiza una copia del parámetro actual en el parámetro formal.
- De esta forma, el módulo invocado trabaja con una copia y no con el valor original.

Para resolver el problema del ejercicio anterior tenemos que trabajar con los datos originales y no con las copias.

## Paso de parámetros

### Por referencia o variable

- No realiza una copia del parámetro actual en el formal, sino un vínculo entre ellos, de tal forma que una modificación en el parámetro formal, conlleva la misma modificación en el parámetro actual.



## Paso de parámetros

### Por referencia o variable

- No realiza una copia del parámetro actual en el formal, sino un vínculo entre ellos, de tal forma que una modificación en el parámetro formal, conlleva la misma modificación en el parámetro actual.
- Se usa **&** entre el tipo y el identificador del argumento para indicar que el paso se realiza por referencia.

## Paso de parámetros

### Por referencia o variable

- No realiza una copia del parámetro actual en el formal, sino un vínculo entre ellos, de tal forma que una modificación en el parámetro formal, conlleva la misma modificación en el parámetro actual.
- Se usa **&** entre el tipo y el identificador del argumento para indicar que el paso se realiza por referencia.

### Ejemplos

```
1 void Swap (char &c1, char &c2);  
2 void Division (int dividendo, int divisor,  
3               int &coc, int &resto);  
4 void ElegirOpcion (char &opcion);
```

## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

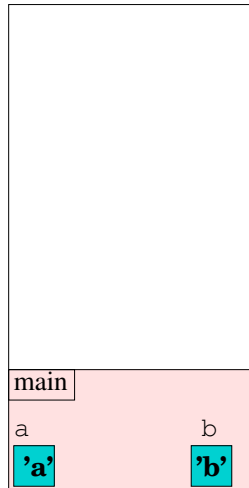
```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char &c1, char &c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```

PILA

## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char &c1, char &c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```



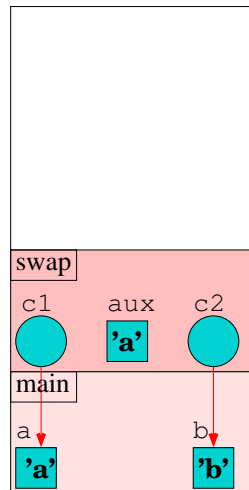
## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

```

1  #include <iostream>
2  using namespace std;
3
4  void Swap(char &c1, char &c2){
5      char aux=c1;
6      c1=c2;
7      c2=aux;
8  }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }

```



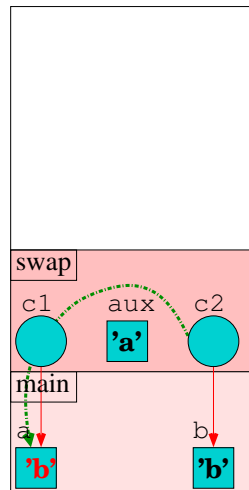
## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

```

1  #include <iostream>
2  using namespace std;
3
4  void Swap(char &c1, char &c2){
5      char aux=c1;
6      c1=c2;
7      c2=aux;
8  }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }

```



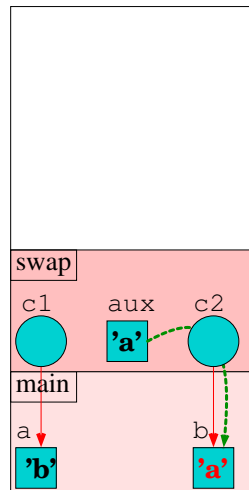
## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

```

1  #include <iostream>
2  using namespace std;
3
4  void Swap(char &c1, char &c2){
5      char aux=c1;
6      c1=c2;
7      c2=aux;
8  }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15         << " y b=" << b << endl;
16 }

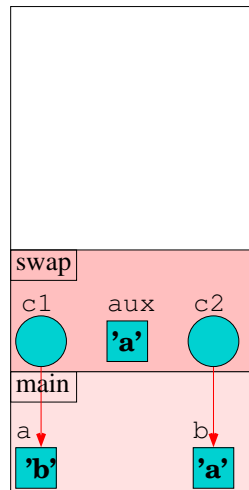
```



## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char &c1, char &c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```

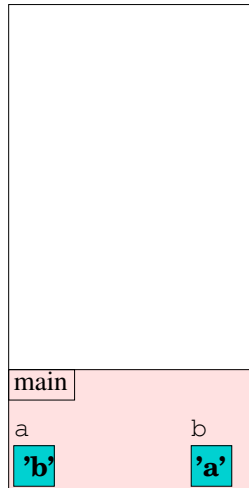




## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

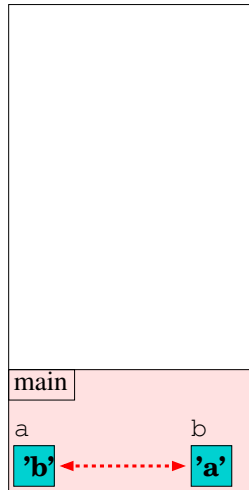
```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char &c1, char &c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```



## Solución del ejercicio

Construir una función que intercambie el valor de dos variables

```
1 #include <iostream>
2 using namespace std;
3
4 void Swap(char &c1, char &c2){
5     char aux=c1;
6     c1=c2;
7     c2=aux;
8 }
9
10 int main(){
11     char a='a', b='b';
12
13     Swap(a,b);
14     cout << "a=" << a
15          << " y b=" << b << endl;
16 }
```



# Valor/Referencia versus Entrada/Salida

## Valor/Referencia versus Entrada/Salida

La identificación de la información que aporta un argumento en una función nos indica la forma en la que debe ser pasado dicho argumento.

## Valor/Referencia versus Entrada/Salida

La identificación de la información que aporta un argumento en una función nos indica la forma en la que debe ser pasado dicho argumento.

- Si el argumento es usado como vehículo para obtener la solución, entonces nos encontramos ante un **parámetro de entrada**.

## Valor/Referencia versus Entrada/Salida

La identificación de la información que aporta un argumento en una función nos indica la forma en la que debe ser pasado dicho argumento.

- Si el argumento es usado como vehículo para obtener la solución, entonces nos encontramos ante un **parámetro de entrada**.

### PASO POR VALOR

## Valor/Referencia versus Entrada/Salida

La identificación de la información que aporta un argumento en una función nos indica la forma en la que debe ser pasado dicho argumento.

- Si el argumento es usado como vehículo para obtener la solución, entonces nos encontramos ante un **parámetro de entrada**.

### PASO POR VALOR

- Si el argumento es usado para almacenar la solución o parte de ella, entonces nos encontramos ante un **parámetro de salida**.

## Valor/Referencia versus Entrada/Salida

La identificación de la información que aporta un argumento en una función nos indica la forma en la que debe ser pasado dicho argumento.

- Si el argumento es usado como vehículo para obtener la solución, entonces nos encontramos ante un **parámetro de entrada**.

### PASO POR VALOR

- Si el argumento es usado para almacenar la solución o parte de ella, entonces nos encontramos ante un **parámetro de salida**.
- Si el argumento es tanto vehículo para obtener la solución como parte de la misma, entonces nos encontramos con un **parámetro de entrada/salida**.



# Valor/Referencia versus Entrada/Salida

La identificación de la información que aporta un argumento en una función nos indica la forma en la que debe ser pasado dicho argumento.

- Si el argumento es usado como vehículo para obtener la solución, entonces nos encontramos ante un **parámetro de entrada**.

## PASO POR VALOR

- Si el argumento es usado para almacenar la solución o parte de ella, entonces nos encontramos ante un **parámetro de salida**.
- Si el argumento es tanto vehículo para obtener la solución como parte de la misma, entonces nos encontramos con un **parámetro de entrada/salida**.

## PASO POR REFERENCIA

## Ejercicio: Escribir el prototipo de:

a) Determinar si un número es primo.

## Ejercicio: Escribir el prototipo de:

a) Determinar si un número es primo.

```
bool Es_Primo (int);
```

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.

```
int Num_Primos (int, int);
```

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).

```
void MaxAndMix (double &, double &);
```

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.



## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.

```
void Menu ();
```

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.
- e) Calcular la suma de dos números complejos. Supóngase que representamos un número complejo usando dos números reales.

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.
- e) Calcular la suma de dos números complejos. Supóngase que representamos un número complejo usando dos números reales.

```
void SumaComplejos (double c1r, double c1i,  
double c2r, double c2i, double &c3r, double &c3i);
```

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.
- e) Calcular la suma de dos números complejos. Supóngase que representamos un número complejo usando dos números reales.
- f) Calcular la derivada de un polinomio de grado 3.

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.
- e) Calcular la suma de dos números complejos. Supóngase que representamos un número complejo usando dos números reales.
- f) Calcular la derivada de un polinomio de grado 3.

```
void DerivadaPol (double a, double b, double c,  
double d, double &ad, double &bd, double &cd);
```

## Ejercicio: Escribir el prototipo de:

- a) Determinar si un número es primo.
- b) Calcular el número de primos existentes en un intervalo de valores.
- c) Calcular el máximo y el mínimo de una secuencia de valores reales introducidos por la entrada estándar (los valores se leen dentro de la función).
- d) Escribir por la salida estándar un menú.
- e) Calcular la suma de dos números complejos. Supóngase que representamos un número complejo usando dos números reales.
- f) Calcular la derivada de un polinomio de grado 3.

# Valor/Referencia versus Entrada/Salida

Debes tener en cuenta que

cuando el paso de parámetros es por valor, el argumento actual puede ser una expresión, una constante o una variable.

## Valor/Referencia versus Entrada/Salida

### Debes tener en cuenta que

cuando el paso de parámetros es por valor, el argumento actual puede ser una expresión, una constante o una variable.

### Sin embargo

cuando el paso de parámetros es por referencia, el argumento actual debe ser obligatoriamente una variable.