Funciones y Procedimientos

Funciones y Procedimientos

- Las funciones y procedimientos se utilizan para dividir un algoritmo en bloques distintos. Cada bloque se encarga de una operación diferente, por lo cual al ponerlas definidas como una estructura es posible llamarlas desde cualquier parte del algoritmo principal para que se ejecute la tarea requerida. A veces estos bloques son indispensables en otros programas, por lo que se crean librerías almacenando esta información e invocándolas luego en otros códigos.
- De igual manera, esta forma de programación resulta ventajosa para cuando se necesite programar modularmente, pues un equipo de desarrollo se encargaría de dividirse el trabajo que se necesita hacer y luego se implementarían todos estos códigos en un único main.

Funciones y Procedimientos

- Divide and conquer : Es una filosofía muy popular entre los programadores, proveniente de una frase del latín “Divide et impera” que quiere decir “Divide y reinaras”, hace alusión a que ante un problema difícil, este se debe dividir en partes mas simples lo suficiente como para que su resolución se torne menos complicada de implementar.
- Esta filosofía es fuertemente apoyada por la programación modular, pues cada modulo debe encargarse de una tarea mas simple para así poder resolver un algoritmo mas complejo, esto es en esencia el objetivo de este estilo de programación.

Funciones y Procedimientos

- Funciones: las funciones son una especie de sub programa que nos ayuda ejecutar operaciones dentro de un código principal. El flujo del algoritmo se ve interrumpido a la hora de invocar una función, la cual al terminar, retorna una variable de un tipo de dato especifico, justamente en el lugar del código donde fue llamada la función.

Funciones y Procedimientos

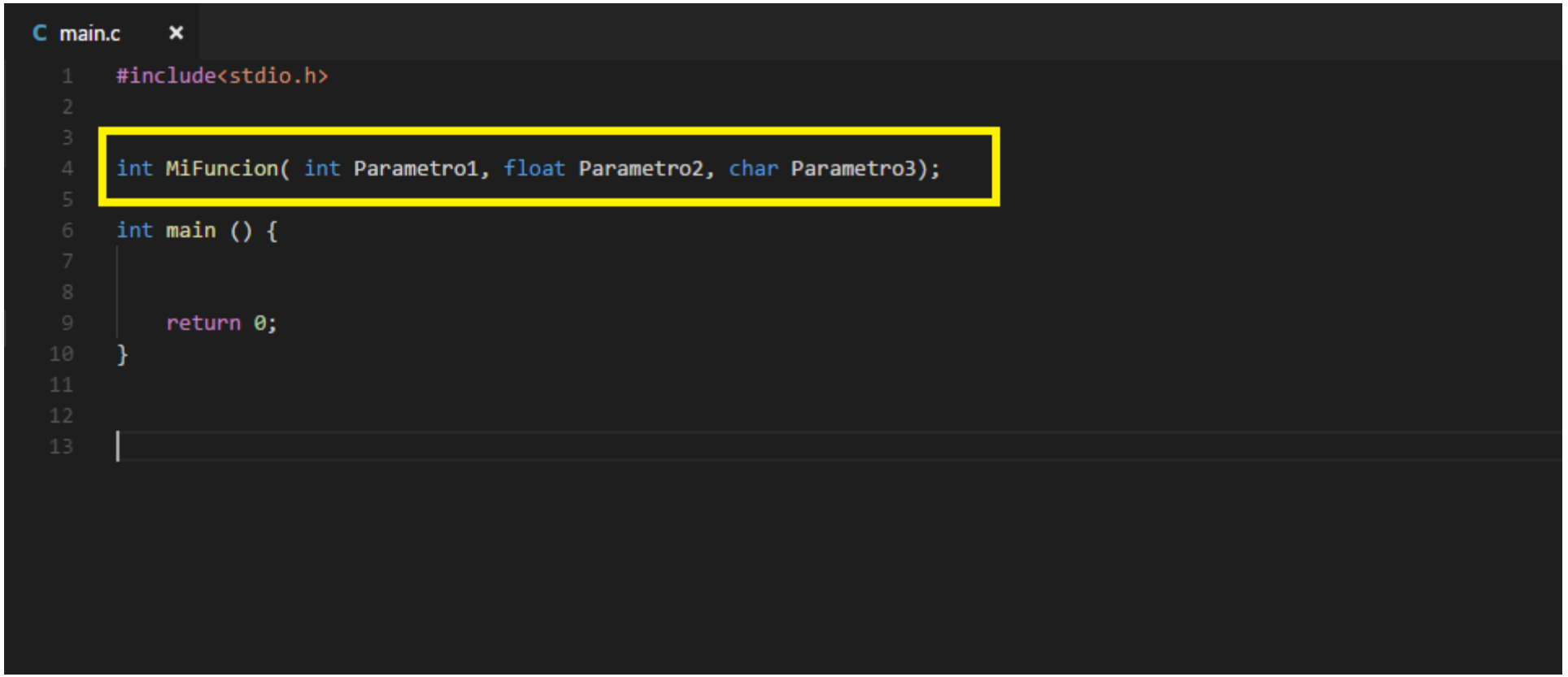
- La función puede o no necesitar parámetros, los cuales son variables que se le pueden pasar a la función a la hora de ser invocadas, sin embargo, cuando estos datos son pasados a la función, la función solo hace un copia de estos datos y trabaja con ellos, lo que quiere decir que todos los posibles cambios que se le puedan hacer a la variable solo se quedan en la función y por esto la variable en el programa principal queda intacta.
- Las únicas tres formas en las que se puede cambiar una variable pasada por parámetros son : mediante un puntero, pase por referencia y por ultimo que cuando la función retorne el valor este sea asignado a la variable que se quiera cambiar.

Funciones y Procedimientos

- Forma de declararlas : Hay varias formas de declarar una función dentro de un programa, la mas básica es primero definir el prototipo de la función. El prototipo es la forma básica de la función que se quiere implementar. El prototipo no es necesario para que el programa corra o compile la función, su sentido es mas por una buena praxis de programación para poder entender fácilmente el código de otra persona.

- Para declarar el prototipo se debe poner primero el tipo de dato que se desea retornar, luego de esto se escribe el nombre de la función seguido de un paréntesis, dentro de ellos se escriben los parámetros requeridos de la función, estos serán descritos por el tipo de dato que son mas el nombre que ellos tendrán dentro de esta función. Para separarlos se debe escribir una coma después de cada parámetro, finalmente fuera de los paréntesis se escribe “;”.

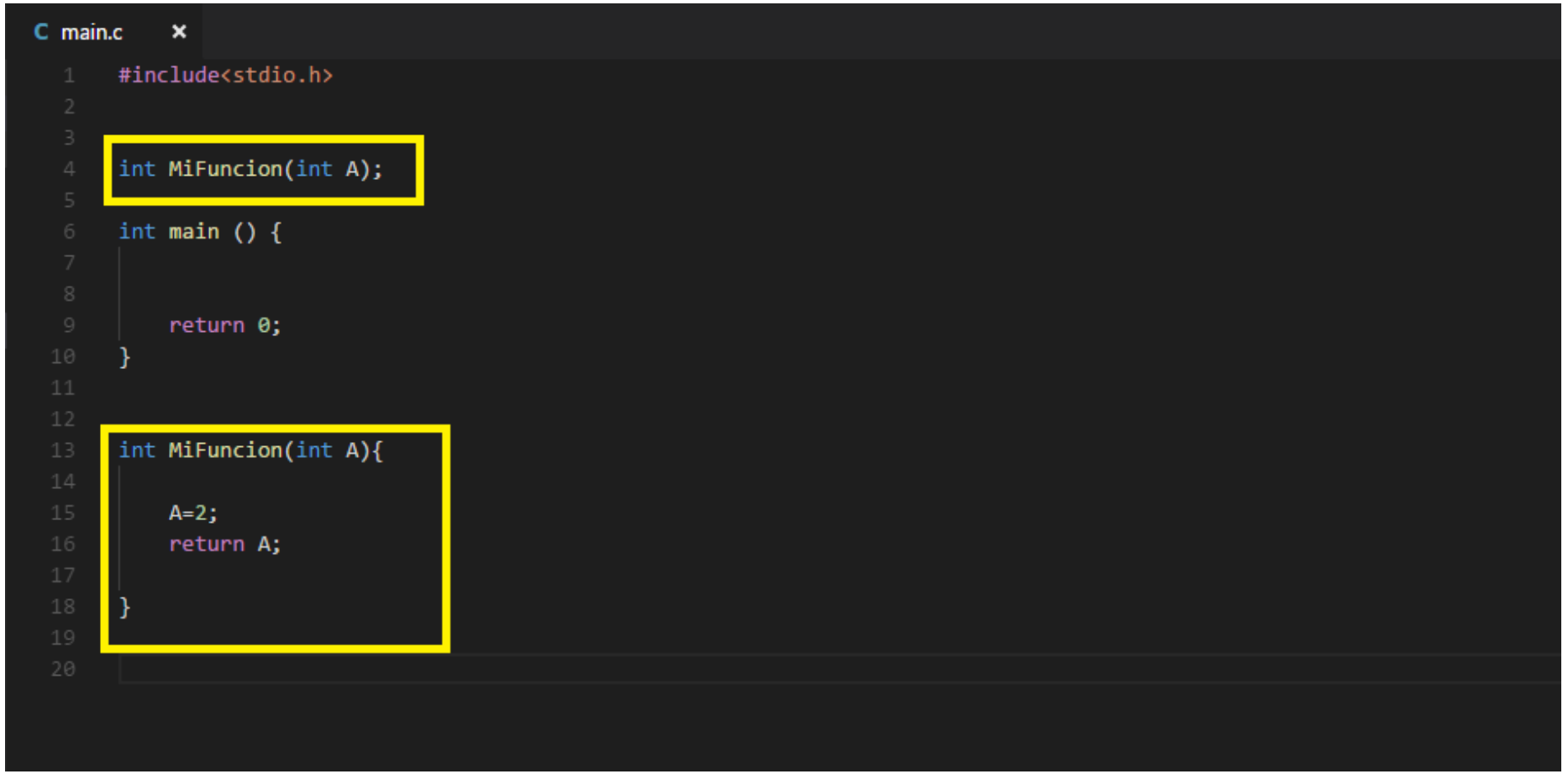
Funciones y Procedimientos



Funciones y Procedimientos

- Luego de haber declarado el prototipo de la función deberemos declarar la función luego del main, para esto deberemos escribir exactamente lo mismo que en el prototipo solo que con la diferencia que en lugar de poner ";" al final deberemos abrir llaves, las cuales tendrán dentro el cuerpo de la función. El cuerpo es donde se desarrolla la función, es en este pedazo del algoritmo donde declaramos lo que nuestra función va a hacer y que es lo que va a retornar cuando sea llamada. Ellas pueden alojar variables, sentencias, ciclos de repetición e inclusive otras funciones. Finalmente no debemos de olvidarnos de la sentencia "return" que es la sentencia en la que nosotros retornamos el valor que queramos a la variable donde ha sido llamada la función. El return debe estar seguido de la variable que queramos retornar y luego cerrarlos con ";" como cualquier otra sentencia, solo que en este caso esta sentencia siempre tiene que estar declarada y retornando un valor del mismo tipo de dato que el que se declaro la función al principio.

Funciones y Procedimientos



Funciones y Procedimientos

- Modo de uso: Para poder utilizar nuestra función debemos invocarla. Esta invocación puede ser hecha dentro de un main, función o procedimiento cualquiera. Para poder llevar a cabo la invocación solo hace falta escribir el nombre de la función y pasarle los parámetros que ella requiera, por ejemplo si declaramos una función con dos parámetros de tipo entero, entonces deberemos pasarle 2 variables de tipo entero. Es importante resaltar que la función siempre retorna una variable y esta debe ser guardada en algún lugar, así que por lo general, la funciones se invocan en una sentencia de asignación para una variable. También es importante decir que la variable en la cual le hagamos la asignación debe ser el mismo tipo de dato que el que retorna la función.

Funciones y Procedimientos

main.c

```c
#include<stdio.h>


int MiFuncion(int A);

int main () {
    int aux=0;

    MiFuncion(aux);

    return 0;
}


int MiFuncion(int A){

    A=2;
    return A;

}
```

Funciones y Procedimientos

- En este caso vimos como cambiar una variable con un función por medio de la asignación al retornar, también podemos cambiarlas por medio de un puntero

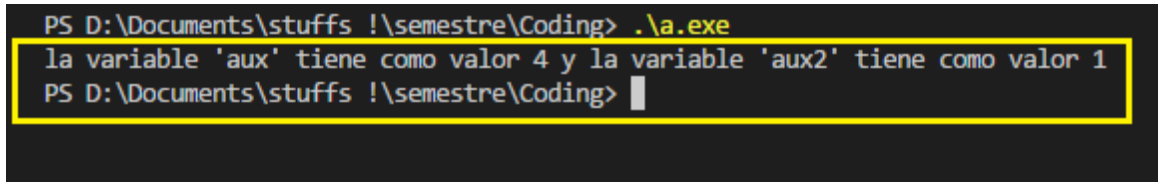
- Para poder hacer esto es necesario pasarle el parámetro puntero a la función, y a continuación trabajamos con el como cualquier otro puntero.

Funciones y Procedimientos

main.c

```c
#include<stdio.h>


int MiFuncion(int * A);

int main () {

    int * aux;
    int aux2=0;

    aux2=MiFuncion(aux);

    printf("la variable 'aux' tiene como valor %i y la variable 'aux2' tiene como valor %i",*aux,aux2);

    return 0;
}


int MiFuncion(int * A){
    int b=1;

    *A=b+3;

    return b;

}
```

Funciones y Procedimientos



Funciones y Procedimientos

- Finalmente para poder pasarla por referencia tan solo le pasamos la dirección de la variable, esto se logra escribiendo ampersand (“&”) luego del nombre de la variable que se quiere pasar. Al igual que en el anterior ejemplo debemos recordar que como estamos pasando una dirección dentro de la funcion se trabajara esta variable como puntero.

Funciones y Procedimientos

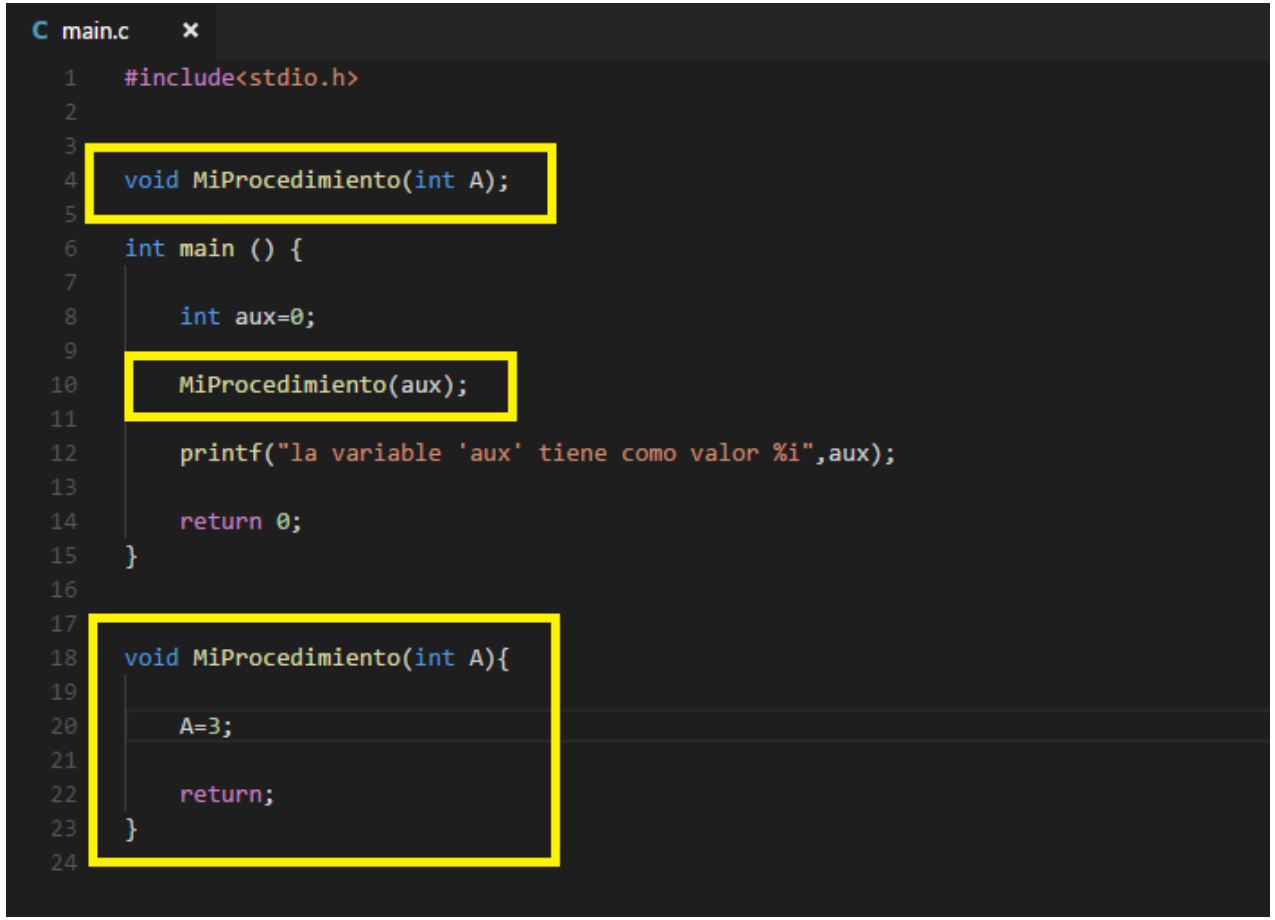
main.c

```c
#include<stdio.h>


int MiFuncion(int * A);

int main () {

    int aux=0;
    int aux2=0;

    aux2=MiFuncion(&aux);

    printf("la variable 'aux' tiene como valor %i y la variable 'aux2' tiene como valor %i",aux,aux2);

    return 0;
}


int MiFuncion(int * A){
    int b=1;

    *A=b+3;

    return b;

}

```

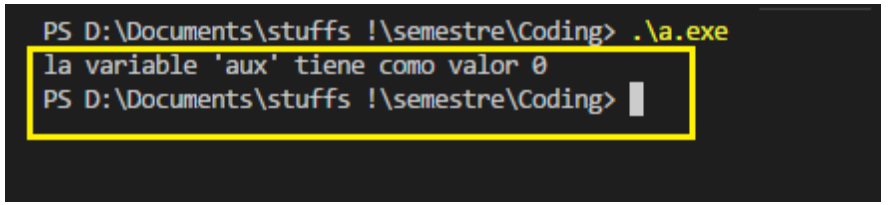
Funciones y Procedimientos

- Procedimientos : Procedimientos es exactamente igual que funciones, la única diferencia es que Procedimientos no retorna ningún valor,
- Para declarar el prototipo del procedimiento debemos escribir "void" seguido del nombre del procedimiento y entre paréntesis los parámetros del procedimiento, de la misma manera que hacíamos con funciones.
- Para desarrollarlo hacemos los mismos pasos que en funciones, No es necesario poner "return" sin embargo la buena praxis de programación nos indica que es mejor ponerlo al final para evitar ambigüedades.
- Es importante entender que cuando se le pasan parámetros ya sea a una funcion o a un procedimiento estas son solo copias de las variables que están en el main, por lo tanto todos los cambios que se hagan dentro de estas solo se conservaran ahí , es decir que la variable en el main no se vera afectada ante ninguna operación dentro de la funcion y/o procedimiento.
- Para invocarla tan solo escribimos el nombre del procedimiento seguido de paréntesis y los parámetros requeridos, luego de esto cerramos con ";". Recordemos que al ser un procedimiento no devuelve nada así lo podemos declarar en cualquier parte del código.

Funciones y Procedimientos



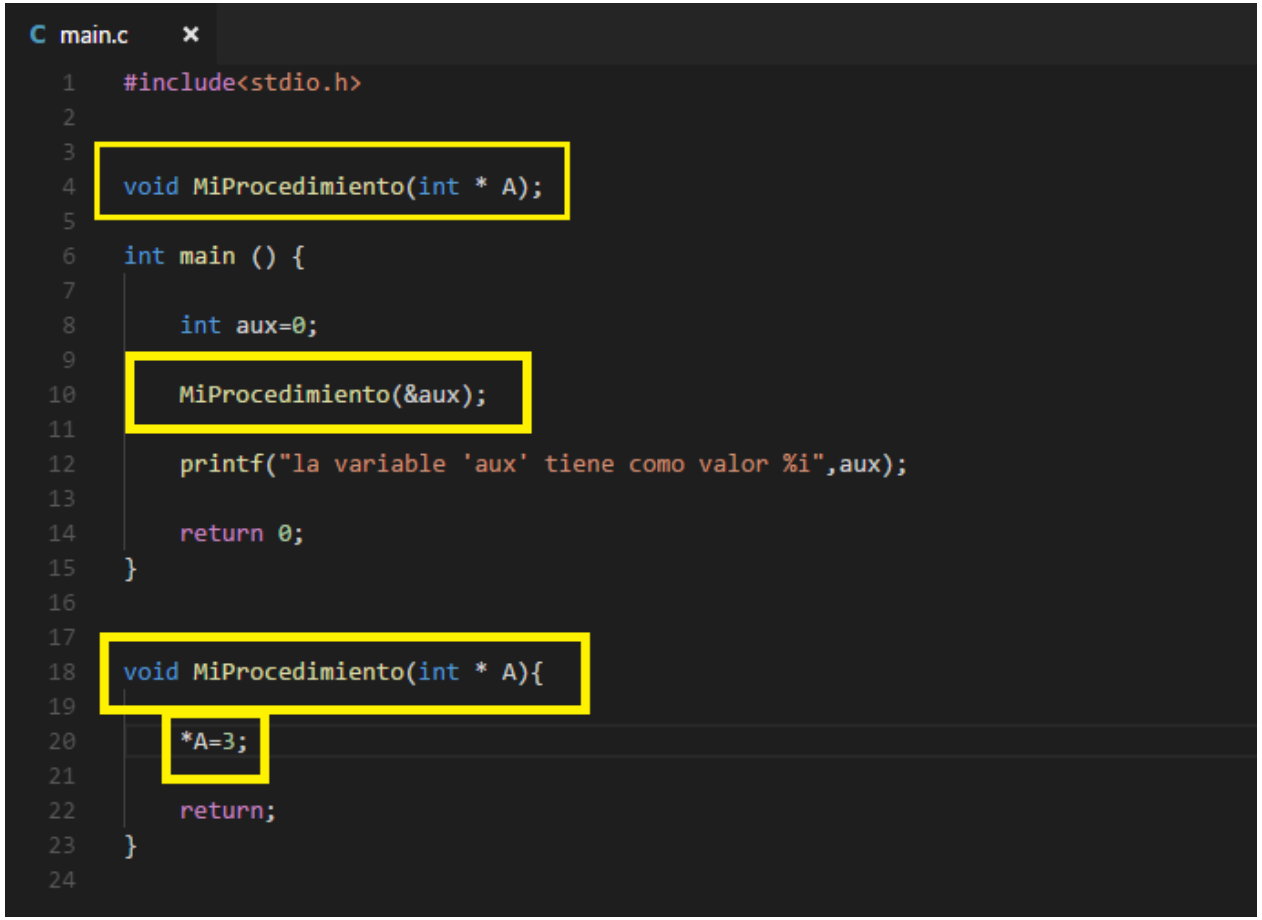
Funciones y Procedimientos



Funciones y Procedimientos

- Para poder cambiar variables en el main tan solo procedemos a los dos métodos ya vistos en funciones, es decir, por puntero y/o por referencia. Estos dos métodos son exactamente iguales tanto para funciones como para procedimientos

Funciones y Procedimientos



Funciones y Procedimientos