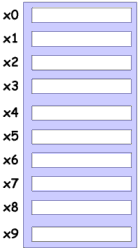
Autoestudio 9: Arreglos

Hay ocasiones en las que necesitamos trabajar con muchas variables, por ejemplo, supongamos que necesitamos trabajar con 10 valores enteros; el código para definir a estas 10 variables sería algo como lo siguiente:



**int x0;**

**int x1;**

**int x2;**

**int x3;**

**int x4;**

**int x5;**

**int x6;**

**int x7;**

**int x8;**

**int x9;**

El código donde se asigne a cada variable el valor -1 , sería el siguiente

**x0=-1;**

**x1=-1;**

**x2=-1;**

**x3=-1;**

**x4=-1;**

**x5=-1;**

**x6=-1;**

**x7=-1;**

**x8=-1;**

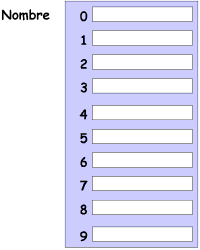
**x9=-1;**

Y cada vez que se requiera modificar los 10 valores, necesitamos escribir el código de cada asignación.

¿Qué tal si en lugar de trabajar con 10 necesitamos definir 50?, ¿Y si necesitamos definir 1000?

Para lidiar con estas situaciones se creó lo que se conoce como **arreglos**.

Un **arreglo** es una asociación entre **un nombre** y **un grupo de localidades.**  Cada localidad está asociada con un número, de tal manera que para identificar a la localidad específica del arreglo es necesario escribir su número.

La **primera** localidad siempre corresponde a la localidad **0** y la **última** corresponde a **n-1**, en donde **n** es el número de localidades del arreglo.

Por ejemplo, en un arreglo de **10** localidades, la primera localidad es **0** y la última localidad es **9**.

La característica de un arreglo son los corchetes**[ ]**. Una definición de variables que los incluya, indica que la variable es un **arreglo.**

La declaración de una variable arreglo tiene la siguiente forma:

**tipo** **nombre[tamaño];**

En donde **tipo** es el tipo de datos (int,char, float, etc…) y **nombre** es el nombre de la variable. Los corchetes **[ ]** indican que es una variable de tipo arreglo. Y tamaño el número de casillas.

La declaración de la variable arreglo **arg\_num** que contiene 15 valores enteros es la siguiente:

**int** **arg\_num[15];**

La declaración de la variable arreglo **arg\_letras** que contiene 30 valores de tipo char es la siguiente:

**char** **arg**\_**B[30];**

Una vez que hemos declarado el arreglo, para obtener el valor de una localidad específica del arreglo debemos escribir el **nombre de la variable y entre los corchetes el número de la localidad,** recuerda que las localidades de un arreglo están numeradas de **0** a **n-1** (en donde **n** es el tamaño del arreglo)**.**

Así, por ejemplo, para obtener el primer y último valor de arg\_num a usaríamos

**arg\_num[0]; // primer valor**

**arg\_num[14] // ultimo valor**

La forma de asignar un valor a una localidad específica del arreglo es la siguiente:

**nombre[ localidad ] = valor;**

En donde **nombre** es el nombre del arreglo, **localidad** es el número de la localidad del arreglo (entre **0** y **n-1**, donde **n** es el tamaño del arreglo) y **valor** es cualquier dato del tipo con que fue definido el arreglo.

así para darle valores al arreglo arg\_num haríamos:

**arg\_num[0] = 2;**

**arg\_num[1] = 4;**

**arg\_num[2] = 6;**

**arg\_num[3] = 8;**

**arg\_num[4] = 10;**

**arg\_num[5] = 12;**

**…**

**arg\_num[14] = 30;**

y la instrucción:

**printf(“ la posición %i guarda el valor %i”, 5, arg\_num[5]);**

Imprimiría en consola :

**la posición 5 guarda el valor 12**

La estructura compañera de los arreglos es el ciclo **for**. La mayoría de los problemas de arreglos son codificados con una estructura básica de **for**:

**for (int i=0; i<n; i++) { }**

en donde la variable del ciclo (i) toma los valores:

**0, 1, 2, ... , n-1**

que son precisamente los números de cada localidad de un arreglo. Cada pasada del ciclo (es decir, en cada valor de i) corresponde al código de cada localidad (la localidad i).

## Arreglos como parámetros en funciones y procedimientos

Cuando declaramos una función o procedimiento que recibe un arreglo como parámetro tenemos que declararlo con corchete como se muestra en el ejemplo:

**void inicializa (int arreglo[10]) {**

for (int i=0; i<10; i++)

arreglo[i] = i;

**}**

Cuando mandamos a llamar una función y le pasamos a un arreglo como parámetro, no es necesario poner los corchetes:

int main() {

int **arreglo[10]**;

inicializa(**arreglo**);

}

Hay que destacar que a diferencia de las variables comunes, los arreglos por ser una estructura de memoria pueden modificar los valores de sus localidades cuando se utilizan como parámetros de una función o procedimiento (tal y como pasa con las variables globales). Entonces, tomando el ejemplo de código anterior, si utilizamos el procedimiento inicializa dentro del main()y desplegamos cualquiera de las localidades del arreglo en la consola, la localidad tendrá el valor asignado en el procedimiento incializa.

Ver ejemplo en presentación Power Point anexa.

# Si deseas saber más puedes consultar, donde se explican los temas a mayor detalle:

(Joyanes Aguilar, L. y Zahonero Martínez, I. *Programación en C, C++, Java y UML*. México : McGraw Hill, 2010.) las siguientes secciones:

* Arreglos. (pp. 235)
* Inicialización de un arreglo. (pp. 238)