|  |
| --- |
| Laboratorio de Computación  Salas A y B |
| |  |  | | --- | --- | | Profesor: | Claudia Rodríguez Espino | | Asignatura: | Fundamentos de Programación | | Grupo: | 1104 | | No de Práctica(s): | 11 | | Integrante(s): | Téllez Torres Lorena Alejandra | |  |  | |  |  | |  |  | | No. de Equipo de cómputo empleado: | 51 | | Semestre: | 1ª | | Fecha de entrega: |  | | Observaciones: |  | |  |  | |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Objetivo**

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

**Desarrollo**

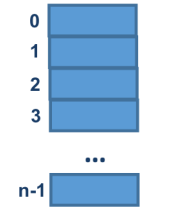
Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.

**Arreglos Unidimensionales**

Un arreglo unidimensional de n elementos en la memoria se almacena de la siguiente manera:



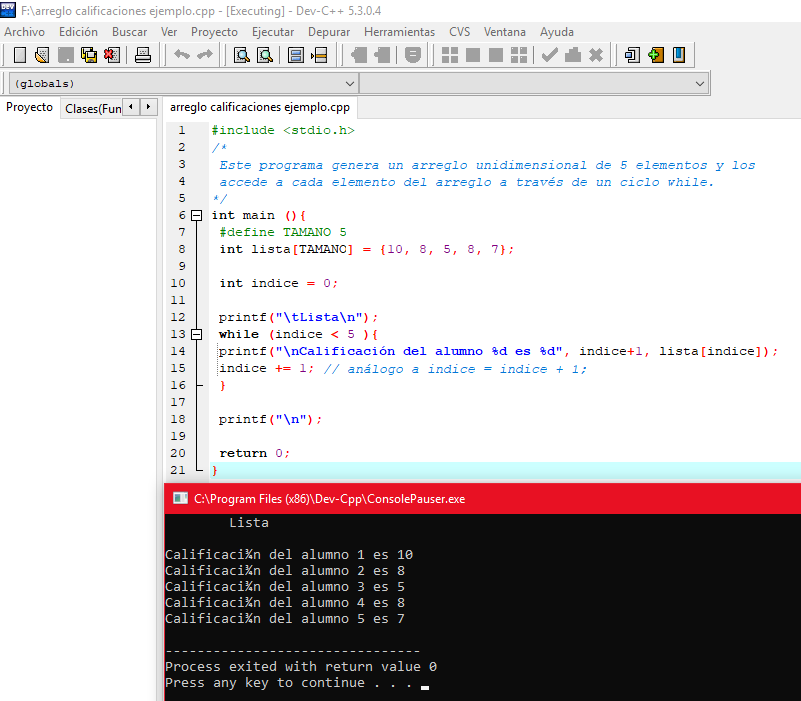
La primera localidad del arreglo corresponde al índice 0 y la última corresponde al índice n-1, donde n es el tamaño del arreglo.

La sintaxis para definir un arreglo en lenguaje C es la siguiente:

*tipoDeDato nombre[tamaño]*

Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo. Un arreglo puede ser de los tipos de dato entero, real, carácter o estructura.

**Código (arreglo unidimensional while)**



**Apuntadores**

Un apuntador es una variable que contiene la dirección de una variable, es decir, hace referencia a la localidad de memoria de otra variable. Debido a que los apuntadores trabajan directamente con la memoria, a través de ellos se accede con rapidez a un dato.

La sintaxis para declarar un apuntador y para asignarle la dirección de memoria de otra variable es, respectivamente:

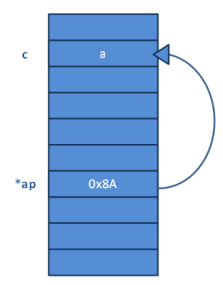
*TipoDeDato \*apuntador, variable;*

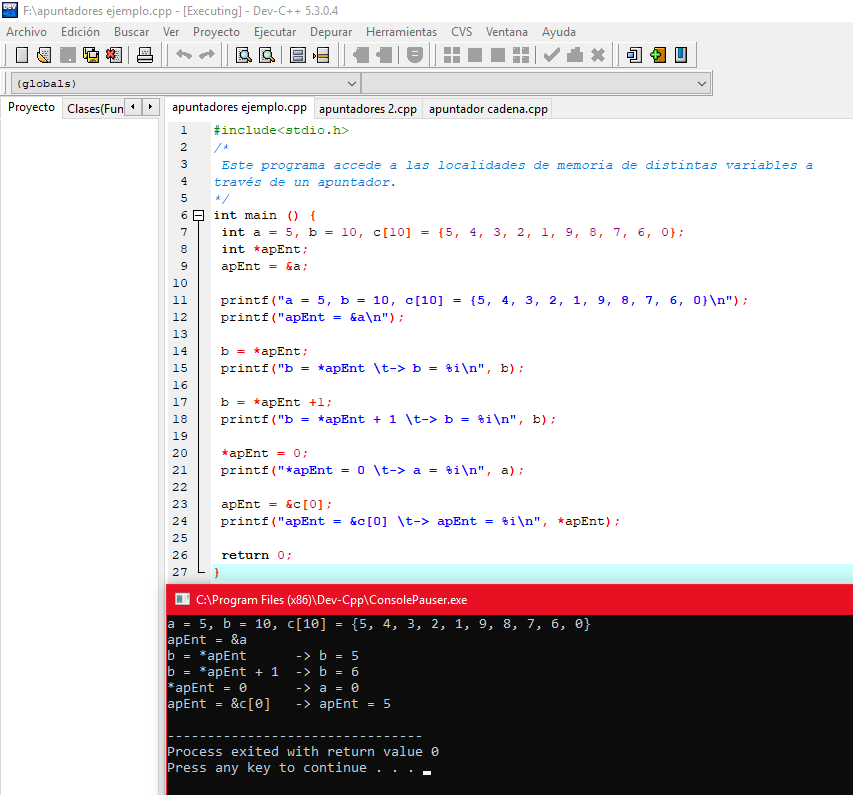
*apuntador = &variable;*

La declaración de una variable apuntador inicia con el carácter \*. Cuando a una variable le antecede un ampersand, lo que se hace es acceder a la dirección de memoria de la misma (es lo que pasa cuando se lee un dato con scanf).

Los apuntadores solo pueden apuntar a direcciones de memoria del mismo tipo de dato con el que fueron declarados; para acceder al contenido de dicha dirección, a la variable apuntador se le antepone \*.

Un apuntador almacena la dirección de memoria de la variable a la que apunta.

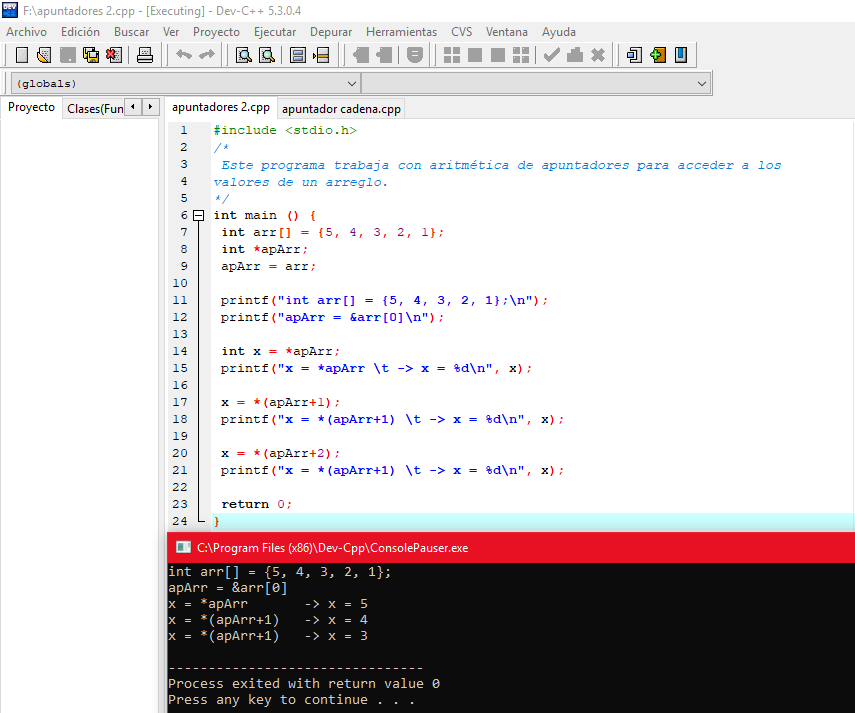




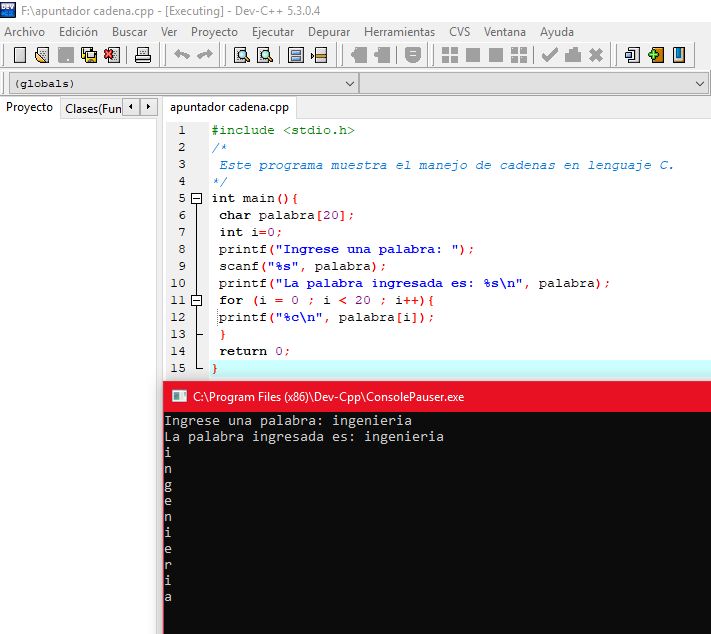
Cabe mencionar que el nombre de un arreglo es un apuntador fijo al primero de sus elementos; por lo que las siguientes instrucciones, para el código de arriba, son equivalentes:

*apEnt = &c[0];*

*apEnt = c;*



**Código (apuntadores en cadenas)**



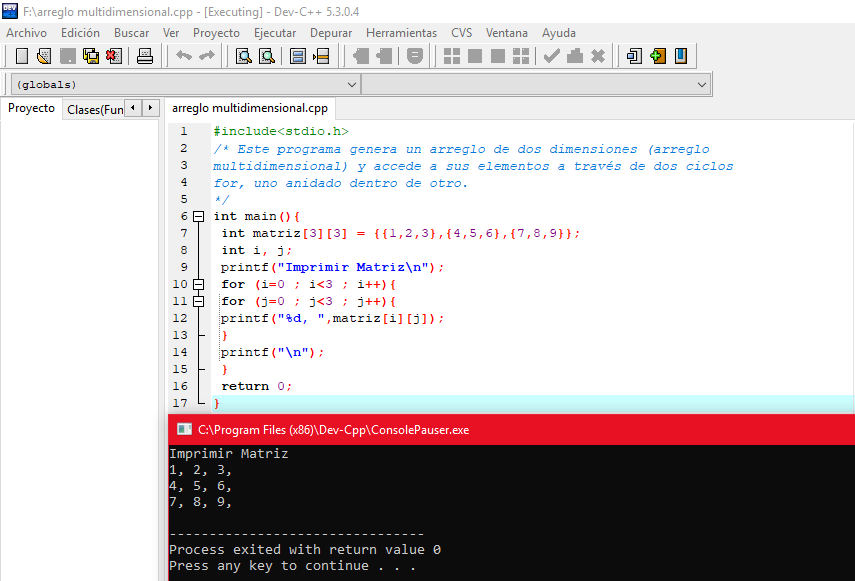
**Arreglos multidimensionales**

Lenguaje C permite crear arreglos de varias dimensiones con la siguiente sintaxis:

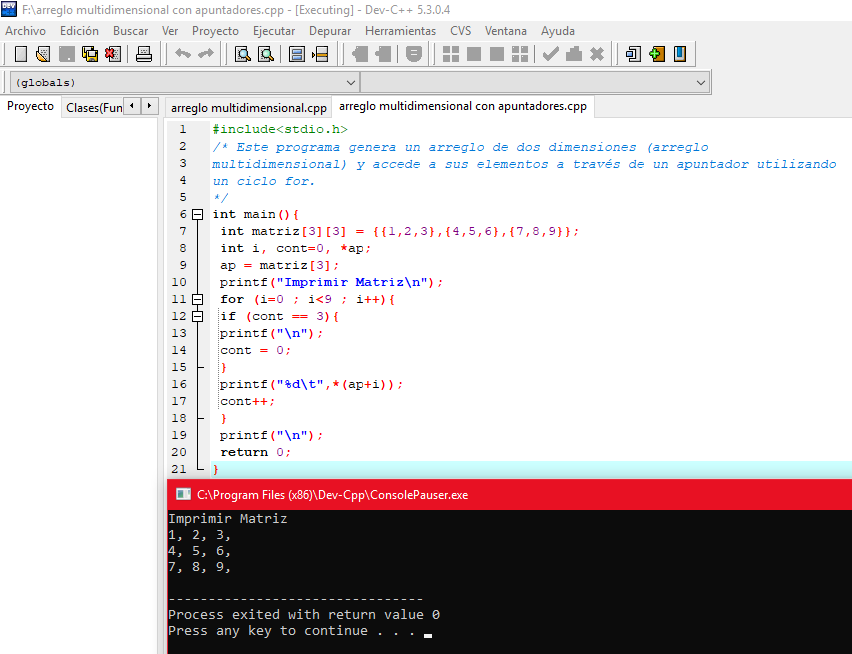
*tipoDato nombre[ tamaño ][ tamaño ]…[tamaño];*

Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimensión (el número de dimensiones está determinado por el número de corchetes). Los tipos de dato que puede tolerar un arreglo multidimensional son: entero, real, carácter o estructura.

De manera práctica se puede considerar que la primera dimensión corresponde a los renglones, la segunda a las columnas, la tercera al plano, y así sucesivamente. Sin embargo, en la memoria cada elemento del arreglo se guarda de forma contigua, por lo tanto, se puede recorrer un arreglo multidimensional con apuntadores.

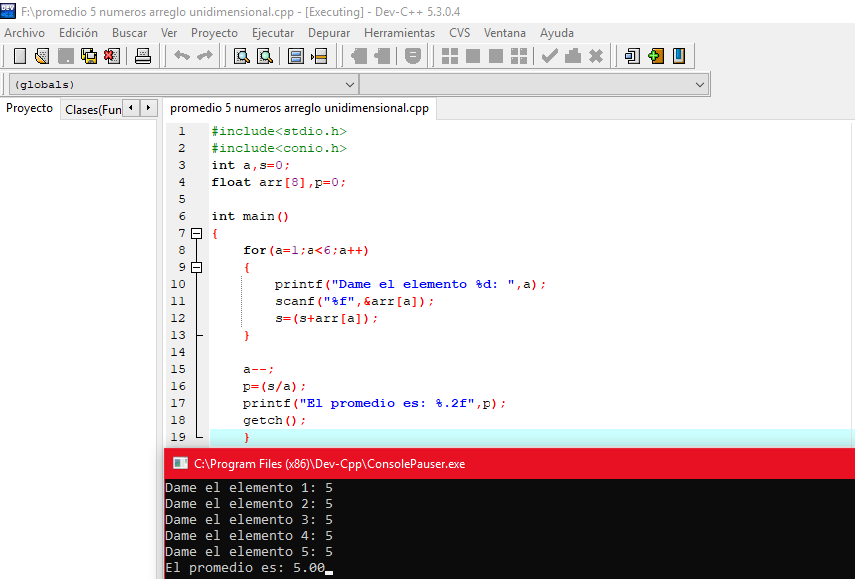
**Código (arreglos multidimensionales)**

**Código (arreglos multidimensionales con apuntadores)**

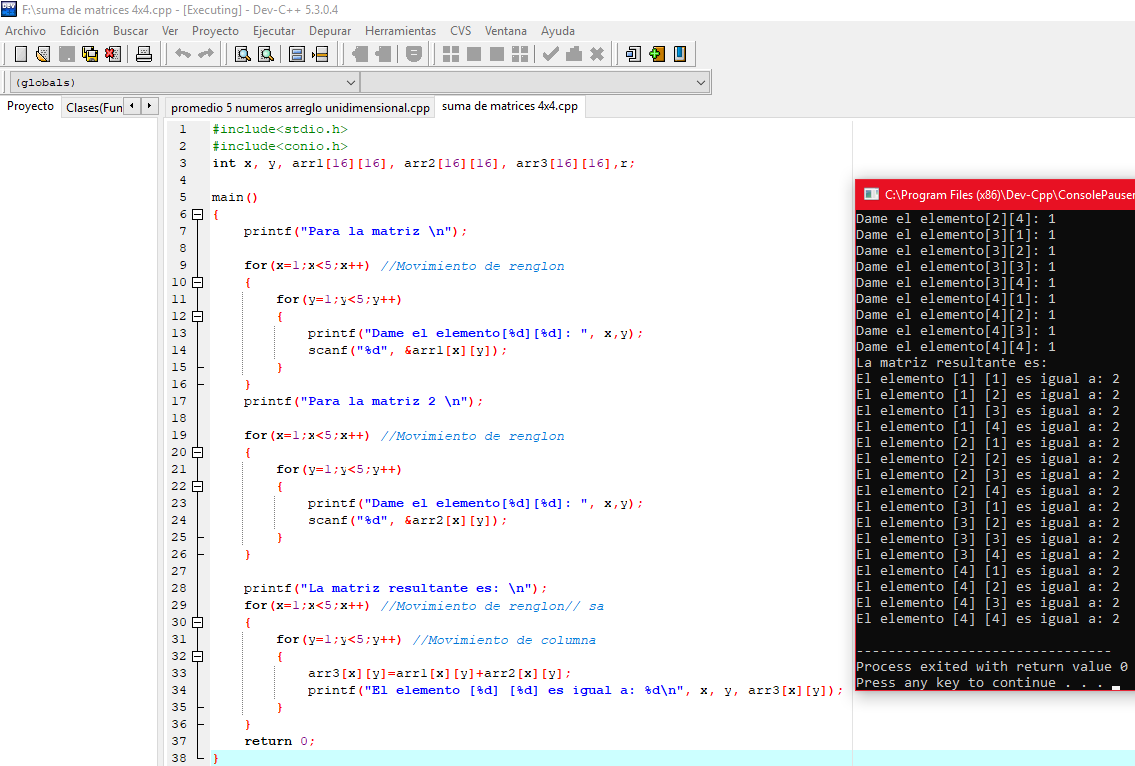


**Ejercicios**

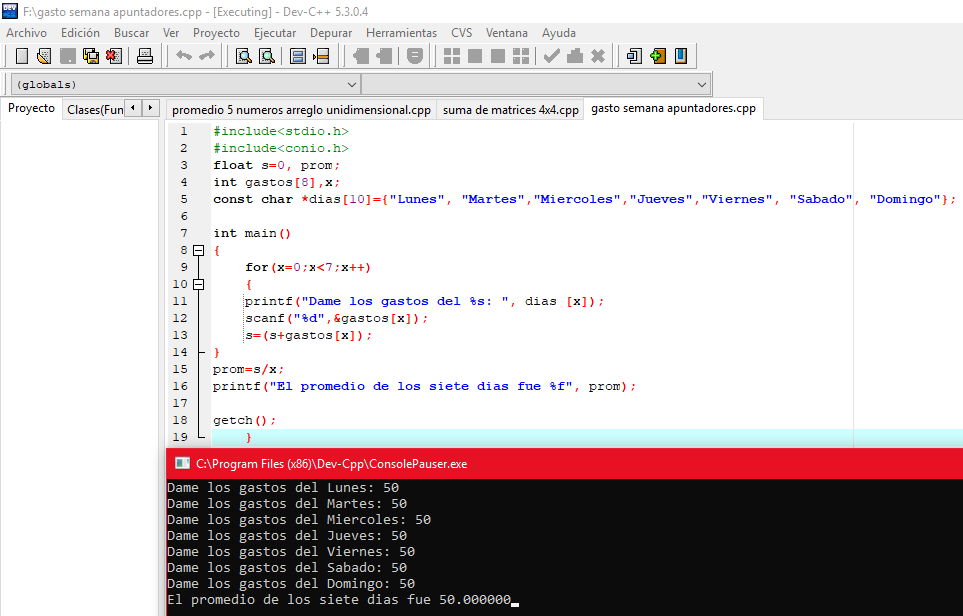
1. **Obtener el promedio de 5 números (arreglo unidimensional)**

****

1. **Suma de matrices 4X4 (arreglo bidimensional)**

****

1. **Gasto de una semana (apuntadores)**

****

**Conclusiones**

En esta práctica aprendí a reforzar mis conocimientos en cuanto al tema de arreglos respecta, además para mí los apuntadores son un poco confusos, pero esto me ayudó a usarlos y sobretodo entenderlos de una mejor manera. Creo que programando y programando vas a mejorar como desarrollador así que siempre es excelente entender un tema lo mejor posible y saber aplicarlo para la resolución de problemas.