|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
|  | RODRIGUEZ ESPINO CLAUDIA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACIÓN |
| *Grupo:* | 1104 |
| *No de Práctica(s):* | PRÁCTICA 11 |
| *Integrante(s):* | VALENCIA MEDINA ISAC |
|  |  |
|  |  |
| *Semestre:* | PRIMER SEMESTRE 2019-1 |
| *Fecha de entrega:* | 22/OCTUBRE/2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

GUÍA PRÁCTICA DE ESTUDIO 11: Arreglo unidimensionales y multidimensionales

**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.

**ACTIVIDADES:**

Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.

Resolver un problema que requiera el uso de un arreglo de dos dimensiones, a través de un programa en lenguaje C.

Manipular arreglos a través de índices y apuntadores.

**INTRODUCCIÓN:**

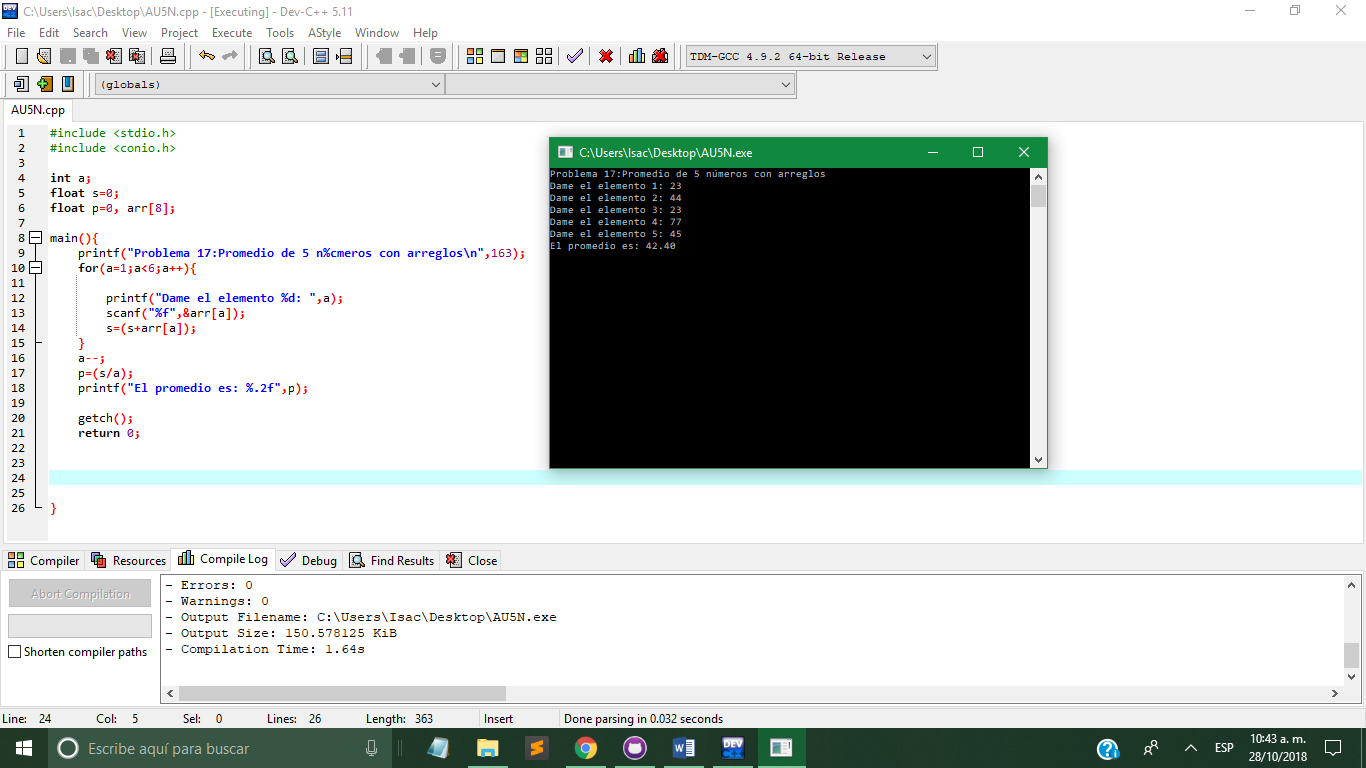
Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

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.

Problema 1: Obtener el promedio de 5 números con arreglo unidimensional

|  |
| --- |
| #include <stdio.h> |
|  | #include <conio.h> |
|  |  |
|  | int a; |
|  | float s=0; |
|  | float p=0, arr[8]; |
|  |  |
|  | main(){ |
|  | printf("Problema 17:Promedio de 5 n%cmeros con arreglos\n",163); |
|  | for(a=1;a<6;a++){ |
|  |  |
|  | printf("Dame el elemento %d: ",a); |
|  | scanf("%f",&arr[a]); |
|  | s=(s+arr[a]); |
|  | } |
|  | a--; |
|  | p=(s/a); |
|  | printf("El promedio es: %.2f",p); |
|  |  |
|  | getch(); |
|  | return 0; |
|  |  |
|  |  |
|  |  |
|  |  |
|  | } |



Problema 2: Suma de matrices (4x4) con arreglo bidimensional

#include <stdio.h>

#include <conio.h>

#define p printf

#define s scanf

int matriz1[4][4];

int matriz2[4][4];

int matriz3[5][5];

int x,y;

main(){

p("\t\t\tSuma de matrices\n");

p("Para la matriz 1\n");

for(x=1;x<=4;x++){

for(y=1;y<=4;y++){

p("Dame el valor para la localidad [%d][%d]: ",x,y);

s("%d",&matriz1[x][y]);

}

}

p("Para la matriz 2\n");

for(x=1;x<=4;x++){

for(y=1;y<=4;y++){

p("Dame el valor para la localidad [%d][%d]: ",x,y);

s("%d",&matriz2[x][y]);

}

}

p("Para la matriz resultante:\n");

for(x=1;x<=4;x++){

for(y=1;y<=4;y++){

matriz3[x][y]=matriz1[x][y] + matriz2[x][y];

p("Para la localidad [%d][%d] la suma es: %d\n",x,y,matriz3[x][y]);

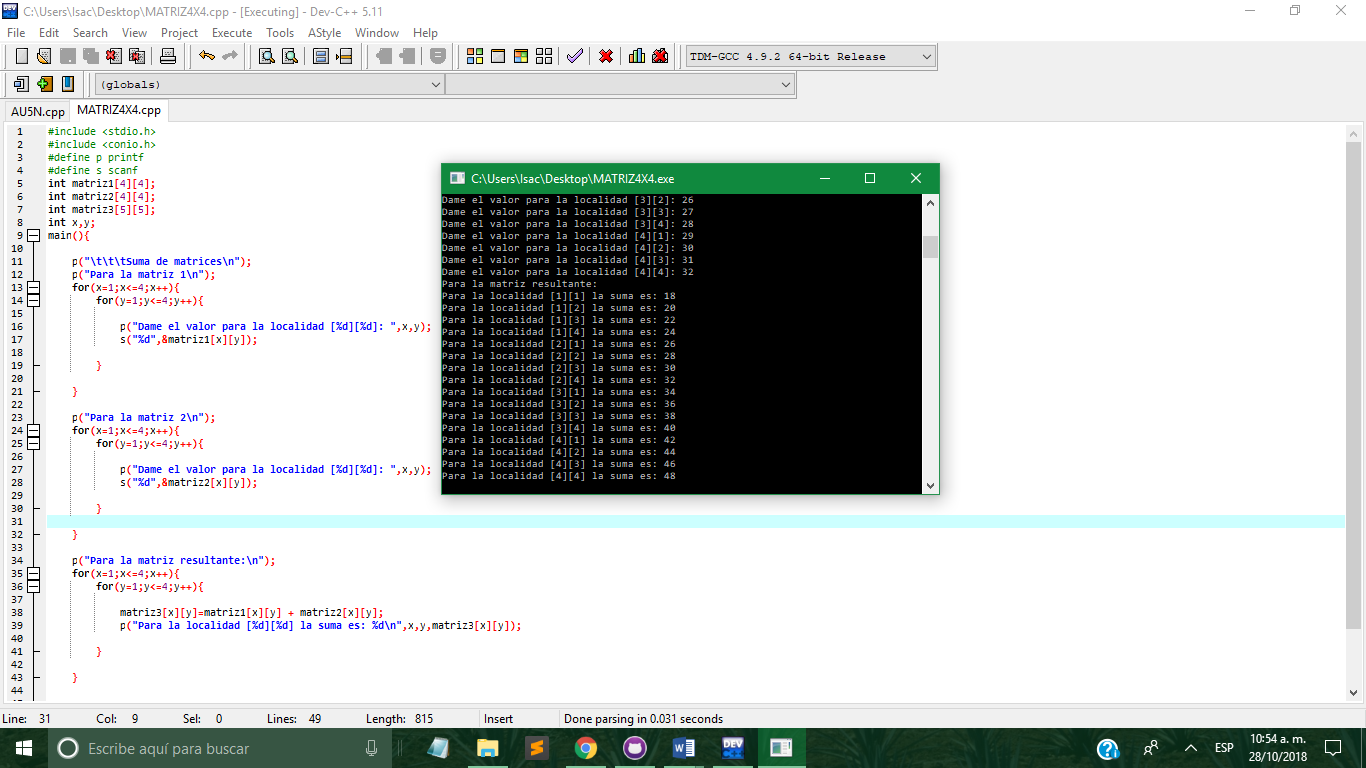
}

}

getch();

return 0;

}



Problema 3: Gastos semanales con Apuntadores

#include <stdio.h>

#include <conio.h>

const char \*dias[10]={"Lunes","Martes","Miercoles","Jueves","Viernes","Sabado","Domingo"};

int gastos[10];

float prom;

float suma=0;

int x;

main(){

for(x=0;x<7;x++){

printf("Dame los gastos del %s:",dias[x]);

scanf("%d",&gastos[x]);

suma=(suma+gastos[x]);

}

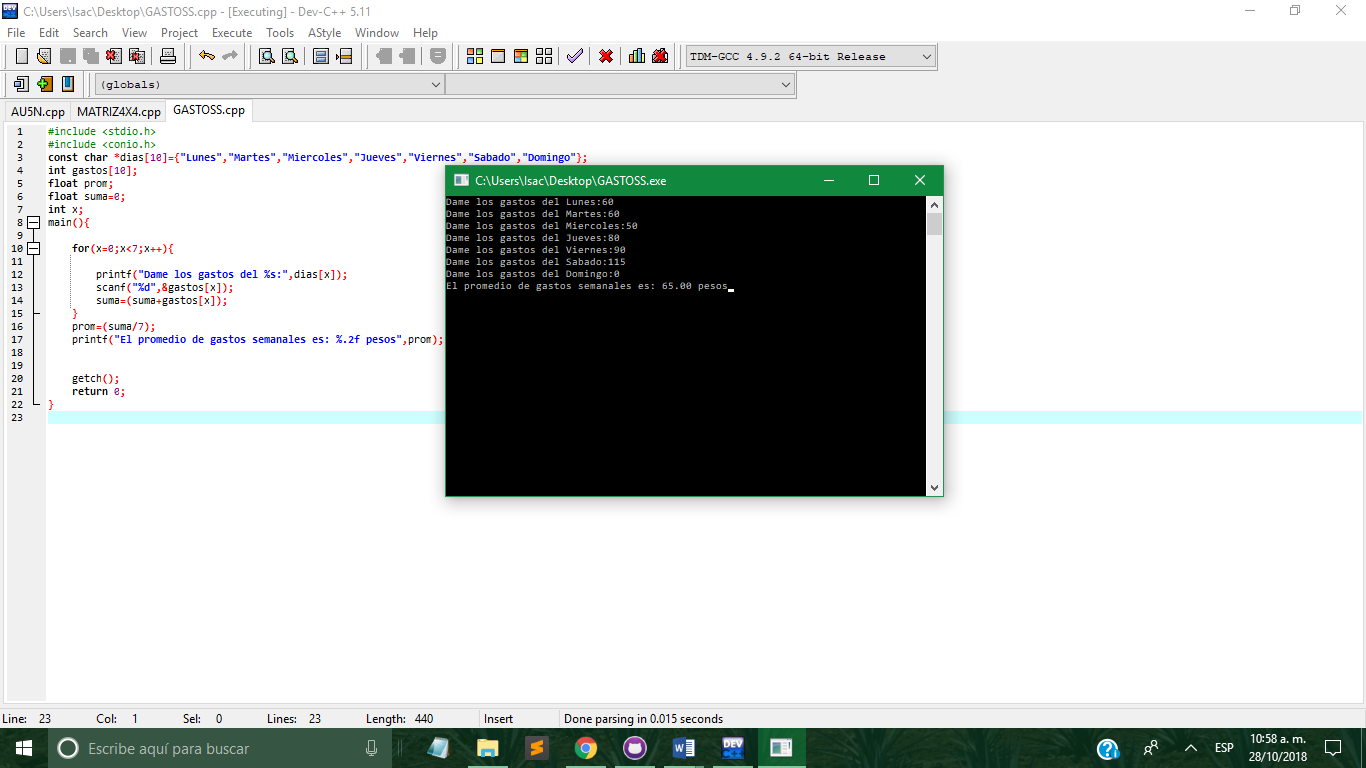
prom=(suma/7);

printf("El promedio de gastos semanales es: %f",prom);

getch();

return 0;

}



**CONCLUSIÓN**

Utilizar arreglos nos permite reducir y simplificar nuestro código puesto que al implementar el arreglo estamos agrupando un conjunto de variables que comparten características entre sí en una sola, esto nos permite aumentar el rendimiento al no tener declaradas demasiadas variables, sino que en una sola se encuentran las que necesitamos y de igual forma podemos manipularlas accediendo al arreglo. Al igual que cualquier otra variable, los arreglos se declaran con un tipo de dato, pero también se define su tamaño lo que hace referencia al “conjunto de variables que podrá almacenar”. Existen dos tipos de arreglo, el primero también conocido como vector, se refiere a un arreglo unidimensional, es decir, va en un solo sentido de forma lineal, un ejemplo arre1[5], donde en un solo renglón de memoria se guardan nuestras variables. Este tipo de arreglo nos sirve por ejemplo cuando tenemos varios datos que almacenar y que pertenecen a un mismo conjunto (gastos por mes de un Año), simplemente entonces a cada posición del arreglo se le asignan los valores en orden de lo solicitado. Por otra parte, están los arreglos bidimensionales también conocidos como matrices los cuales tienen utilidades un poco mayores en distintas áreas de la ingeniería, su característica principal es que están formados por renglones y columnas, formando una especie de red de datos almacenados en uno solo conjunto. Se utilizan por ejemplo cuando tenemos que almacenar varios datos que si pertenecen a un mismo conjunto pero no al mismo tiempo, es decir, como en el ejemplo de los gastos mensuales por año, si solo quisiéramos los gastos de un año en específico bastaría con un arreglo unidimensional pero si queremos los gastos de varios años, sería más conveniente utilizar un arreglo bidimensional, donde al final de cuentas se vuelve a guardar todo en una sola variable que simplemente mediante un doble ciclo for podemos recorrer y obtener los datos requeridos. Finalmente, en esta práctica vimos otra ventaja que tienen los apuntadores al trabajar en conjunto con arreglos, aprendí que podemos trabajar un arreglo como si fuera un apuntador y acceder a cada posición y valor del arreglo simplemente utilizando aritmética de apuntadores que sería recorrerlo con un ciclo, y estos a su vez relacionarlos con otro vector, aunque hayan estado declarados independientemente. En lo personal lo más nuevo para mí y realmente útil es el hecho de trabajar un apuntador como un arreglo y poder relacionarlo con otro de forma simultánea, aquí el detalle es que el arreglo apuntador funciona como un arreglo asociativo donde cada posición está relacionada con un valor que incluso puede ser una cadena, y a su vez, en base a su posición numérica tener un valor numérico relacionado perteneciente a otro arreglo tal y como vimos en el problema 3 de los gastos semanales con apuntadores.