|  |  |  |
| --- | --- | --- |
|  | Arreglos Unidimensionales | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

Salas A y B

|  |  |
| --- | --- |
| *Profesor:* | *Karina Garcia Morales* |
| *Asignatura:* | *Fundamentos de programación.* |
| *Grupo:* | *20* |
| *No de Práctica(s):* | *9* |
| *Integrante(s):* | *1* |
| *No. de Equipo de cómputo empleado:* | *Sin equipo* |
| *No. de Lista o Brigada:* |  |
| *Semestre:* | *1er semestre* |
| *Fecha de entrega:* | 16 de noviembre del 2021 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

TÍTULO: Arreglos Unidimensionales

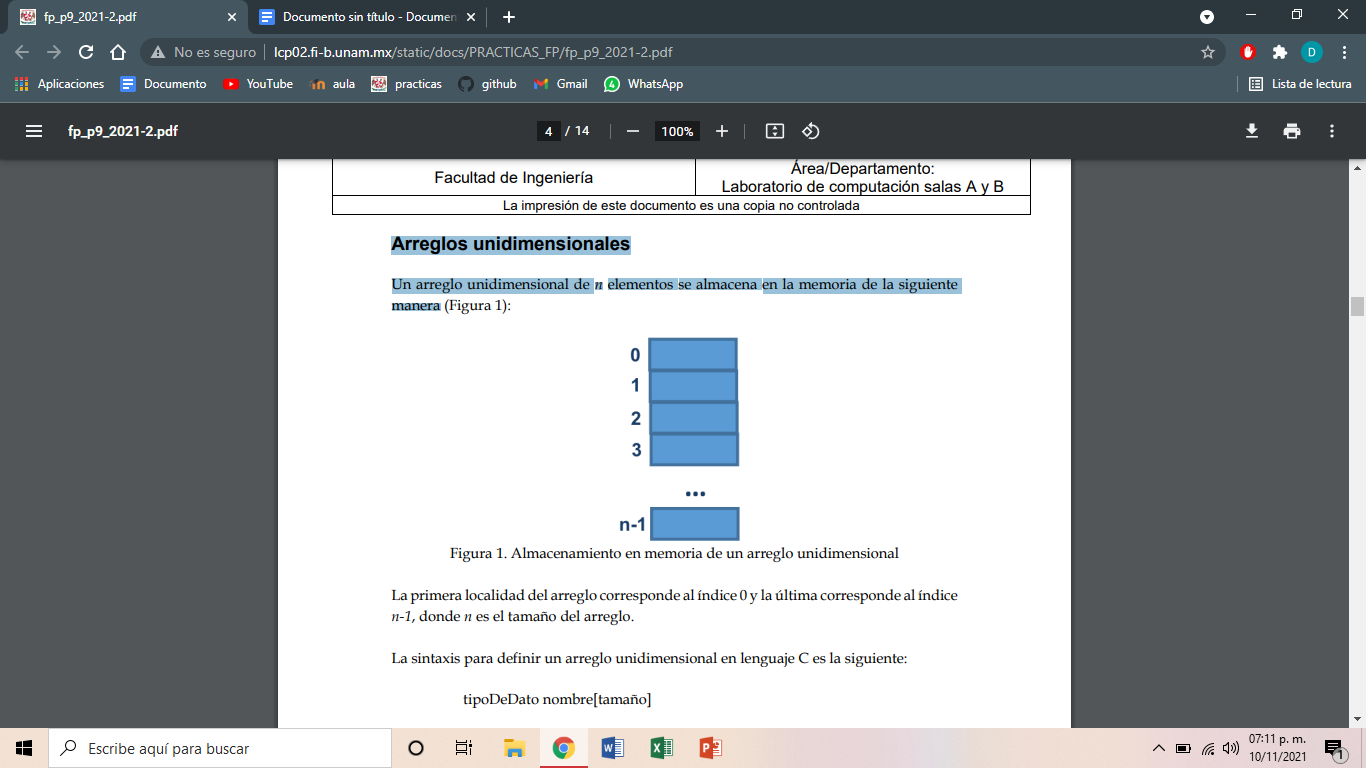
OBJETIVOS: El alumno utilizará arreglos de una dimensión en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, alineados en un vector o lista.

DESARROLLO:

Empezamos viendo los arreglos unidimensionales

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

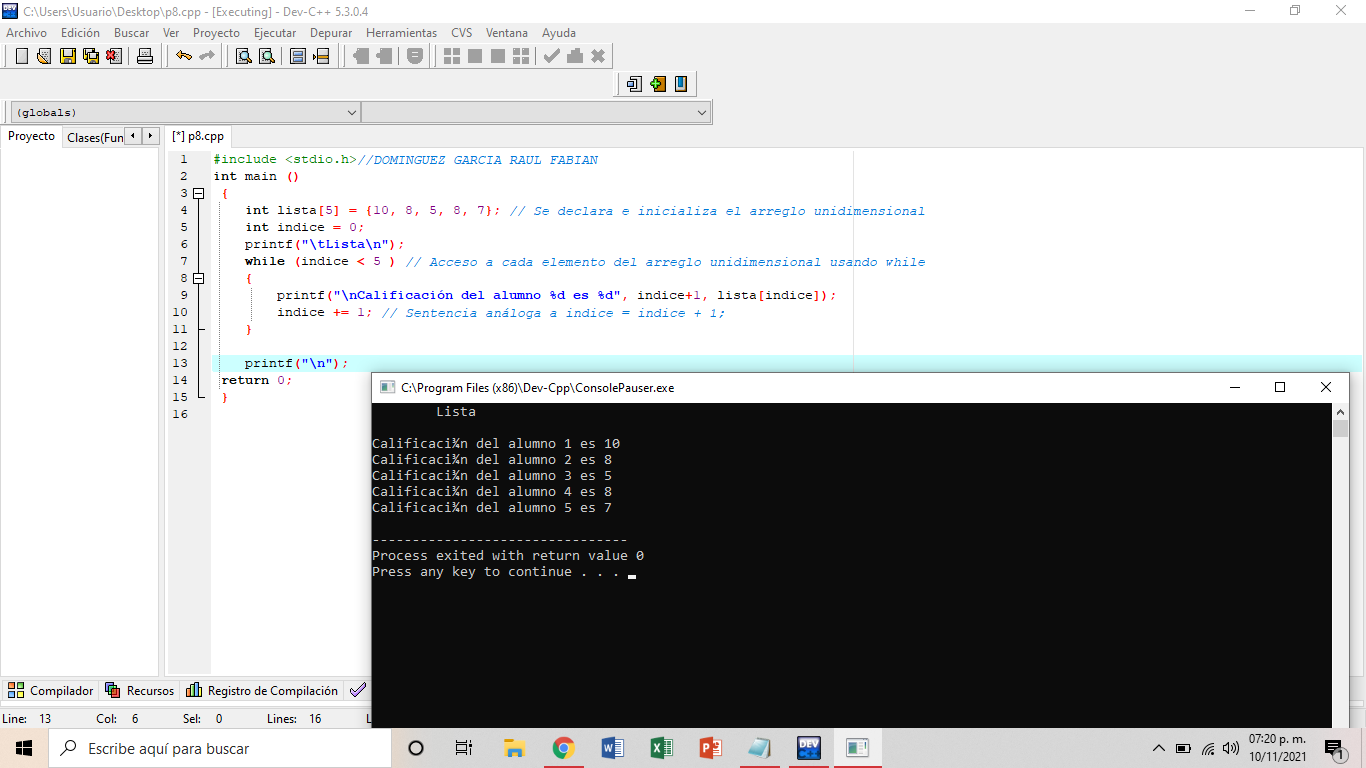
manera



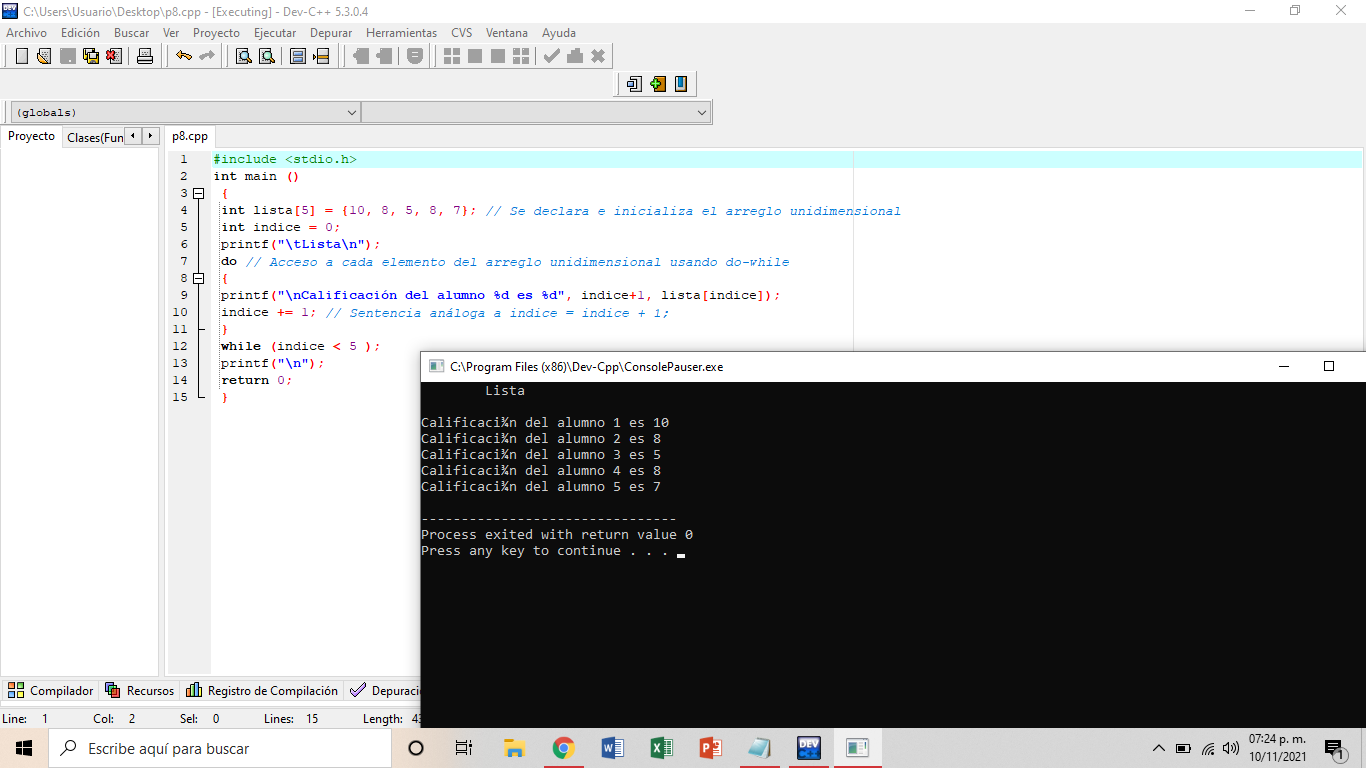
La primera localidad del arreglo corresponde al índice 0 y la última corresponde al índice n-1, donde n es el tamaño del arreglo. La sintaxis para definir un arreglo unidimensional en lenguaje C es la siguiente: tipoDeDato nombre[tamaño] Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo. El tipoDeDato es el tipo de dato de los elementos del arreglo, el cual puede ser tipo entero, real, carácter o estructura.

Luego empezamos a ver Código (arreglo unidimensional empleando la estructura while) y ejecutamos el programa que venía de ejemplo

Este código se genera un arreglo unidimensional de 5 elementos y que, para poder acceder, recorrer y mostrar cada elemento del arreglo se usa la variable índice desde 0 hasta 4

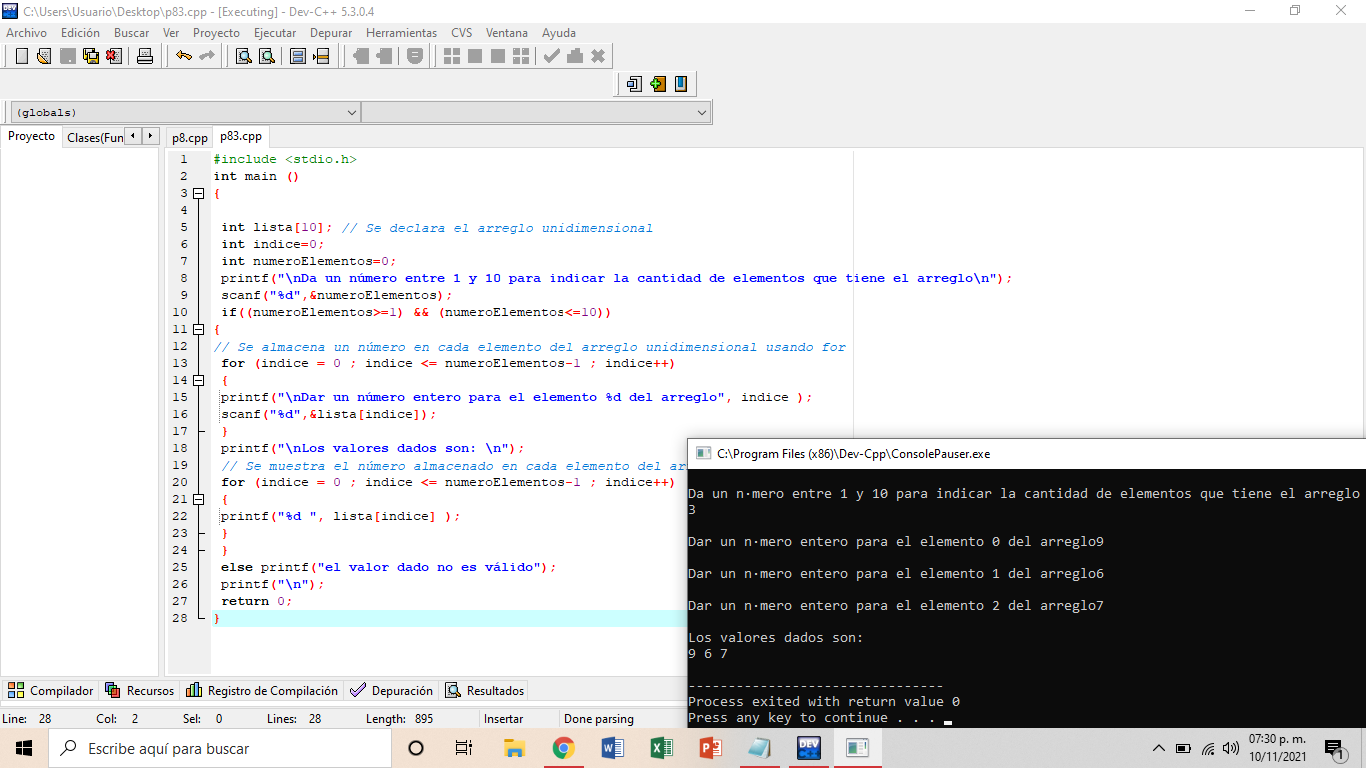


seguimos con el Código (arreglo unidimensional empleando la estructura do-while) y lo ejecutamos es el mismos solo cambian que este es do while es un código se genera un arreglo unidimensional de 5 elementos y que para poder acceder, recorrer y mostrar cada elemento del arreglo se usa la variable índice desde 0 hasta 4

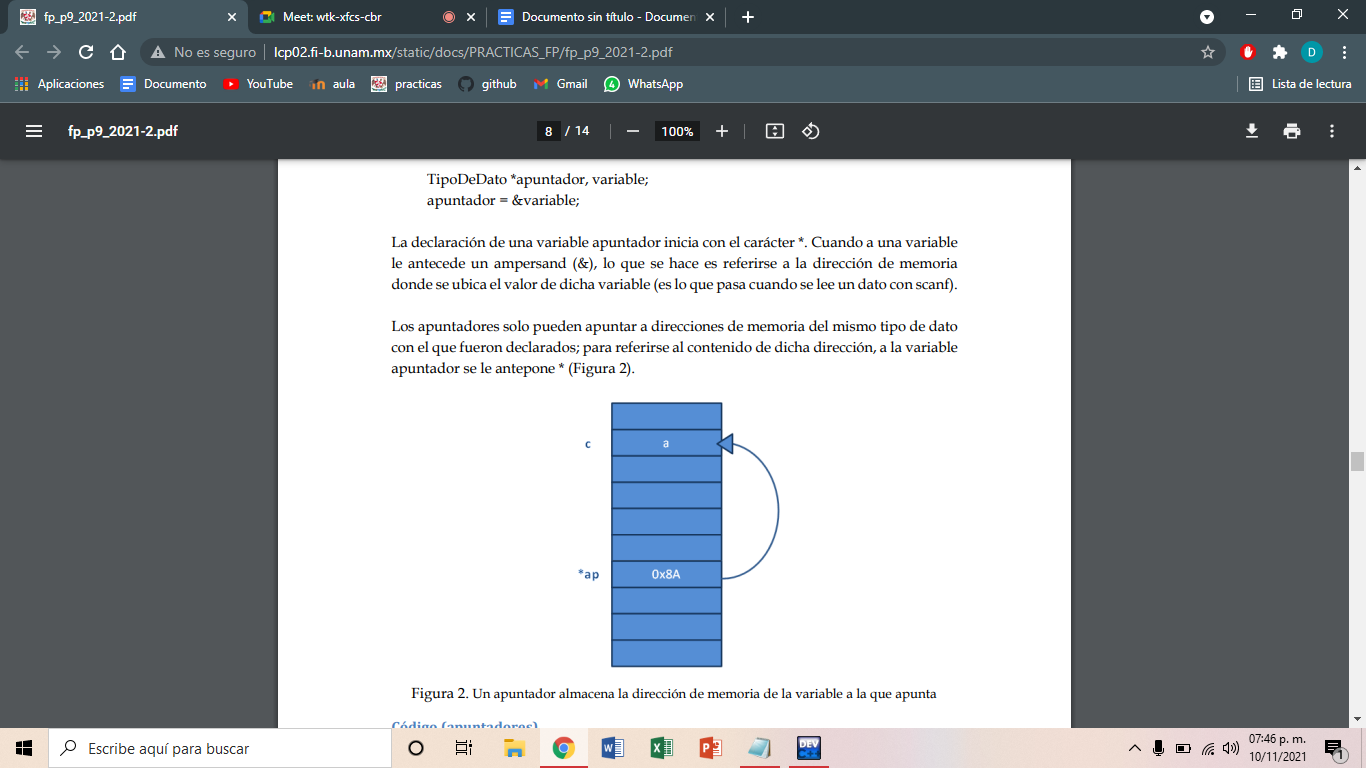


ahora vamos con el código (arreglo unidimensional empleando la estructura for)

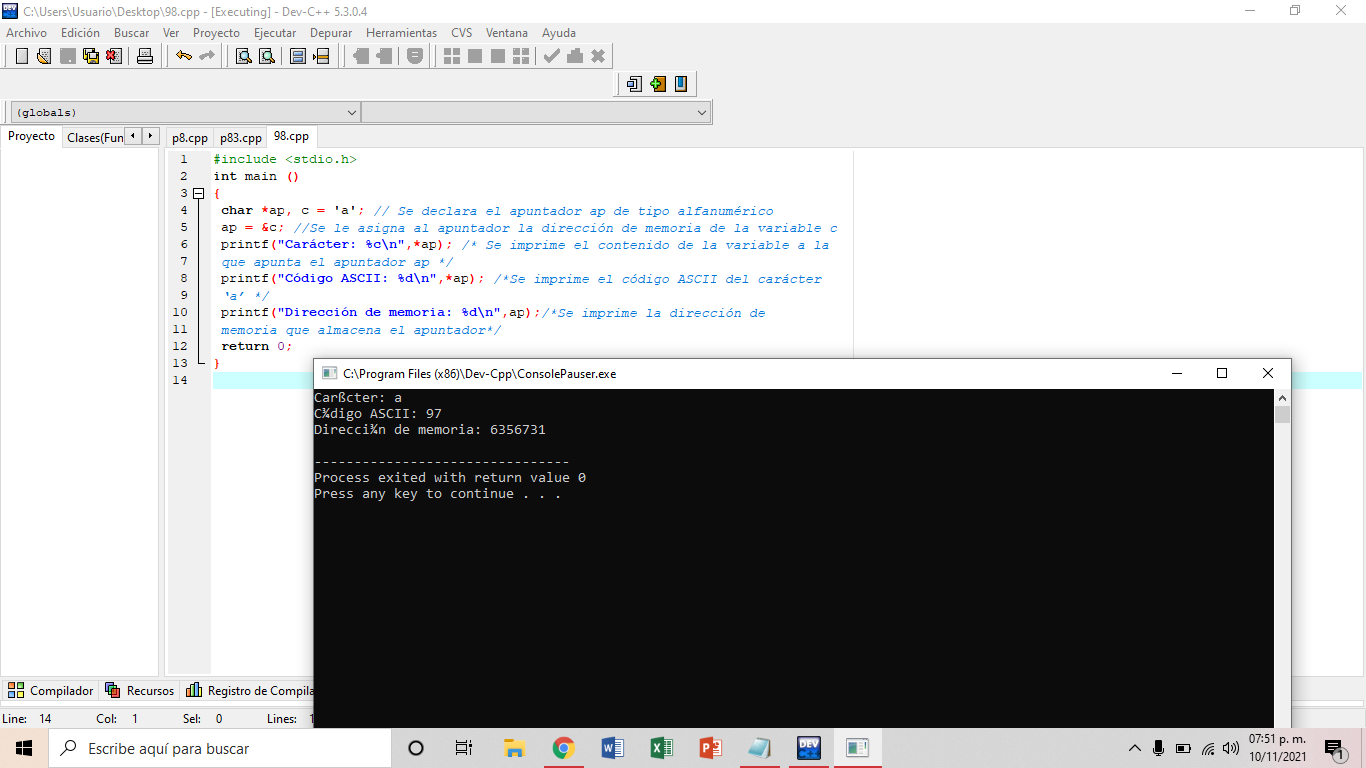
A continuación, se muestra un programa que genera un arreglo unidimensional de máximo 10 elementos. Para poder leer y almacenar datos en cada elemento y posteriormente mostrar el contenido de estos elementos se hace uso de un ciclo for respectivamente y lo ejecutamos



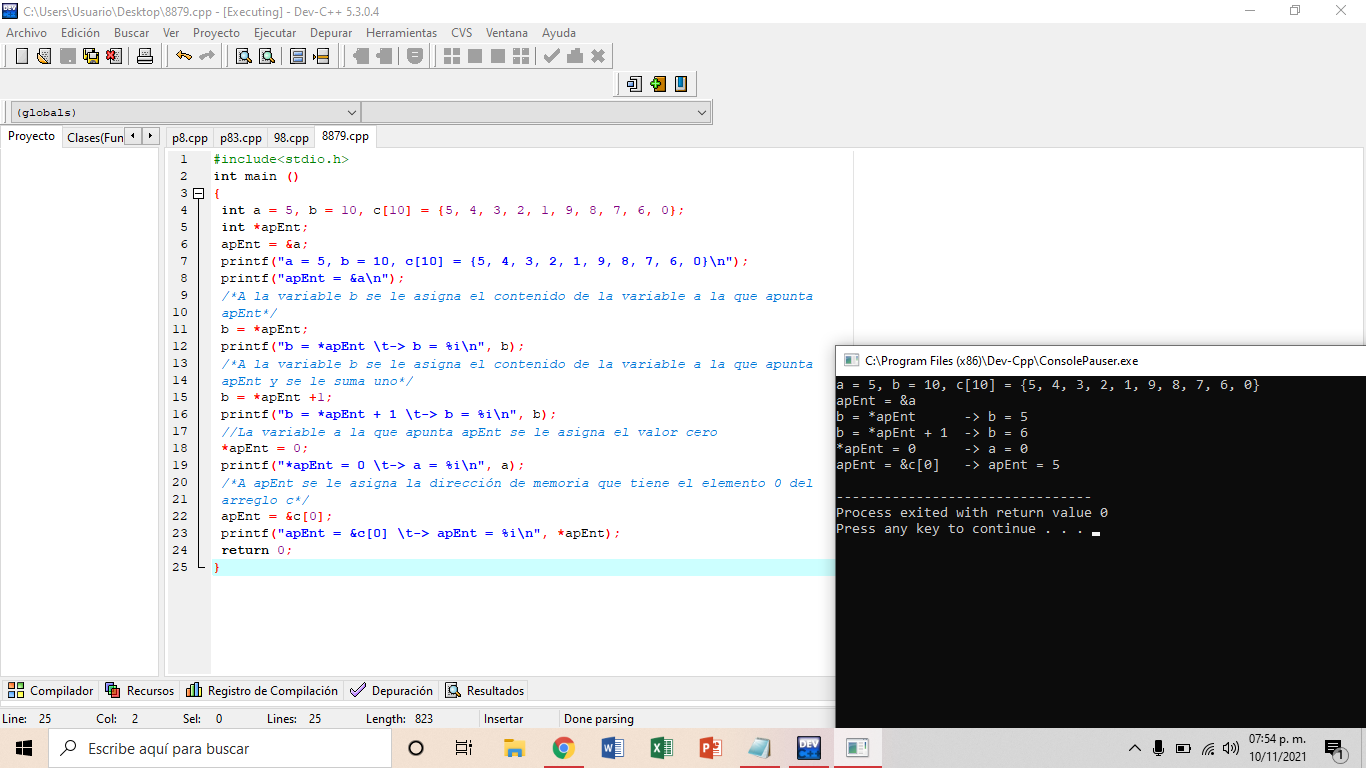
Luego seguimos con Apuntadores: un apuntador es una variable que contiene la dirección de una variable, es decir, hace referencia a la localidad de memoria de otra variable. Debido a que los apuntadores trabajan directamente con la memoria, a través de ellos se accede con rapidez a un dato.



y ejecutamos el problema que siguiente se aprecia la declaración de un apuntador de tipo carácter y se muestra en pantalla, haciendo uso de apuntadores, el contenido de la variable c, así como el código ASCII del carácter ‘a’ que tiene almacenado dicha variable



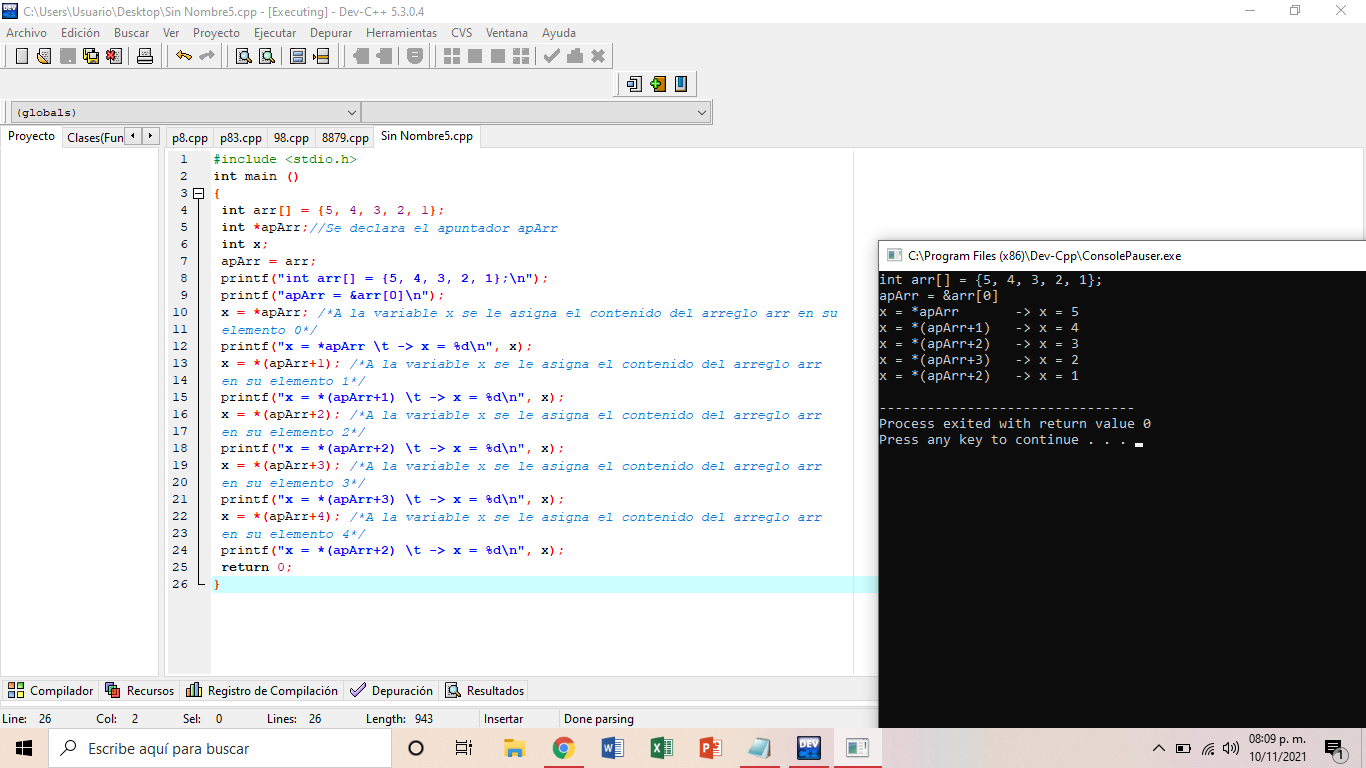
y seguimos con el siguiente ejercicio permite acceder a las localidades de memoria de distintas variables a través de un apuntador



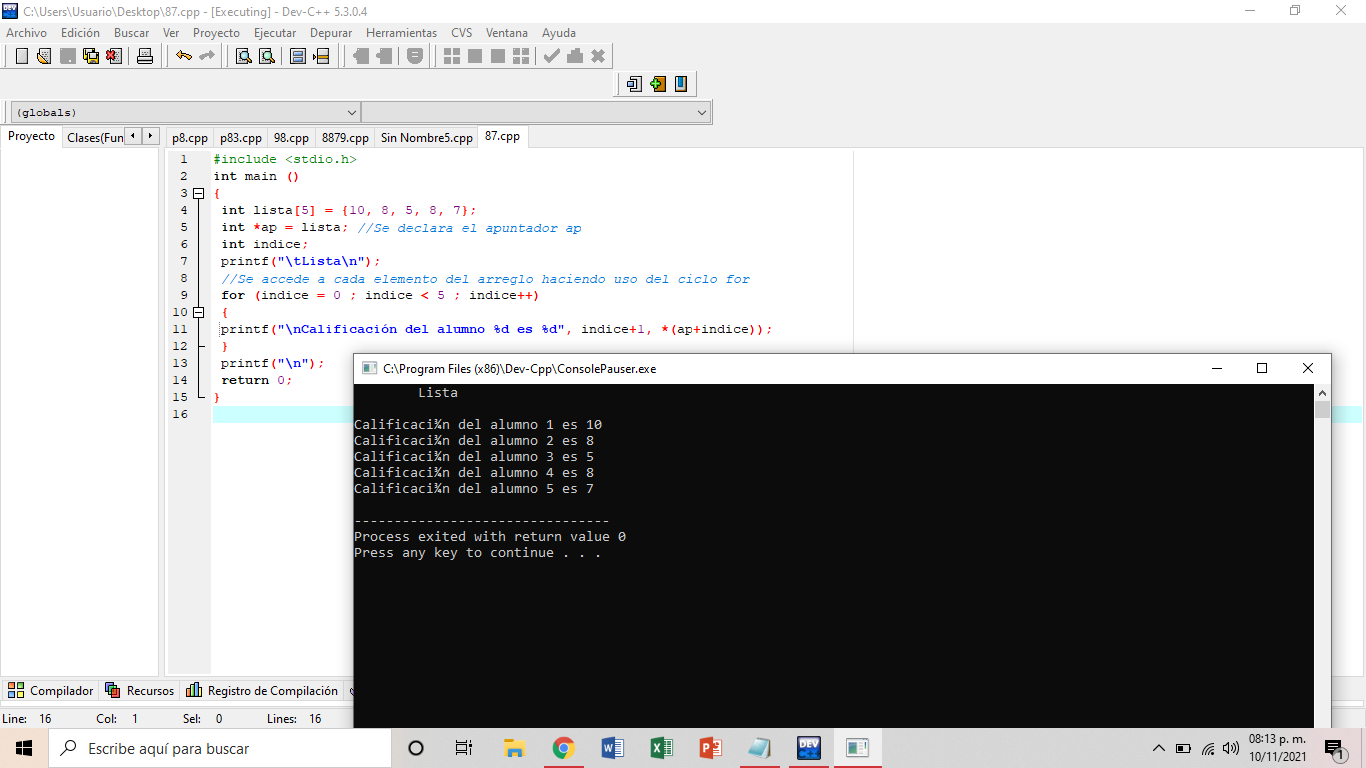
y lo modificamos un poco

cap

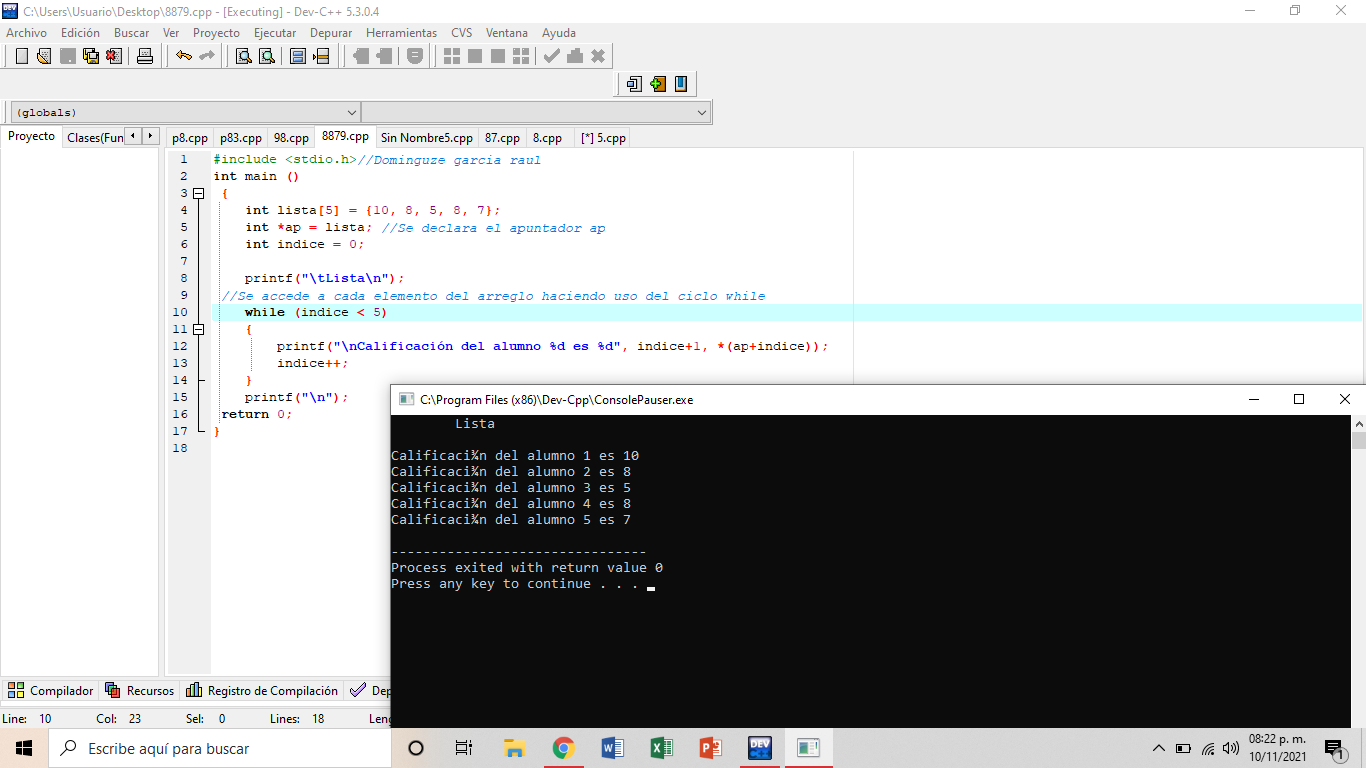
y seguimos con el siguiente problema, el programa que se observa a continuación trabaja con aritmética de apuntadores para acceder a todos los valores que se encuentran almacenados en cada uno de los elementos de un arreglo.



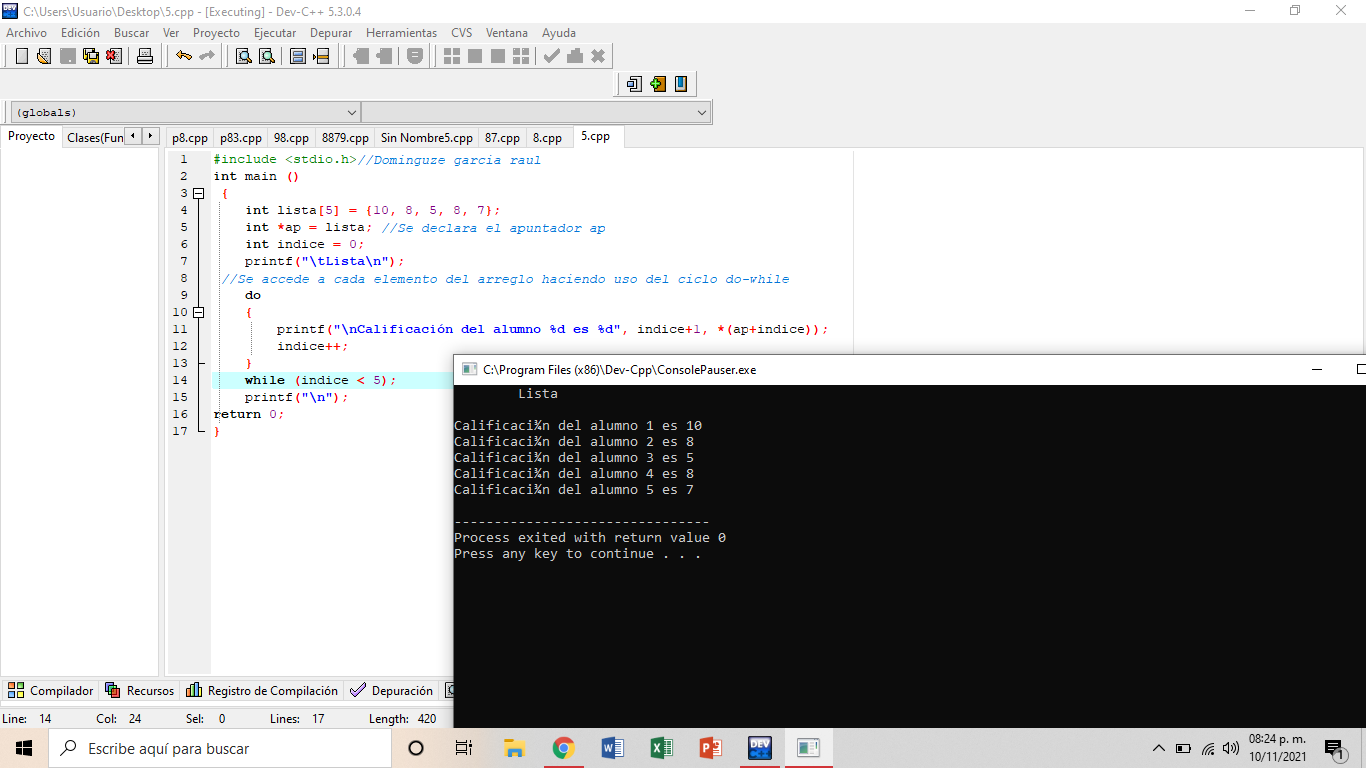
y seguimos con Código (apuntadores en ciclo for) y ejecutamos el programa que sigue el programa genera un arreglo unidimensional de 5 elementos y accede a cada uno de los elementos del arreglo haciendo uso de un apuntador, para ello se utiliza un ciclo for



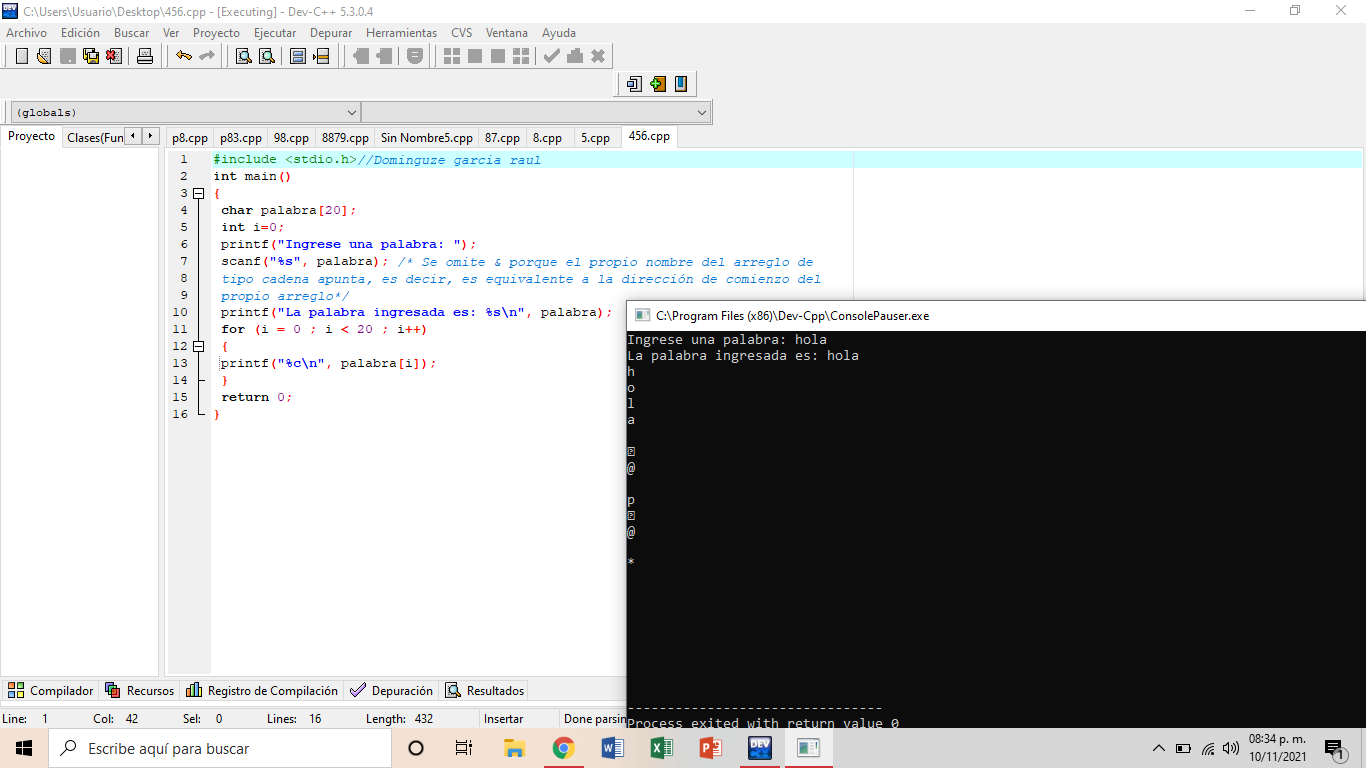
y el mismo programa, pero ahora while



en el siguiente programa usamos apuntadores en ciclo do-while



y seguimos con apuntadores en cadenas El siguiente programa muestra el manejo básico de cadenas en lenguaje C



EJERCICIOS DE TAREA:

1,-Indica que realiza el siguiente programa:

Nos muestra nos arreglos partiendo del elemento 0 hasta el 9 con un ciclo for y nos va enseñando el valor de cada arreglo desde 12 hasta 36 y que primero hace que i se reste y el siguiente for hace que este vuelva a sumar y nos muestra los números en los arreglos.

#include <stdio.h>

int arreglo[] = {20,19,12,17,16,36,14,13,23,31};

int i, j, n, aux;

main() {

n = 10;

for(i=1; i<n; i++) {

j = i;

aux = arreglo[i];

while(j>0 && aux<arreglo[j-1]) {

arreglo[j] = arreglo[j-1];

j=j-1;

}

arreglo[j] = aux;

}

printf("\n\nLos elementos obtenidos del arreglo son: \n");

for(i=0; i<n; i++) {

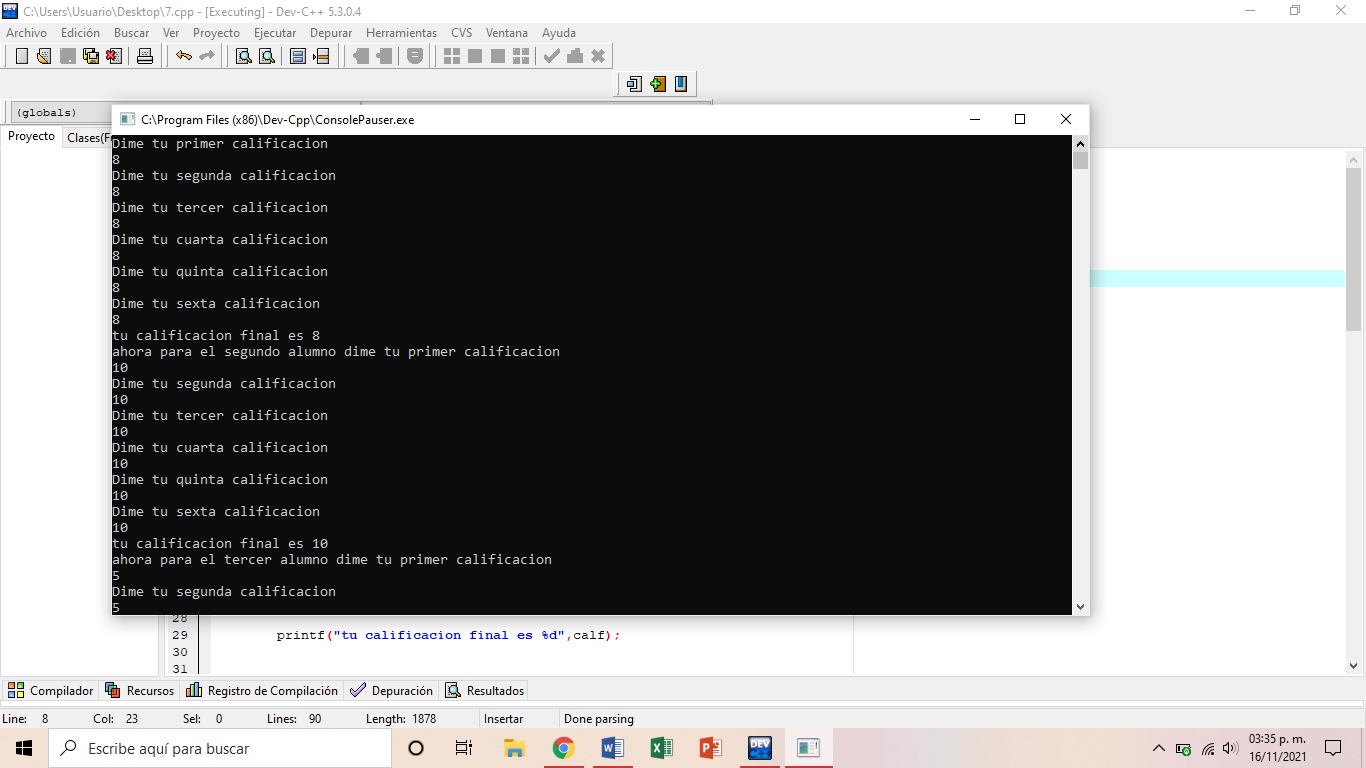
printf("Elemento [%d]: %d\n", i, arreglo[i]);

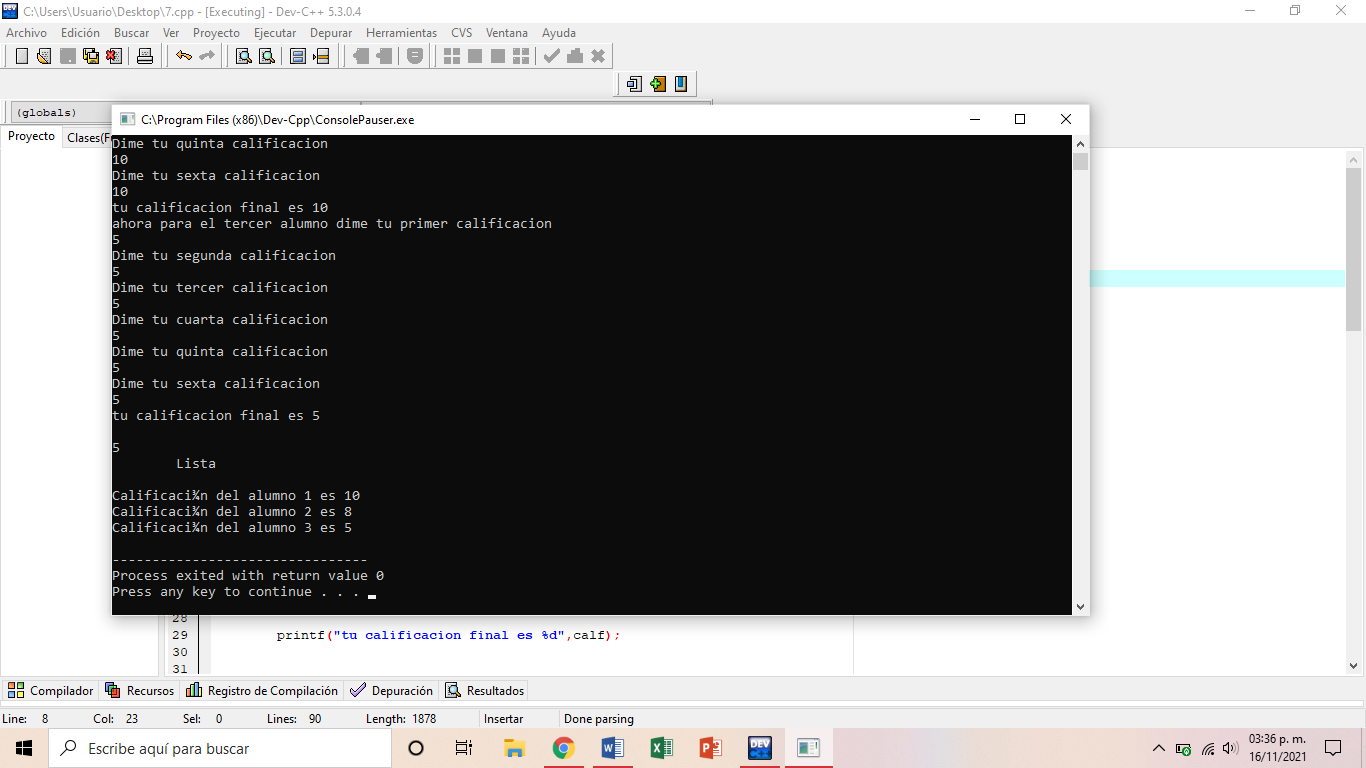
}

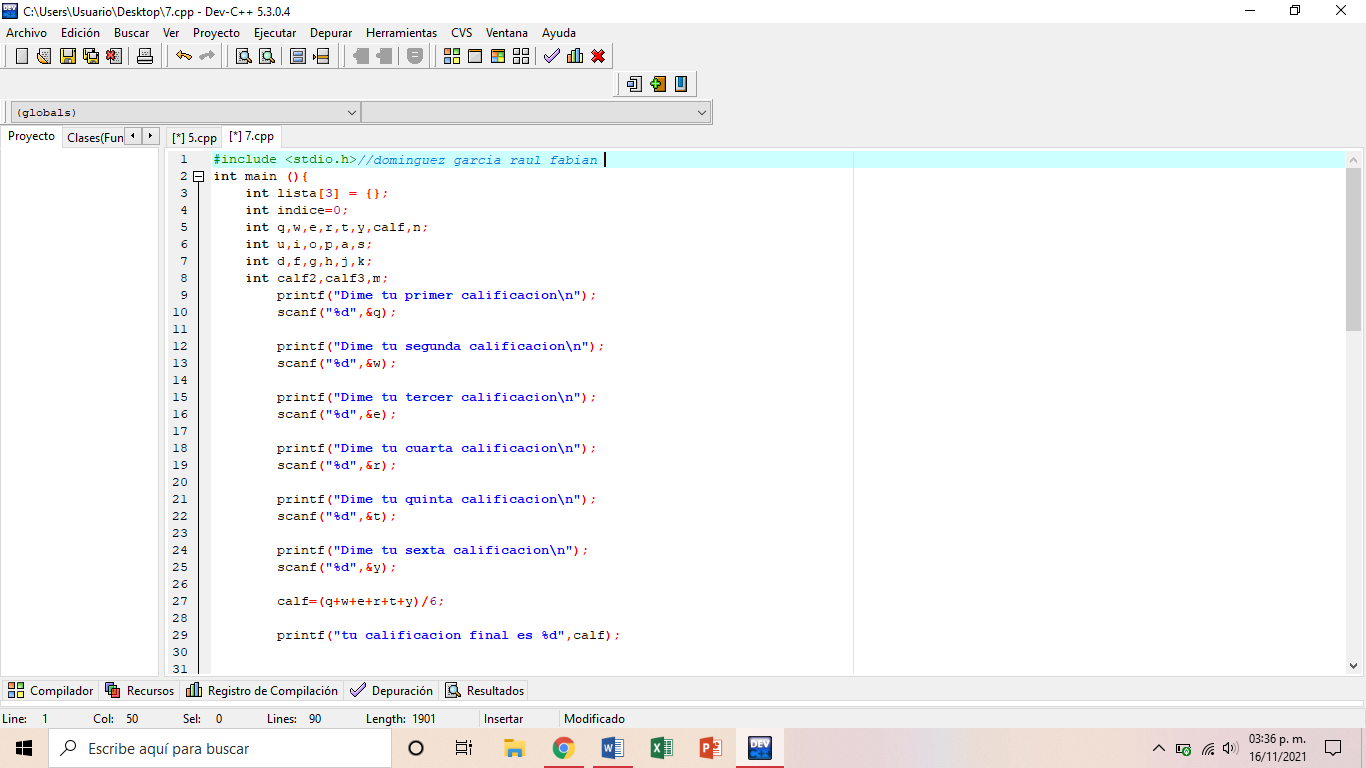
return 0;

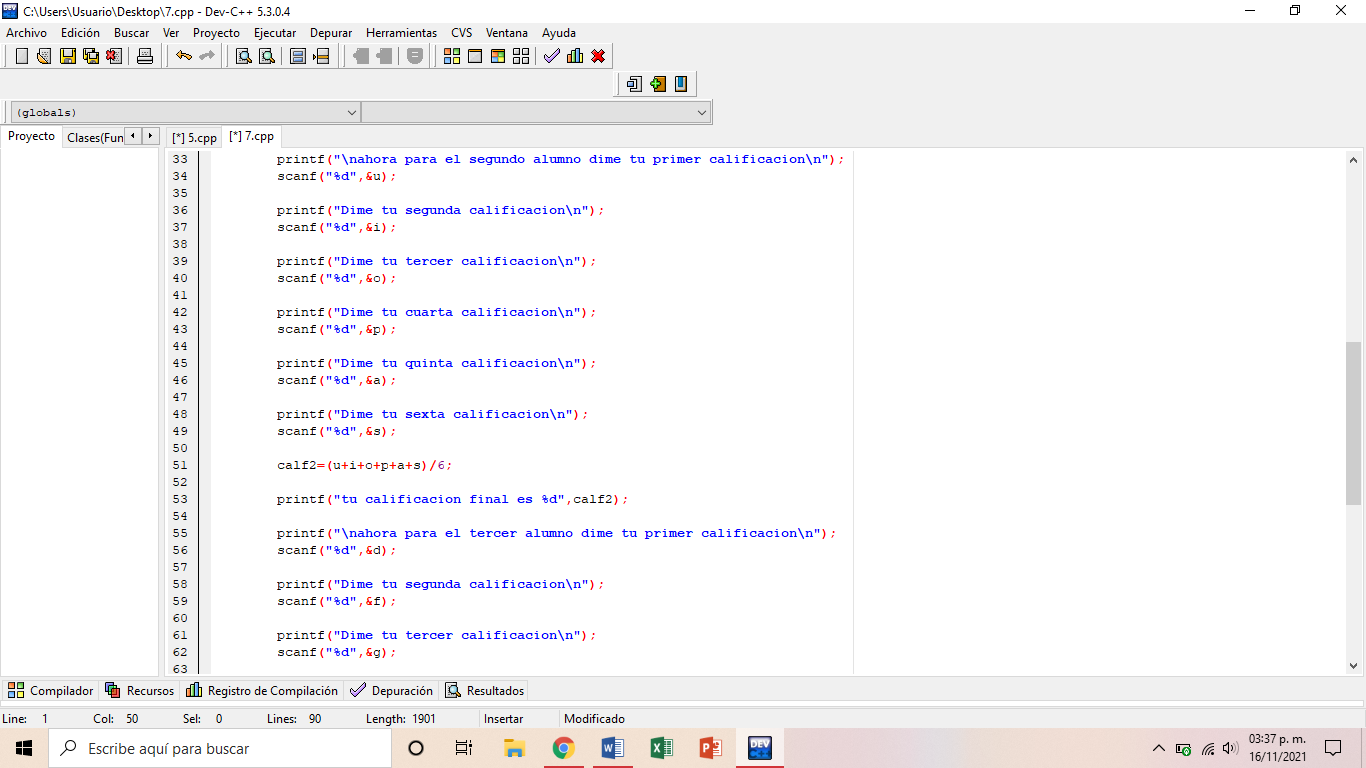
}

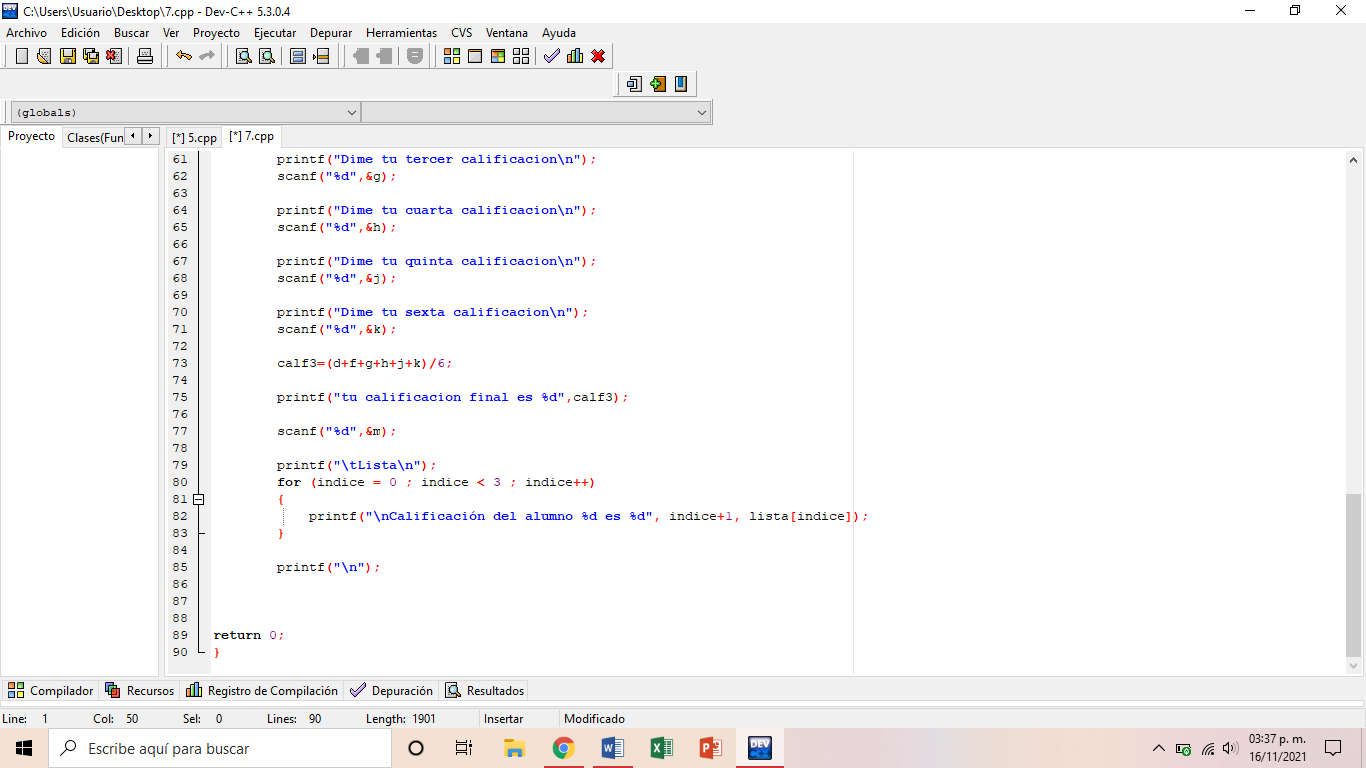
2.-Generar un programa que solicite a 3 usuarios su 6 calificación y obtenga los promedios de cada uno y los almacene en un arreglo, al finalizar mostrar los promedios el arreglo y el promedio de todos los alumnos (utilizar ciclos).











CONCLUSIONES:

En esta práctica logramos identificar y aplicar los arreglos, para poder realizar problemas y que nos sean más fáciles ya que hacen que los problemas sean más cortos y por esto más rápidos de realizar problemas, y logramos hacer ejercicios nosotros solos usando los arreglos,

BIBLIOGRAFÍA:

UNAM, (2015), “Laboratorio de Computación Salas A y B”<http://lcp02.fi-b.unam.mx/>