|  |  |  |
| --- | --- | --- |
|  | Arreglos Multidimensionales | |
| 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):* | *10* |
| *Integrante(s):* | *1* |
| *No. de Equipo de cómputo empleado:* | *Sin equipo* |
| *No. de Lista o Brigada:* |  |
| *Semestre:* | *1er semestre* |
| *Fecha de entrega:* | 23 de noviembre del 2021 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

TÍTULO: Arreglos Multidimensionales

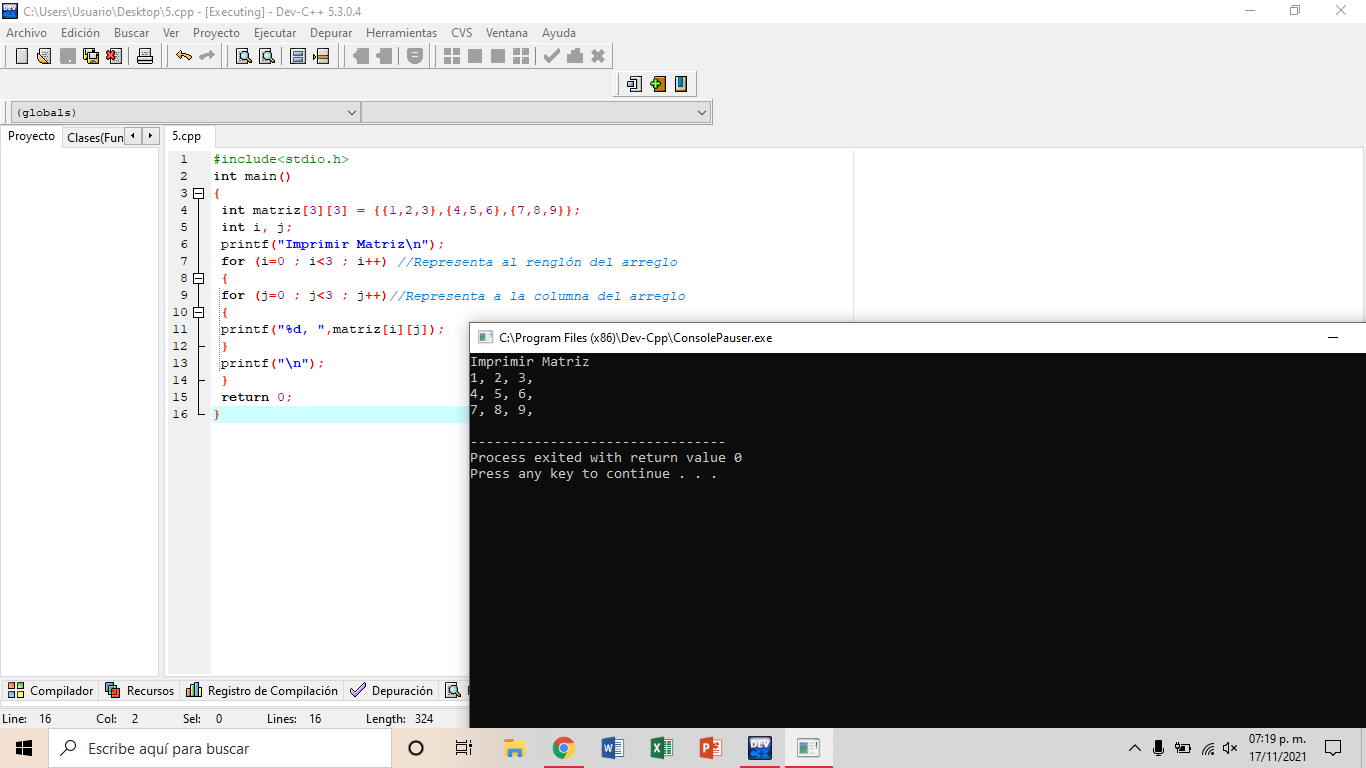
OBJETIVO: El alumno utilizará arreglos de dos dimensiones en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, en estructuras que utilizan dos índices.

DESARROLLO

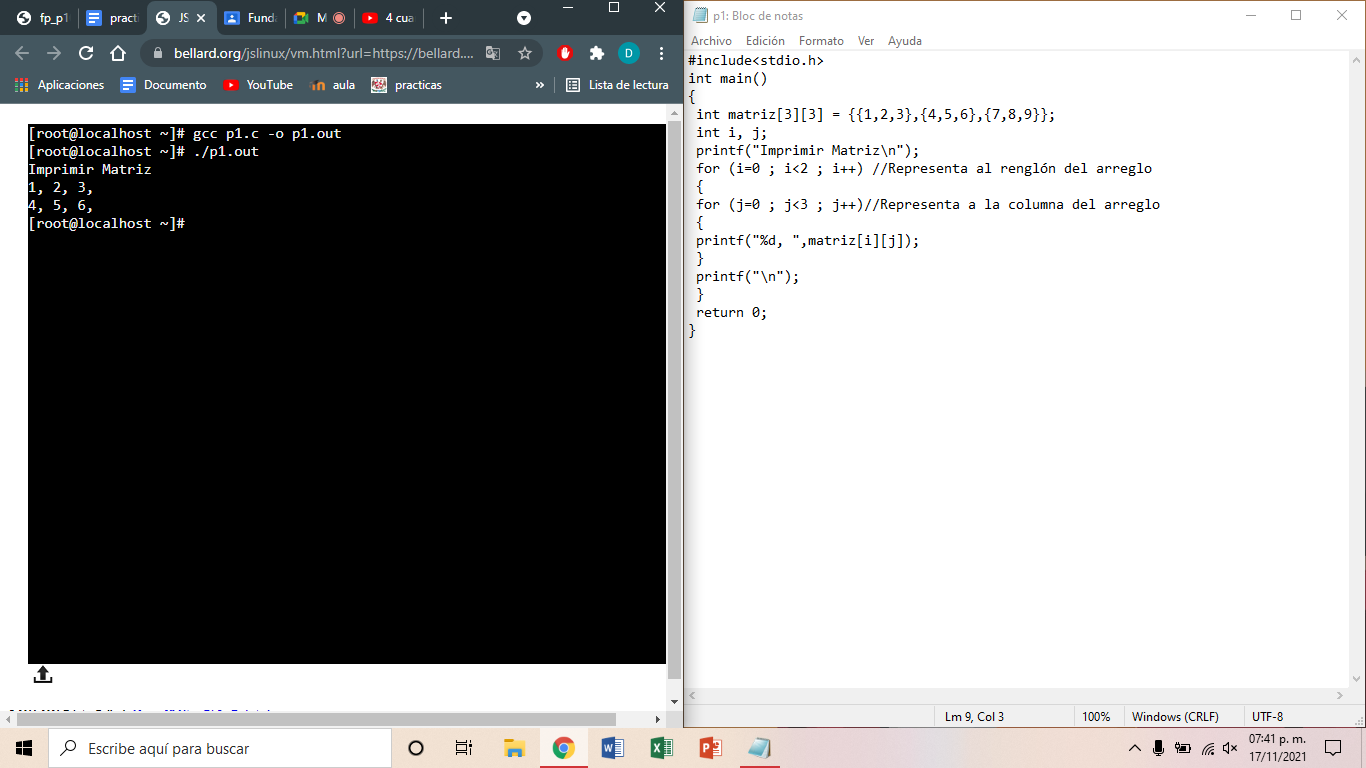
La maestra nos empezó explicando que un arreglo de dos dimensiones es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse. Para acceder a un elemento en este tipo de arreglos se requiere el uso de dos índices. Los arreglos se utilizan para hacer más eficiente el código de un programa, ya que la manipulación de datos del mismo tipo que son agrupados en un arreglo por tener un significado común, se realiza de una forma más clara y eficaz

empezamos ejecutando un programa

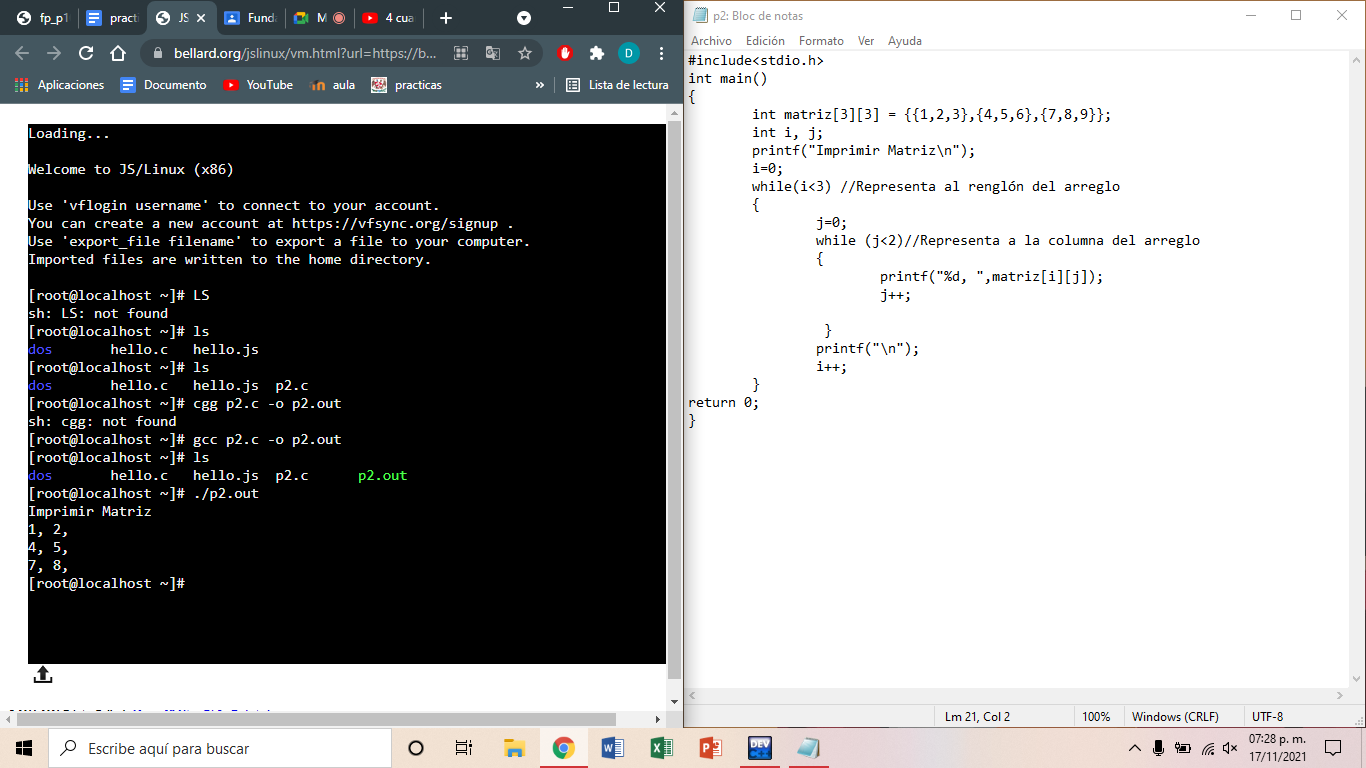
A continuación, se observa un programa que genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a cada uno de sus elementos por la posición que indica el renglón y la columna a través de dos ciclos for, uno anidado dentro de otro.



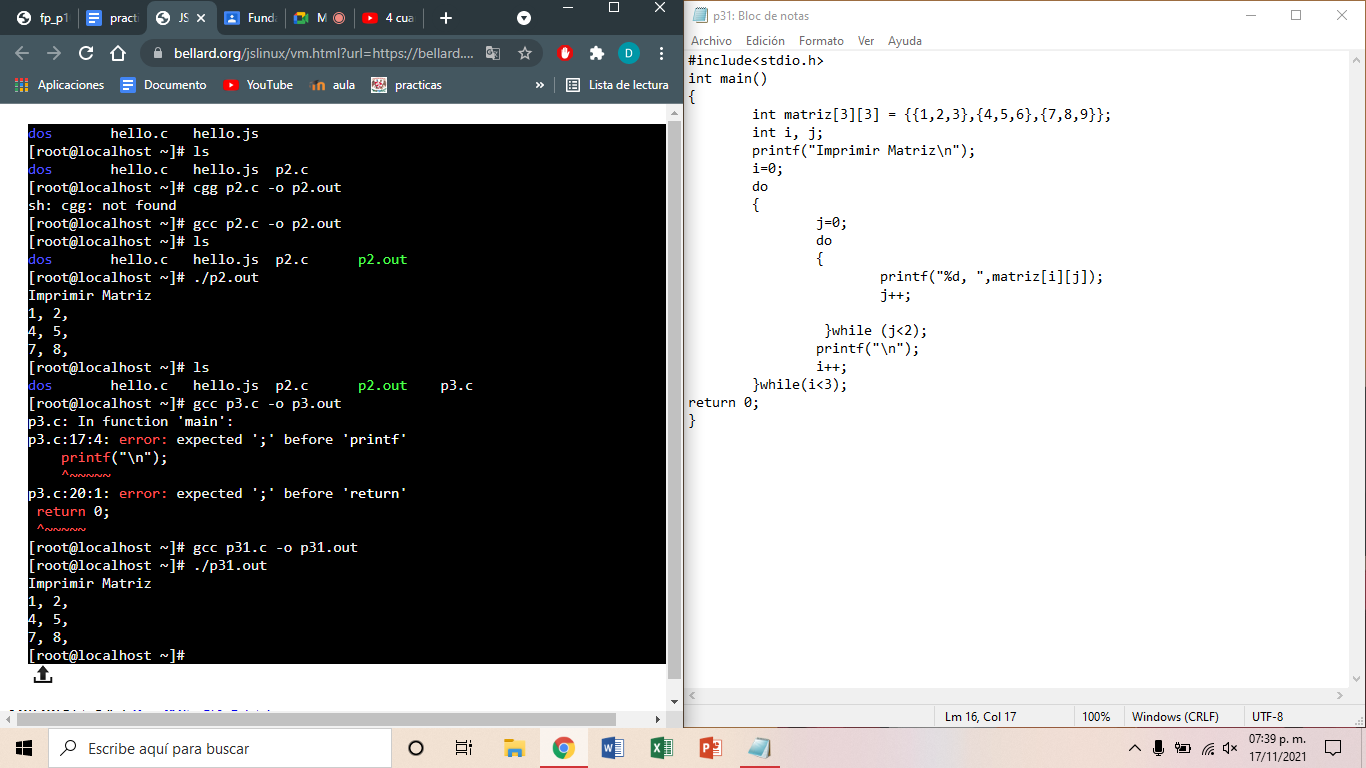
lo modificamos un poco para que en lugar de ser 3x3 sea 3x2



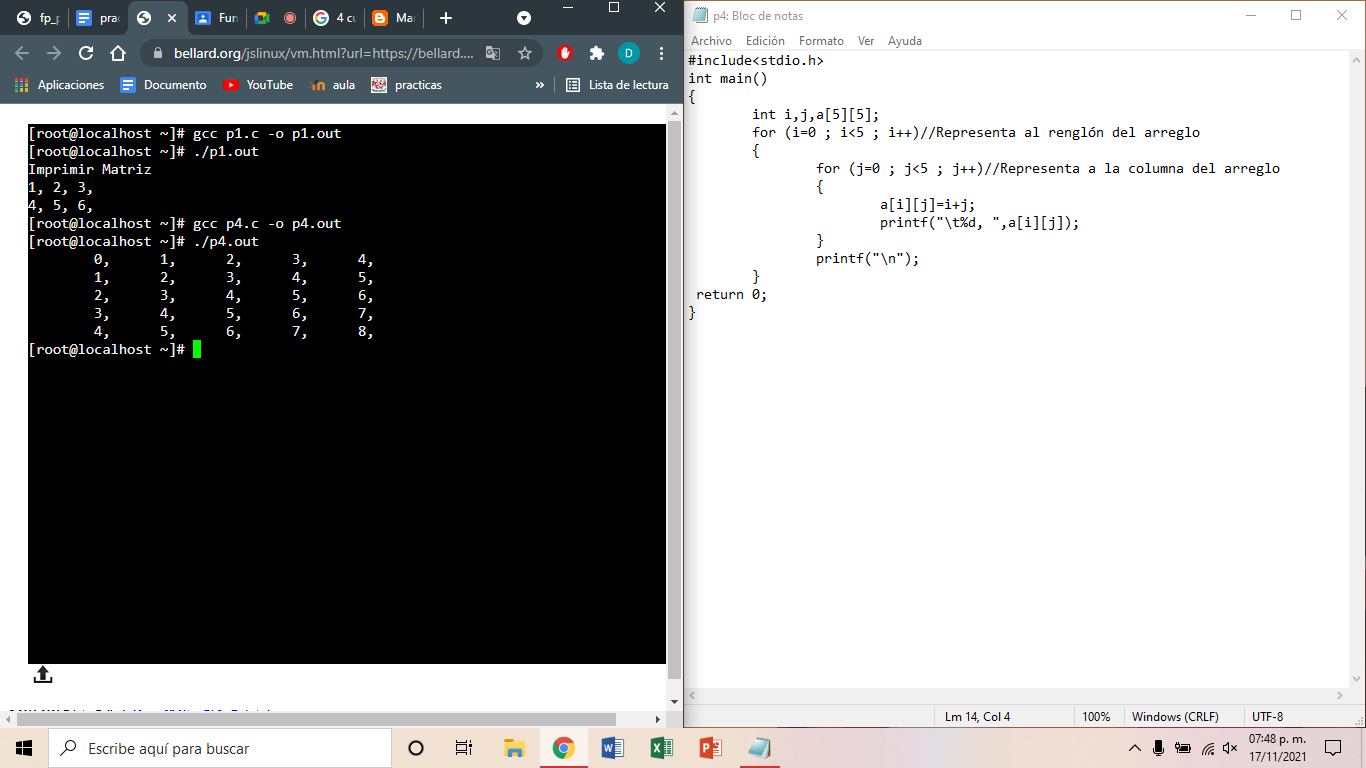
y lo modificamos por un ciclo while



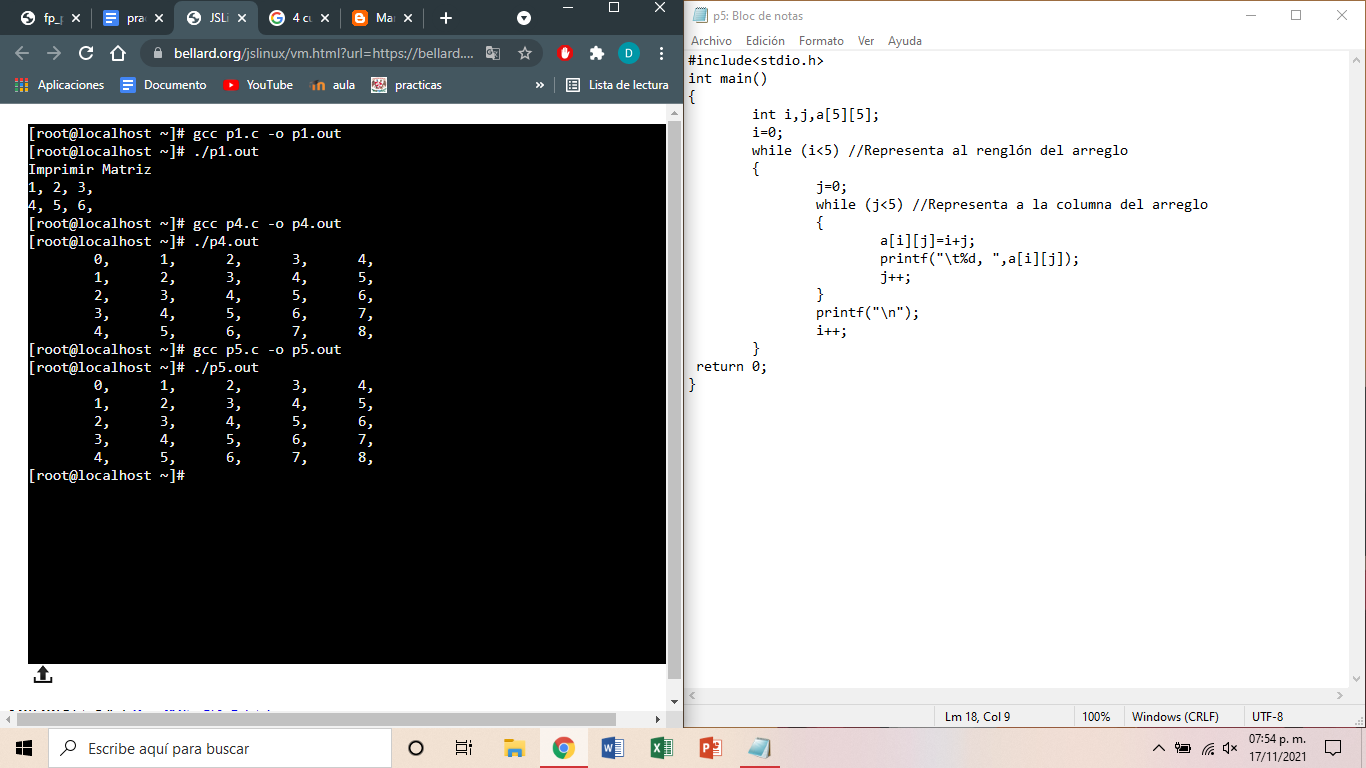
y ahora lo modificamos con un ciclo do-while



y seguimos con el 2do ejercicio que aparece en la práctica que ahora es de 5x5 El siguiente programa genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos for, uno anidado dentro de otro, el contenido de cada elemento de este arreglo es la suma de sus índices.

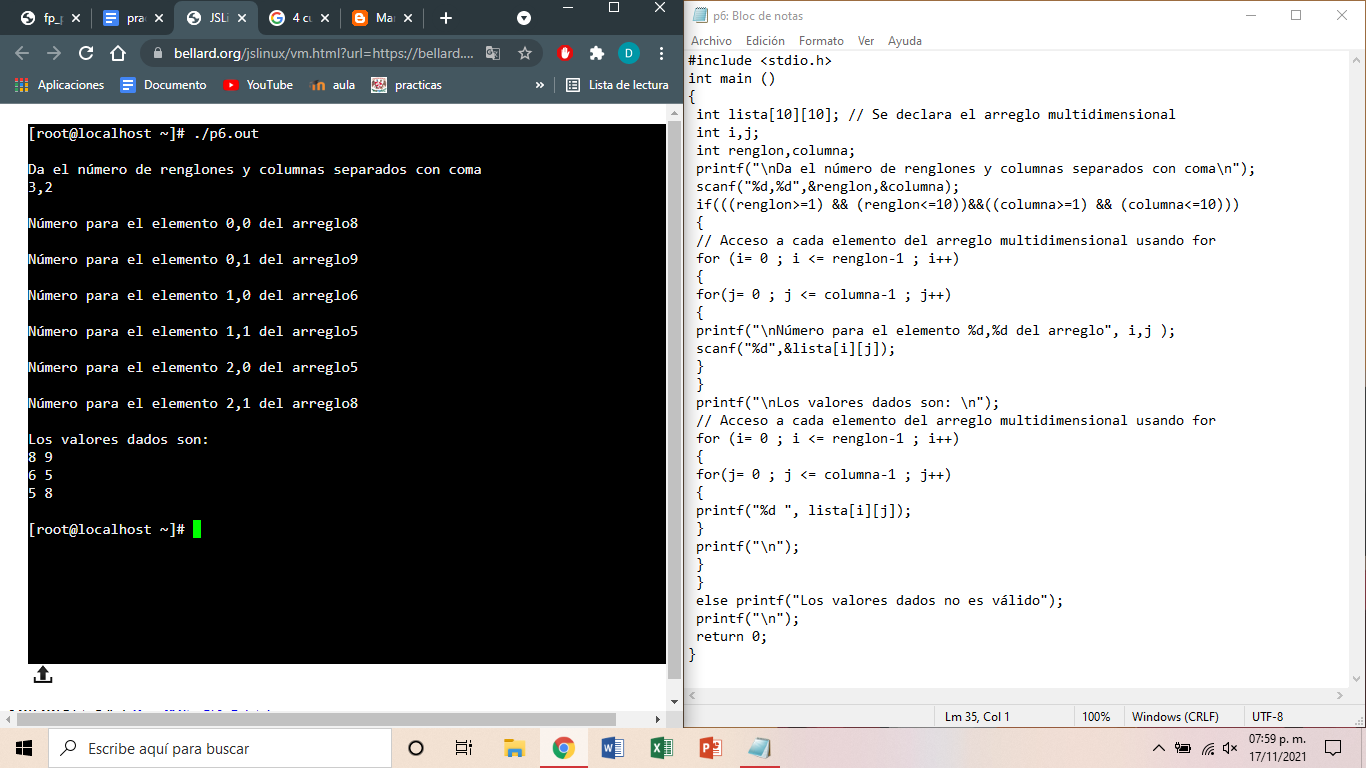


y ahora pasamos al siguiente que es con un ciclo while

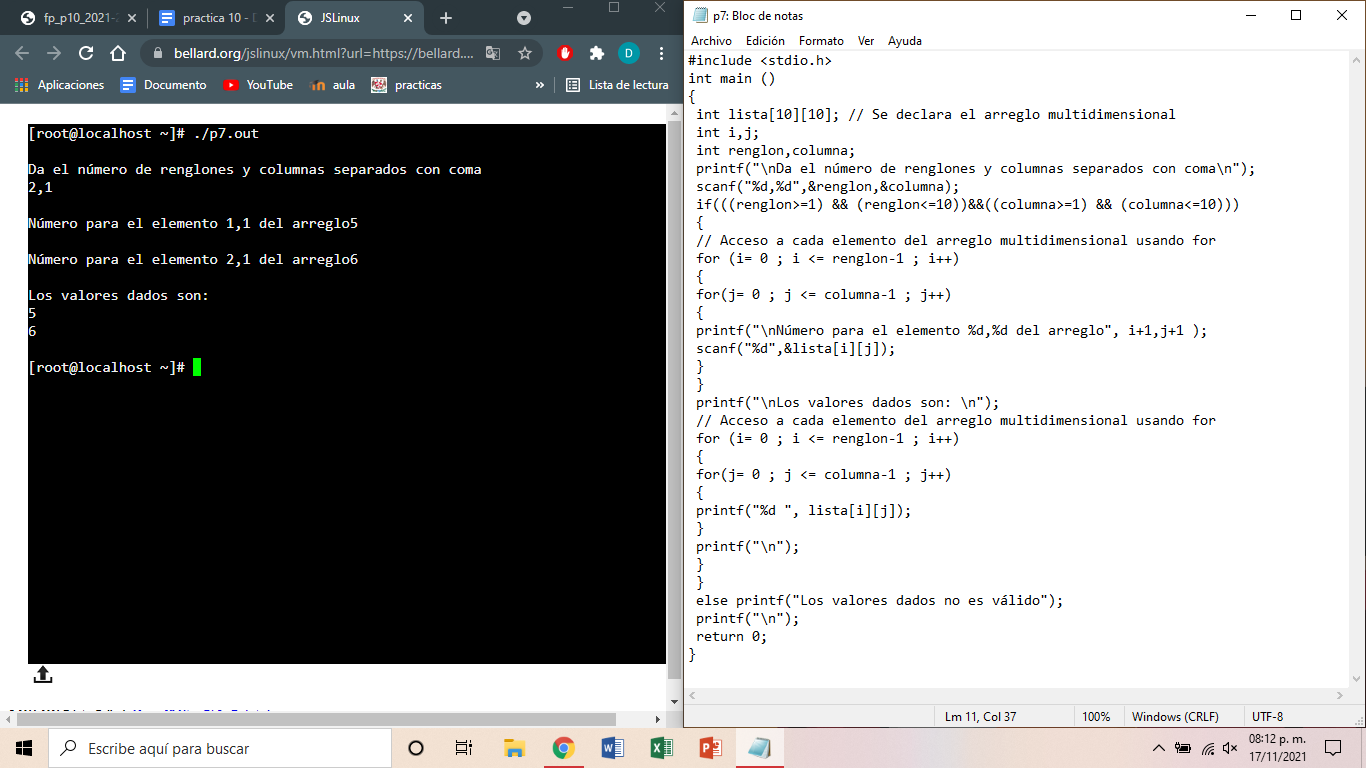


y de aquí vimos los demás ejercicios que son el mismo solo que con do while

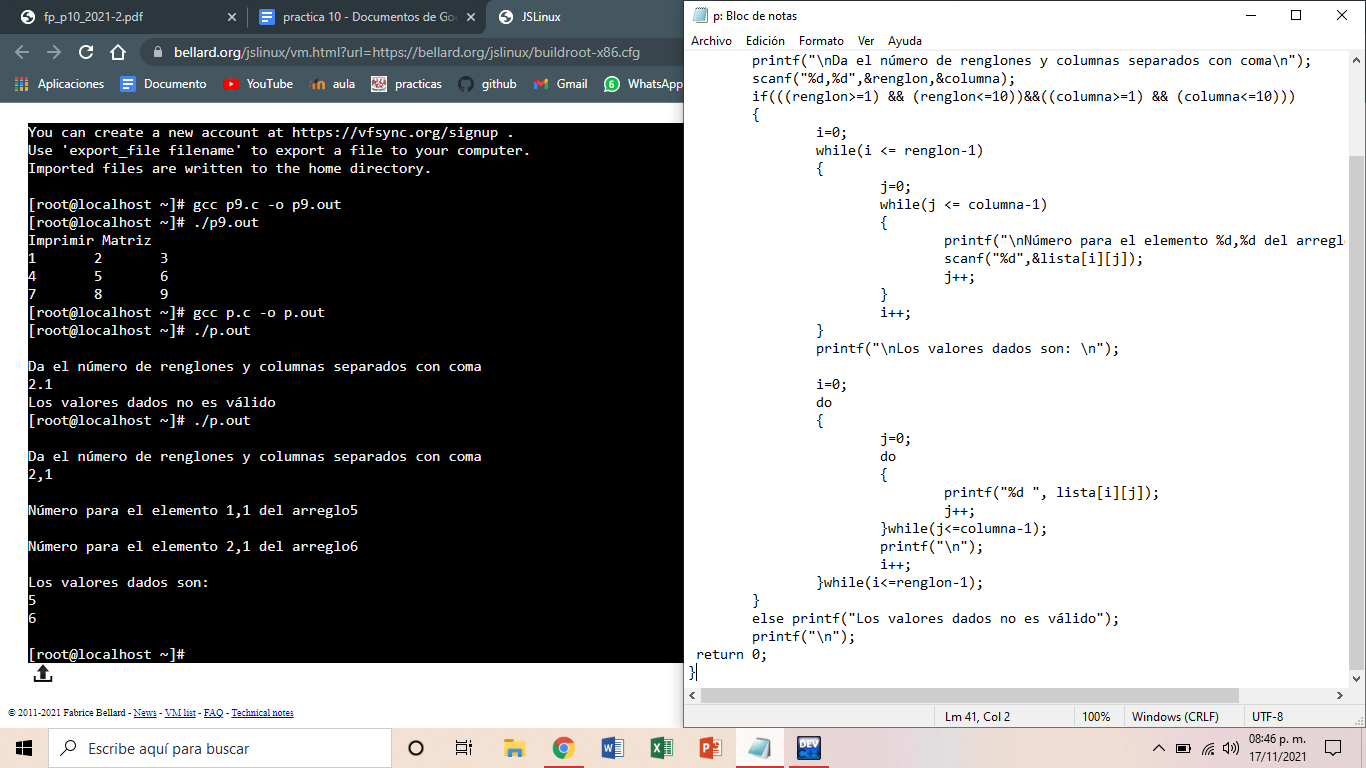
pasamos al siguiente ejercicio se muestra un programa que genera un arreglo multidimensional de máximo 10 renglones y 10 columnas, para poder almacenar datos en cada elemento y posteriormente mostrar el contenido de esos elementos se hace uso de ciclos for anidados respectivamente.



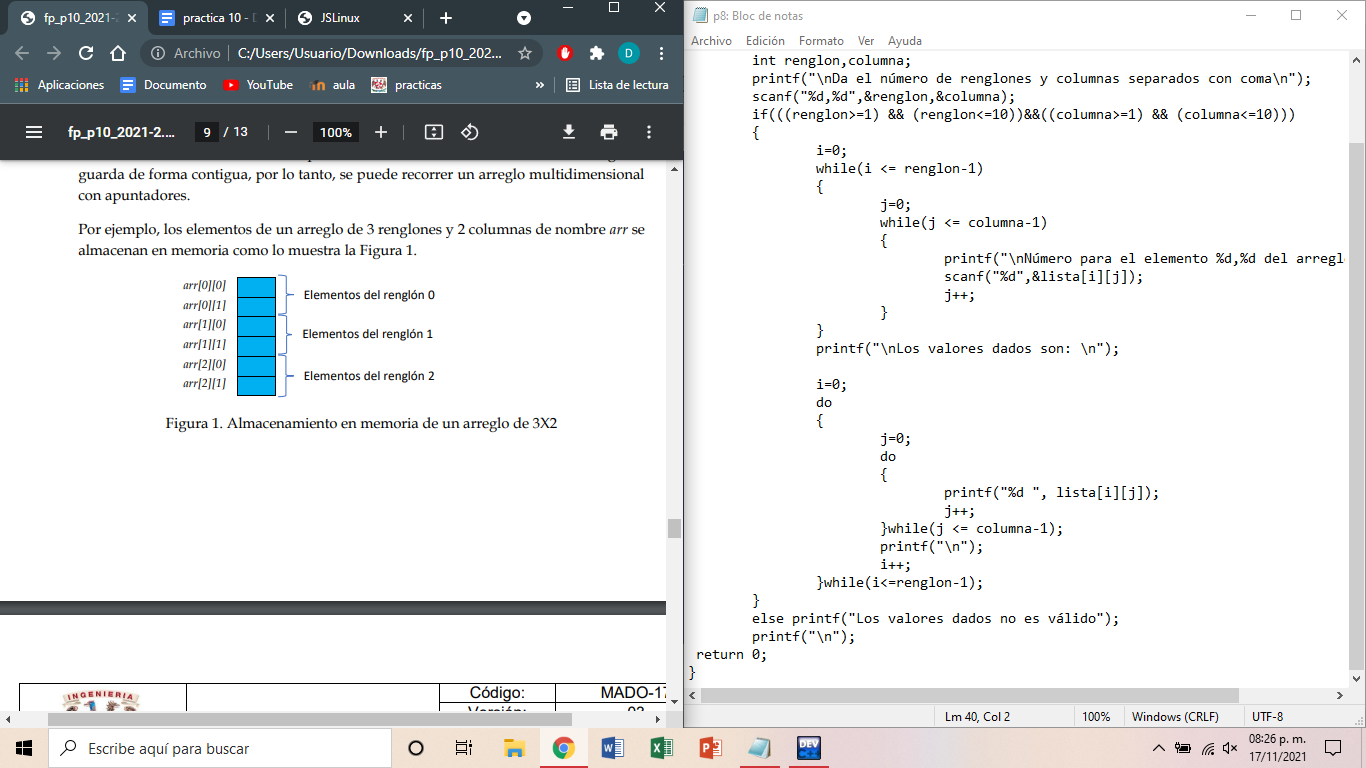
lo modificamos para que sumara



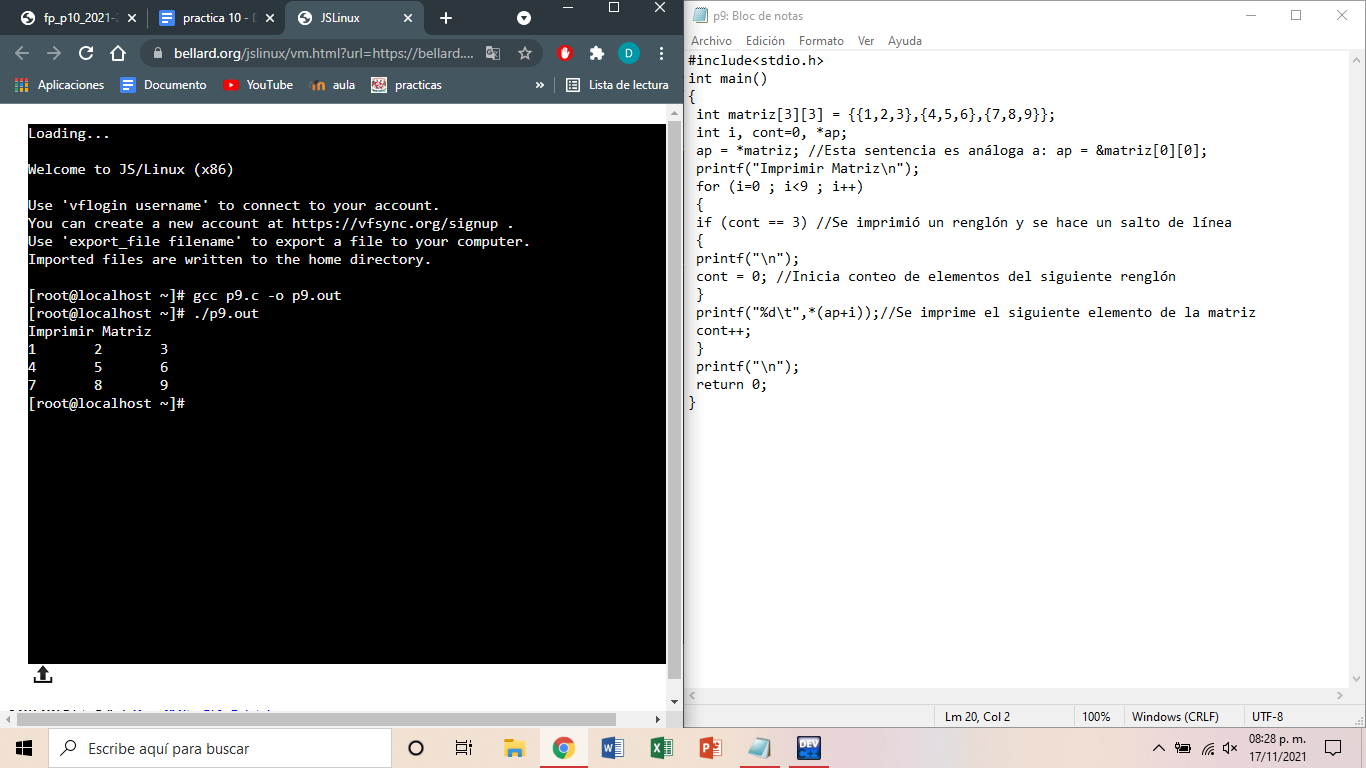
y luego modificamos para ponerle un do while y un while



luego pasamos a ver Recordemos que 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 referirse a la dirección de memoria donde se ubica el valor de dicha variable (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 \*.



y ejecutamos el programa de ejemplo



EJERCICIOS DE TAREA:

1.- Indica que realiza el siguiente programa y completa los letreros del programa:

En este programa nos imprime los nombres que existe en el arreglo en los cuales están los nombre José, teresa, mariana, mario, pedro, juan según el nombre que nosotros ingresamos y nos indica cual escogimos al final del programa

#include <stdio.h>

#define p printf

#define s scanf

int main()

{

char nombres[6][10] = {" jose ","teresa","mariana","mario","pedro","juan"};

char alum[10];

int cont=0;

p("Ingrese el nombre del alumno: \n");

s("%s", alum);

int i;

for(i=0; i<6; i++){

if (strcmp(nombres[i],alum)==0){ // compara el número mayor de nombres[i] con alum

p("El alumno %s aquí indicar que muestra el programa\n", nombres[i]);

}else{

cont++;

}

}

if (cont==6){

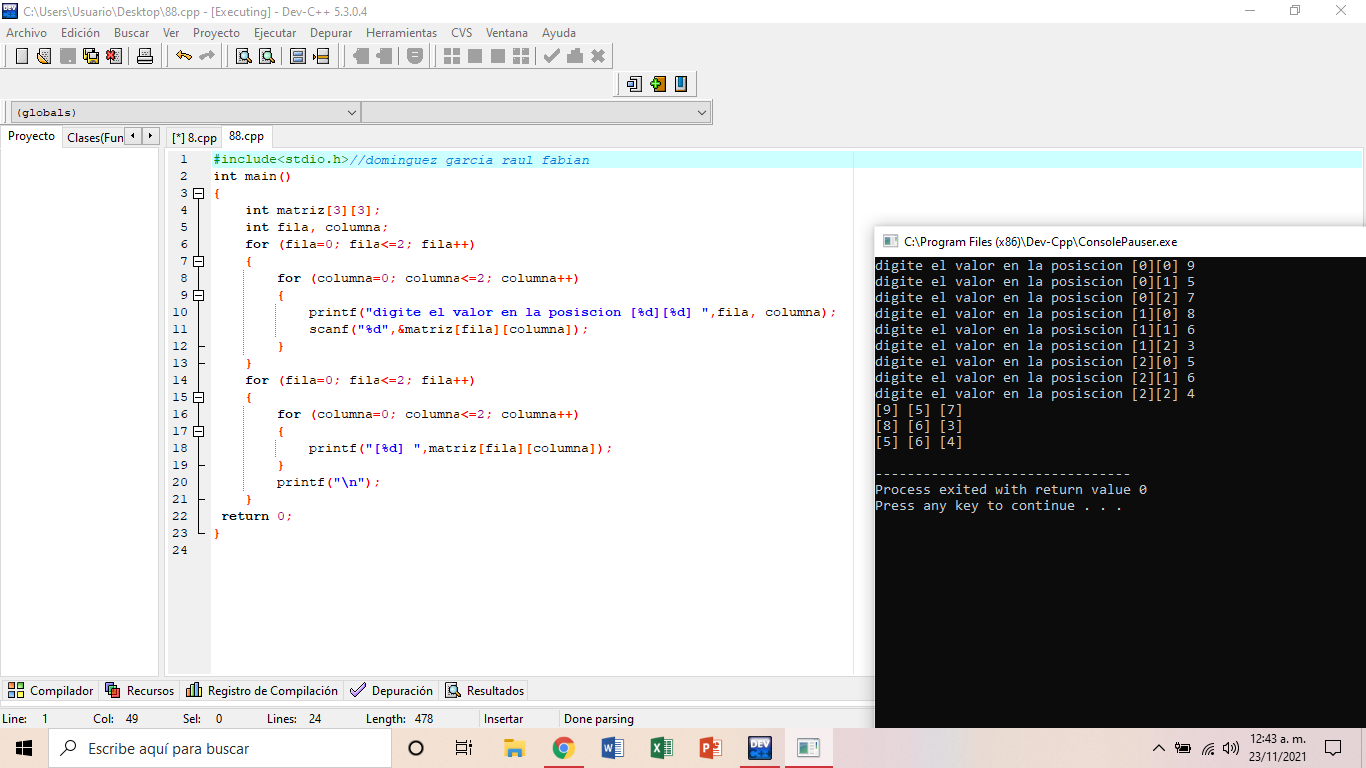
p("%s aquí indicar que muestra el programa ",alum);

}

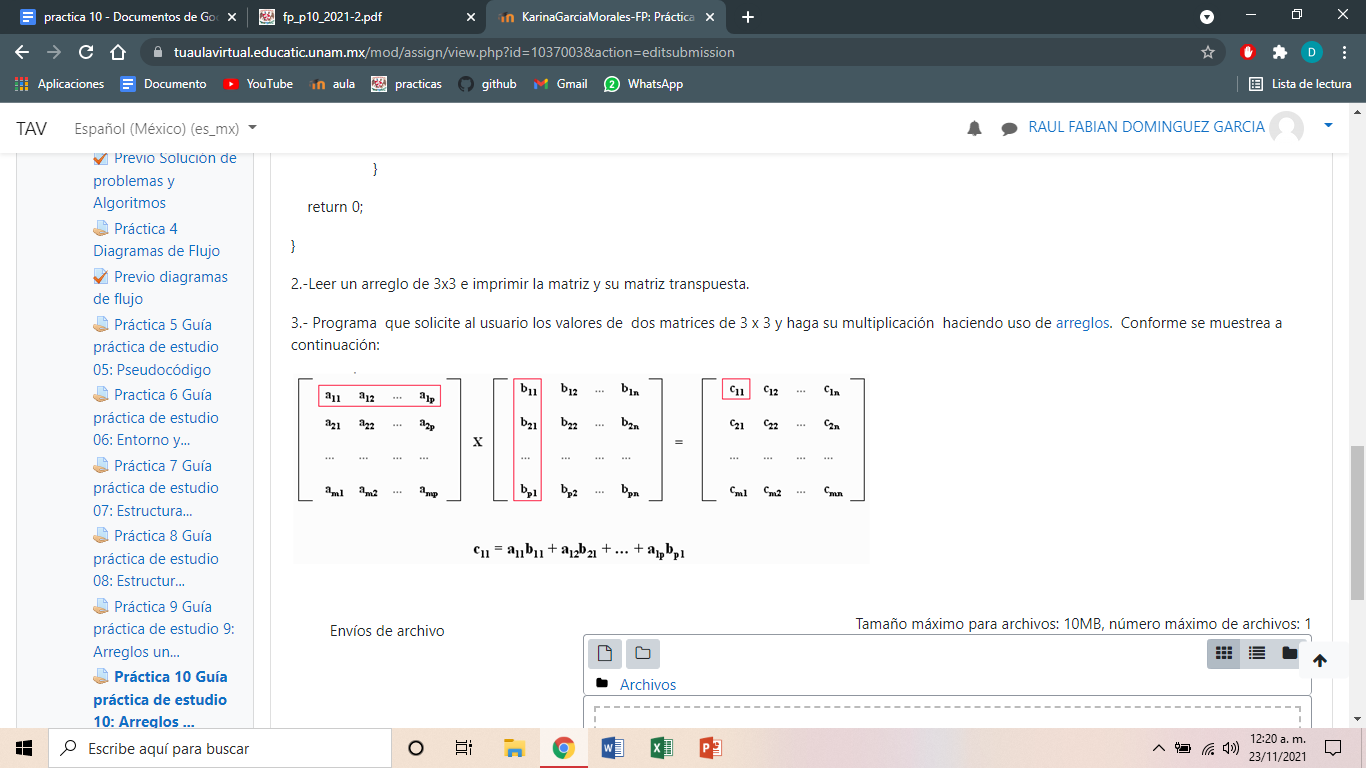
return 0;

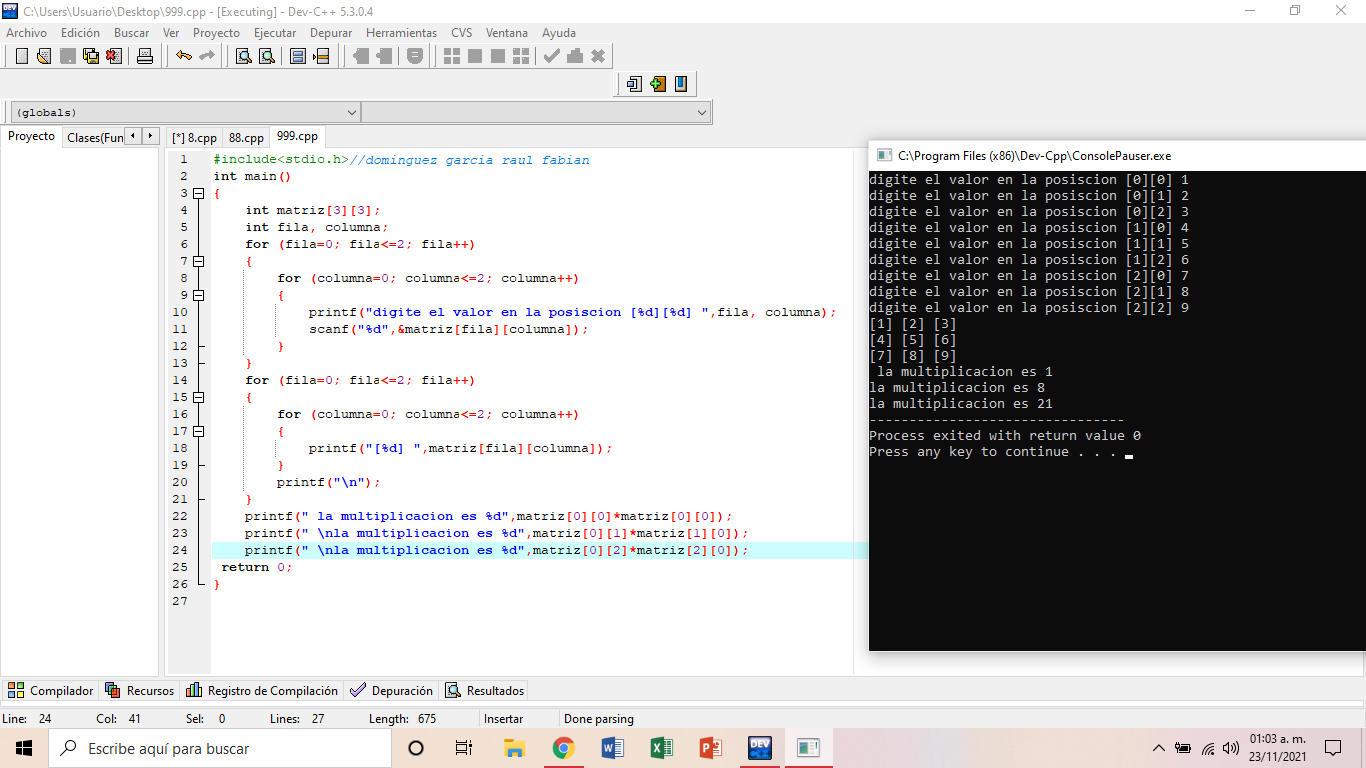
}

2.-Leer un arreglo de 3x3 e imprimir la matriz y su matriz transpuesta.



3.- Programa que solicite al usuario los valores de dos matrices de 3 x 3 y haga su multiplicación haciendo uso de arreglos. Conforme se muestra a continuación:





CONCLUSIONES: logramos comprender y realizar ejercicios donde se utilizan Arreglos multidimensionales y lo logramos realizar nosotros solos desde cero con los ejercicios de tarea y comprendido y sabemos cómo es que se realizar programas para este tipo de arreglos multidimensionales.

BIBLIOGRAFÍA:

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