



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Karina García Morales

Profesor:

Fundamentos de programación

Asignatura:

20

Grupo:

Practica 10

No. de práctica(s):

Vargas Hernandez Edgar Vicente

Integrante(s):

50

No. de lista o brigada:

Primer semestre

Semestre:

06/12/2022

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

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 utilicen dos índices.

Actividades:

- Resolver problemas que requieran el uso de un arreglo de dos dimensiones, a través de programas en lenguaje C.
- Manipular este tipo de arreglos a través de índices.

En la práctica número 10 empezamos a hablar sobre los arreglos multidimensionales

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

```
tipoDato nombre[ tamaño ][ tamaño ]...[tamaño];
```

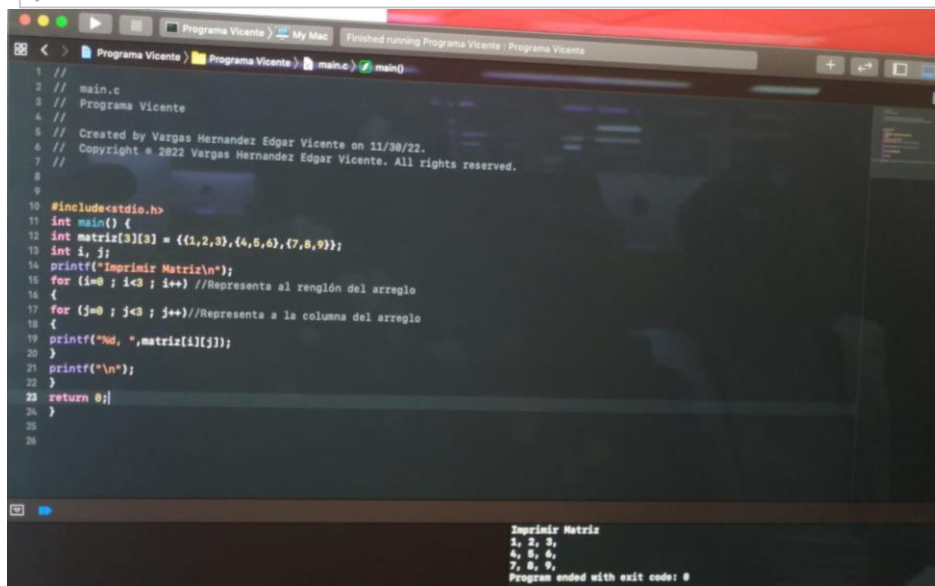
La sintaxis donde se refiere a tamaño es el número entero y define el número máximo de elementos que puede contener un arreglo algo que nos mencionó la profesora es que el número que esté dentro del corchete o paréntesis es lo que va a determinar el número de dimensiones que estén determinadas

De ahí comenzamos a hacer cada 1 de los programas para identificar a quién se refería y cómo funcionaba cada arreglo multidimensional

Programala.c

```
#include<stdio.h>
int main ()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Imprimir Matriz\n");
    for (i=0 ; i<3 ; i++) //Representa al renglón del arreglo
    {
        for (j=0 ; j<3 ; j++)//Representa a la columna del arreglo
        {
            printf("%d, ",matriz[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



The screenshot shows a code editor window titled 'Programa Vicente' with a file named 'main.c'. The code is identical to the one shown in the previous block. Below the code editor, there is a terminal window showing the output of the program. The output is:

```
Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,
Program ended with exit code: 0
```

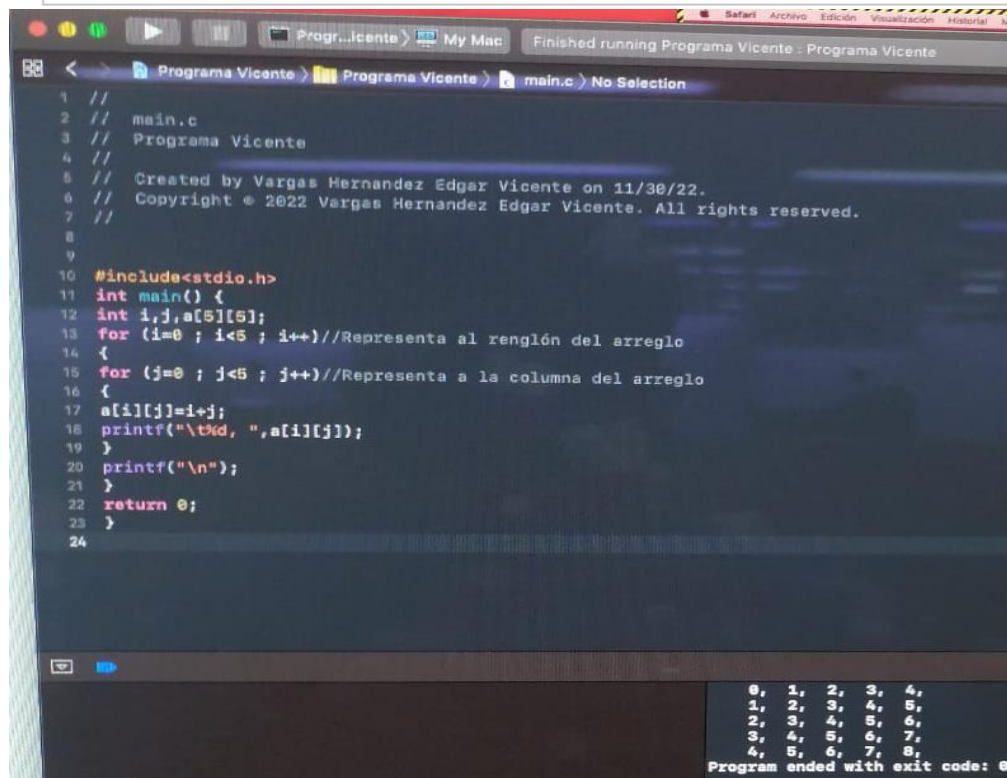
En este programa número 1 la profesora lo único que nos pidió es que viéramos que lo que realiza lo que hace el programa es utilizar un arreglo que está marcado con 3 y luego nos vuelve a poner otro corchete con otro 3 esto refiriéndose a que la cantidad de números que va a haber en cada sección van a ser 3 y la cantidad de secciones van a ser 3 por lo tanto luego nosotros tenemos que especificar los números que vamos a colocar en la parte de la derecha.

Programa2a.c

```
#include<stdio.h>

int main()
{
    int i,j,a[5][5];
    for (i=0 ; i<5 ; i++)//Representa al renglón del arreglo
    {
        for (j=0 ; j<5 ; j++)//Representa a la columna del arreglo
        {
            a[i][j]=i+j;
            printf("\t%d, ",a[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



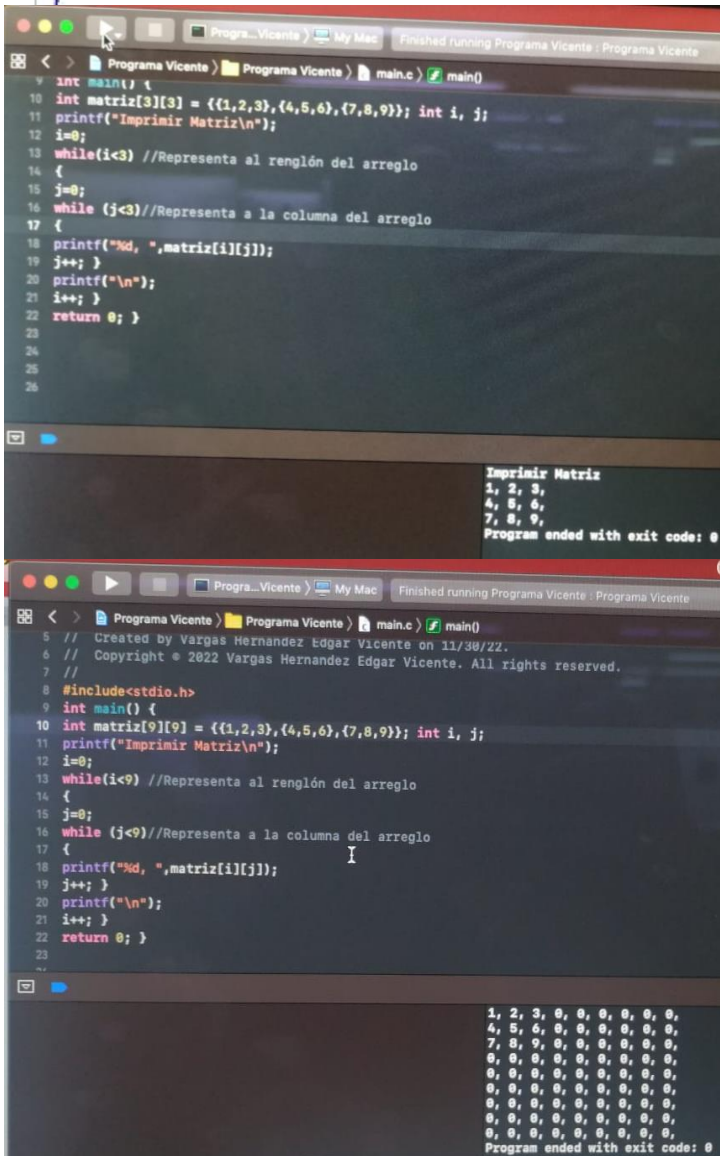
```
1 //
2 // main.c
3 // Programa Vicente
4 //
5 // Created by Vargas Hernandez Edgar Vicente on 11/30/22.
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8
9
10 #include<stdio.h>
11 int main() {
12 int i,j,a[5][5];
13 for (i=0 ; i<5 ; i++)//Representa al renglón del arreglo
14 {
15 for (j=0 ; j<5 ; j++)//Representa a la columna del arreglo
16 {
17 a[i][j]=i+j;
18 printf("\t%d, ",a[i][j]);
19 }
20 printf("\n");
21 }
22 return 0;
23 }
24
```

```
0, 1, 2, 3, 4,
1, 2, 3, 4, 5,
2, 3, 4, 5, 6,
3, 4, 5, 6, 7,
4, 5, 6, 7, 8,
Program ended with exit code: 0
```

Lo mismo sucede en este programa pero en vez de ser el número 3 colocamos el número 5 en este arreglo lo que hacemos es que el mismo programa ejecute los números que se están determinando al momento de ejecutar el “for” cuando nosotros le indicamos que nuestro primer dígito nada más va a abarcar del cero a menor que 5 es como logramos que se establezcan los números que se vayan a colocar en cada 1 de los grupos por eso el resultado resulta en una lista que al principio inicia del cero al cuatro y luego los demás números comienzan del 1 al 5 del 2 al 6 etc

Programa1b.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Imprimir Matriz\n");
    i=0;
    while(i<3) //Representa al renglón del arreglo
    {
        j=0;
        while (j<3)//Representa a la columna del arreglo
        {
            printf("%d, ",matriz[i][j]);
            j++;
        }
        printf("\n");
        i++;
    }
    return 0;
}
```

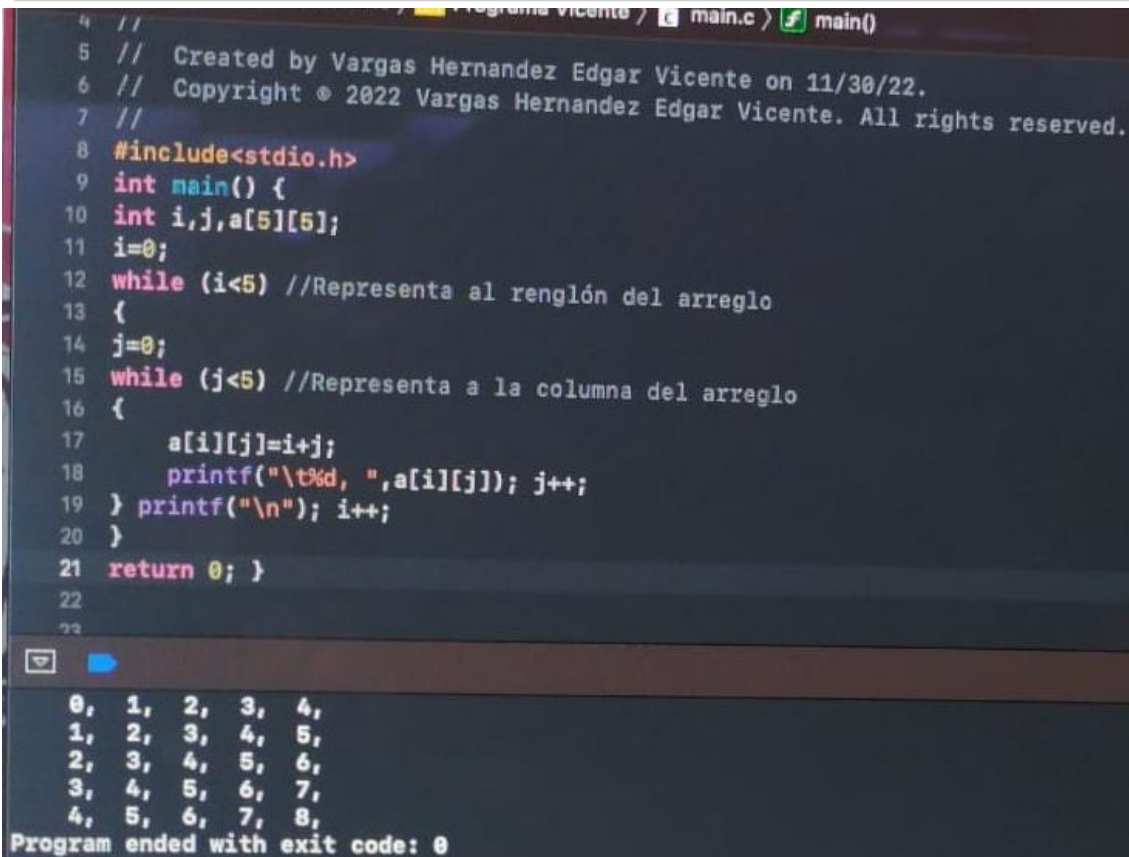


En el siguiente programa lo único que cambiamos es la utilización del while. Donde sin delimitar en qué número comienza en vez de poner i igual a cero dentro del while se coloca antes

Programa2b.c

```
#include<stdio.h>
int main()
{
    int i,j,a[5][5];
    i=0;
    while (i<5) //Representa al renglón del arreglo
    {
        j=0;
        while (j<5) //Representa a la columna del arreglo
```

```
        {
            a[i][j]=i+j;
            printf("\t%d, ",a[i][j]);
            j++;
        }
        printf("\n");
        i++;
    }
    return 0;
}
```



The screenshot shows a code editor with the following C program:

```
4 //
5 // Created by Vargas Hernandez Edgar Vicente on 11/30/22.
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8 #include<stdio.h>
9 int main() {
10 int i,j,a[5][5];
11 i=0;
12 while (i<5) //Representa al renglón del arreglo
13 {
14 j=0;
15 while (j<5) //Representa a la columna del arreglo
16 {
17     a[i][j]=i+j;
18     printf("\t%d, ",a[i][j]); j++;
19 } printf("\n"); i++;
20 }
21 return 0; }
```

The output of the program is a 5x5 matrix of values, where each element is the sum of its row and column indices (i+j). The output is displayed as follows:

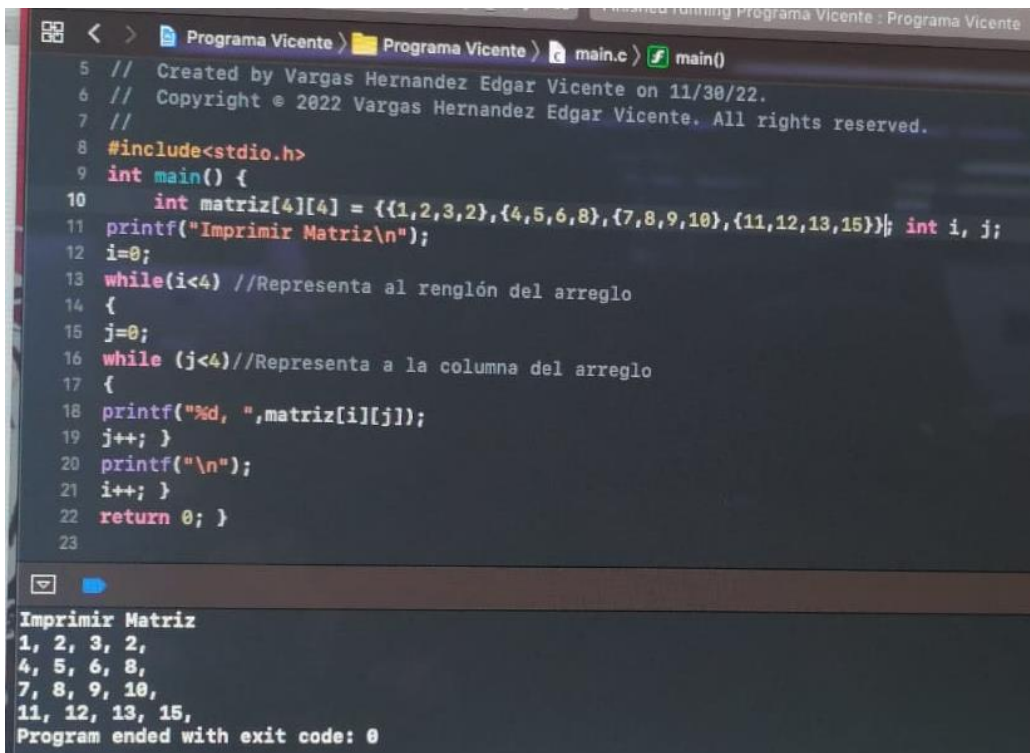
```
0, 1, 2, 3, 4,
1, 2, 3, 4, 5,
2, 3, 4, 5, 6,
3, 4, 5, 6, 7,
4, 5, 6, 7, 8,
Program ended with exit code: 0
```

En este programa la profesora nos pidió que verificásemos cómo funcionaban por lo tanto que teníamos que colocar un número diferente del número 5 que se encontraba en el programa yo coloque el número 9 Al principio no entendía por qué salían todos esos números como ceros cuando yo no los había establecido, pero fue por eso mismo como yo no coloque ningún número que respaldará el valor que faltaba en mis agrupaciones los coloqué como 0

Programal.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Imprimir Matriz\n");
    i=0;
    do //Representa al renglón del arreglo
    {
        j=0;
        do //Representa a la columna del arreglo
        {
            printf("%d, ",matriz[i][j]);
            j++;
        }
        while (j<3);
        printf("\n");

        i++;
    }
    while(i<3);
    return 0;
}
```



```
Programa Vicente > Programa Vicente > main.c > main()
5 // Created by Vargas Hernandez Edgar Vicente on 11/30/22.
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8 #include<stdio.h>
9 int main() {
10     int matriz[4][4] = {{1,2,3,2},{4,5,6,8},{7,8,9,10},{11,12,13,15}}; int i, j;
11     printf("Imprimir Matriz\n");
12     i=0;
13     while(i<4) //Representa al renglón del arreglo
14     {
15         j=0;
16         while (j<4)//Representa a la columna del arreglo
17         {
18             printf("%d, ",matriz[i][j]);
19             j++; }
20         printf("\n");
21         i++; }
22     return 0; }
23
```

Imprimir Matriz
1, 2, 3, 2,
4, 5, 6, 8,
7, 8, 9, 10,
11, 12, 13, 15,
Program ended with exit code: 0

Lo mismo con este programa lo único que hicimos fue cambiar el valor que ya estaba establecido para ver cómo teníamos que agruparlos por lo tanto yo escogí el número cuatro y coloque cualquier número para completar la el arreglo de cuatro dígitos o cuatro números en cuatro columnas

Programa2c.c

```
#include<stdio.h>
int main()
{
    int i,j,a[5][5];
    i=0;
    do //Representa al renglón del arreglo
    {
        j=0;
        do //Representa a la columna del arreglo
        {
            a[i][j]=i+j;
            printf("\t%d, ",a[i][j]);
            j++;
        }
        while (j<5);
        printf("\n");
        i++;
    }
    while (i<5);
    return 0;
}
```

```
5 // Created by Vargas Hernandez Edgar Vicente on 11/30/22.
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8 #include<stdio.h>
9 int main() {
10     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, j;
11     printf("Imprimir Matriz\n");
12     i=0;
13     do //Representa al renglón del arreglo
14     {
15         j=0;
16         do //Representa a la columna del arreglo
17         {
18             printf("%d, ",matriz[i][j]);
19             j++; }
20         while (j<3); printf("\n");
21         i++; }
22     while(i<3);
23     return 0; }
24
25
26
```

Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,
Program ended with exit code: 0

```
1 //
2 // main.c
3 // Programa Vicente
4 //
5 // Created by Vargas Hernandez Edgar Vicente on
6 // Copyright © 2022 Vargas Hernandez Edgar Vics
7 //
8 #include<stdio.h>
9 int main() {
10     int i,j,a[5][5];
11     i=0;
12     do //Representa al renglón del arreglo
13     {
14         j=0;
15         do //Representa a la columna del arreglo
16         {
17             a[i][j]=i+j; printf("\t%d, ",a[i][j]); j++;
18         }
19         while (j<5); printf("\n"); i++;
20     }
21     while (i<5); return 0;
22 }
23
24
25
26
```

0, 1, 2, 3, 4,
1, 2, 3, 4, 5,
2, 3, 4, 5, 6,
3, 4, 5, 6, 7,
4, 5, 6, 7, 8,
Program ended with exit code: 0

Programa3.c

```
#include <stdio.h>
int main ()
{
    int lista[10][10]; // Se declara el arreglo multidimensional
    int i,j;
    int renglon,columna;
    printf("\nDa el número de renglones y columnas separados con coma\n");
    scanf("%d,%d",&renglon,&columna);
    if(((renglon>=1) && (renglon<=10))&&((columna>=1) && (columna<=10)))
    {
        // Acceso a cada elemento del arreglo multidimensional usando for
        for (i= 0 ; i <= renglon-1 ; i++)
        {
            for(j= 0 ; j <= columna-1 ; j++)
            {
                printf("\nNúmero para el elemento %d,%d del arreglo", i,j );
                scanf("%d",&lista[i][j]);
            }
        }
        printf("\nLos valores dados son: \n");
        // Acceso a cada elemento del arreglo multidimensional usando for
        for (i= 0 ; i <= renglon-1 ; i++)
        {
            for(j= 0 ; j <= columna-1 ; j++)
            {
                printf("%d ", lista[i][j]);
            }
            printf("\n");
        }
    }
    else printf("Los valores dados no es válido");
    printf("\n");
    return 0;
}
```

The image shows two side-by-side screenshots of a code editor. The left screenshot displays the source code of 'Programa3.c' with line numbers 1 through 34. The code declares a 10x10 integer array 'lista', prompts the user for the number of rows and columns, validates the input, and then uses nested loops to read values into the array and print them. The right screenshot shows the same code being executed. The output window at the bottom displays the user's input '1,3,4,7' and the program's response, which is a 4x4 grid of zeros, indicating that the input values were not used to populate the array.

```

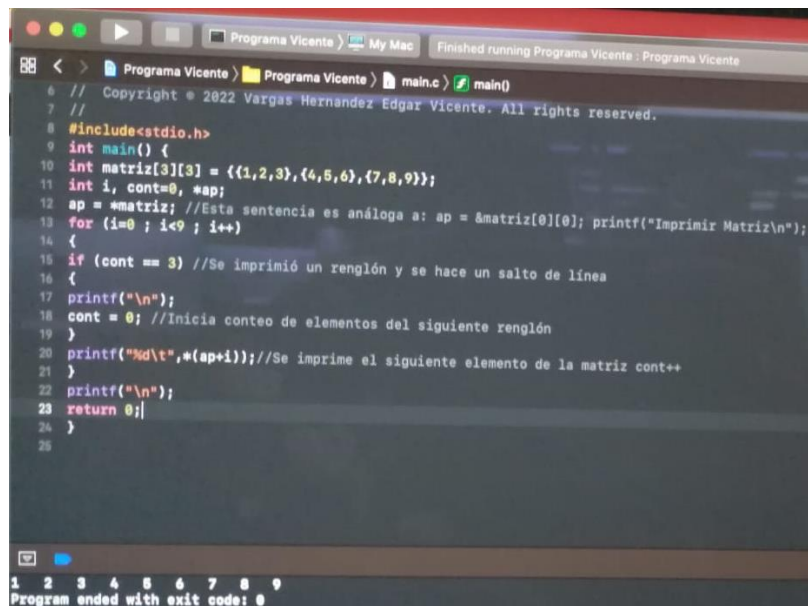
1 // Programa Vicente
2 // main.c
3 // Programa Vicente
4 //
5 // Created by Vargas Hernandez Edgar Vicente on 11/30/22.
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8 #include <stdio.h>
9 int main () {
10     int lista[10][10]; // Se declara el arreglo multidimensional
11     int i,j;
12     int renglon,columna;
13     printf("\nDa el número de renglones y columnas separados con coma\n");
14     scanf("%d,%d",&renglon,&columna);
15     if(((renglon>=1) && (renglon<=10))&&((columna>=1) && (columna<=10)))
16     {
17         // Acceso a cada elemento del arreglo multidimensional usando for
18         for (i= 0 ; i <= renglon-1 ; i++) {
19             for(j= 0 ; j <= columna-1 ; j++) {
20                 printf("\nNúmero para el elemento %d,%d del arreglo", i,j );
21                 scanf("%d",&lista[i][j]);
22             }
23         }
24         printf("\nLos valores dados son: \n");
25         // Acceso a cada elemento del arreglo multidimensional usando for
26         for (i= 0 ; i <= renglon-1 ; i++) {
27             for(j= 0 ; j <= columna-1 ; j++) {
28                 printf("%d ", lista[i][j]);
29             }
30             printf("\n");
31         }
32     }
33     else printf("Los valores dados no es válido"); printf("\n");
34     return 0;
35 }
```

Da el número de renglones y columnas separados con coma
1,3,4,7

Número para el elemento 0,0 del arreglo
Número para el elemento 0,1 del arreglo
Número para el elemento 0,2 del arreglo
Los valores dados son:
0 0 0

Programa4a.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, cont=0, *ap;
    ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
    printf("Imprimir Matriz\n");
    for (i=0 ; i<9 ; i++)
    {
        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
        {
            printf("\n");
            cont = 0; //Inicia conteo de elementos del siguiente renglón
        }
        printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
        cont++;
    }
    printf("\n");
    return 0;
}
```



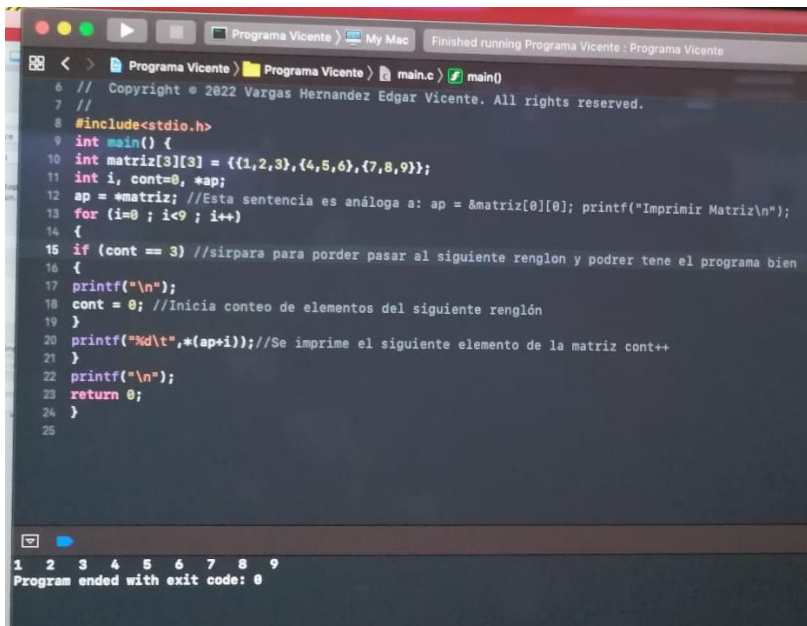
```
Programa Vicente > Programa Vicente > main.c > main()
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8 #include<stdio.h>
9 int main() {
10 int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
11 int i, cont=0, *ap;
12 ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0]; printf("Imprimir Matriz\n");
13 for (i=0 ; i<9 ; i++)
14 {
15 if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
16 {
17 printf("\n");
18 cont = 0; //Inicia conteo de elementos del siguiente renglón
19 }
20 printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz cont++
21 }
22 printf("\n");
23 return 0;
24 }
25

1 2 3 4 5 6 7 8 9
Program ended with exit code: 0
```

Programa4b.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, cont=0, *ap;
    ap = *matriz;//Esta sentencia es análoga a: ap = &matriz[0][0];
    printf("Imprimir Matriz\n");
    i=0;
    while (i<9)
    {
        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
        {
            printf("\n");
            cont = 0; //Inicia conteo de elementos del siguiente renglón
        }
        printf("%d\t",*(ap+i));//Se imprime el siguiente elemento de la matriz

        cont++;
        i++;
    }
    printf("\n");
    return 0;
}
```



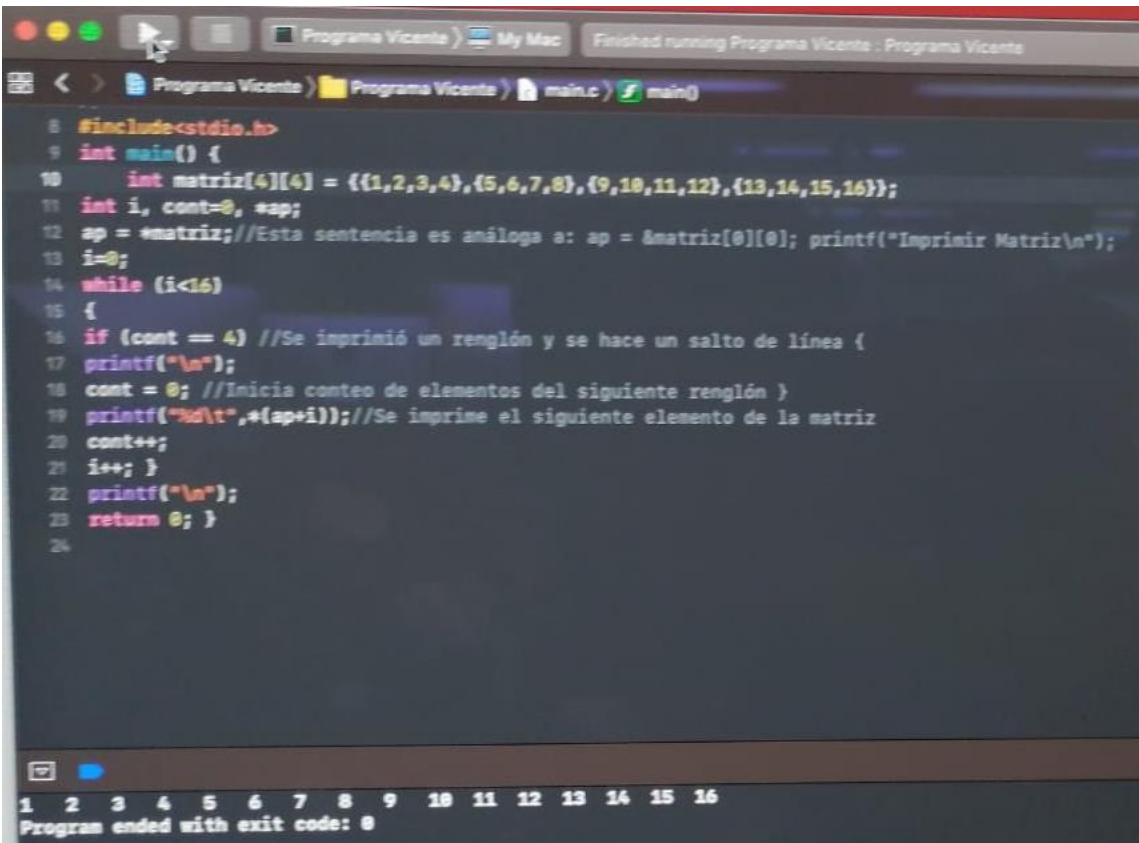
```
Programa Vicente > My Mac Finished running Programa Vicente : Programa Vicente
6 // Copyright © 2022 Vargas Hernandez Edgar Vicente. All rights reserved.
7 //
8 #include<stdio.h>
9 int main() {
10 int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
11 int i, cont=0, *ap;
12 ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0]; printf("Imprimir Matriz\n");
13 for (i=0 ; i<9 ; i++)
14 {
15 if (cont == 3) //sirpara para poder pasar al siguiente renglon y podrer tene el programa bien
16 {
17 printf("\n");
18 cont = 0; //Inicia conteo de elementos del siguiente renglón
19 }
20 printf("%d\t",*(ap+i));//Se imprime el siguiente elemento de la matriz cont++
21 }
22 printf("\n");
23 return 0;
24 }
25

1 2 3 4 5 6 7 8 9
Program ended with exit code: 0
```

```

#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, cont=0, *ap;
    ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
    printf("Imprimir Matriz\n");
    i=0;
    do
    {
        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
        {
            printf("\n");
            cont = 0; //Inicia conteo de elementos del siguiente renglón
        }
        printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
        cont++;
        i++;
    }
    while (i<9);
    printf("\n");
    return 0;
}

```



```

8 #include<stdio.h>
9 int main() {
10     int matriz[4][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
11     int i, cont=0, *ap;
12     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0]; printf("Imprimir Matriz\n");
13     i=0;
14     while (i<16)
15     {
16         if (cont == 4) //Se imprimió un renglón y se hace un salto de línea {
17             printf("\n");
18             cont = 0; //Inicia conteo de elementos del siguiente renglón }
19             printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
20             cont++;
21             i++; }
22     printf("\n");
23     return 0; }
24
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Program ended with exit code: 0

```

En la última parte lo único que hacemos es quitar las filas y hacerlo todo en un solo renglón esto aun respetando los arreglos que se pusieron que en este caso sería (4) (4)

La profesora nos pidió hacer un cuadro comparativo entre for, while, do while de los ejercicios dos para que notemos la diferencia entre cada 1

Ejercicios numero 2

For	While	Do-while
<pre>int main() { int i,j,a[5][5]; for (i=0 ; i<5 ; i++)//Representa al { for (j=0 ; j<5 ; j++)//Representa { a[i][j]=i+j; printf("\t%d, ",a[i][j]); } printf("\n"); } return 0; }</pre> <p>En For notó que la restricción se coloca totalmente dentro o al lado de El For</p>	<pre>#include<stdio.h> int main() { int i,j,a[5][5]; i=0; while (i<5) //Representa al renglón del { j=0; while (j<5) //Representa a la colu { a[i][j]=i+j; printf("\t%d, ",a[i][j]); j++; } printf("\n"); i++; } return 0; }</pre> <p>En while nada más se coloca la restricción para saber hasta qué punto parar de colocar números</p>	<pre>#include<stdio.h> int main() { int i,j,a[5][5]; i=0; do //Representa al renglón del arr { j=0; do //Representa a la columna de { a[i][j]=i+j; printf("\t%d, ",a[i][j]); j++; } while (j<5); printf("\n"); i++; } while (i<5); return 0; }</pre> <p>En el último lo que noto es que utiliza mucho la estructura del segundo Pero al momento de determinar las variables lo hace dentro del do</p>

Ejercicio numero 4

For	While	Do-while
<pre>#include<stdio.h> int main() { int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, cont=0, *ap; ap = *matriz; //Esta sentencia es análoga a: ap = printf("Imprimir Matriz\n"); for (i=0 ; i<9 ; i++) { if (cont == 3) //Se imprimió un renglón y se { printf("\n"); cont = 0; //Inicia conteo de elementos } printf("%d\t",*(ap+i)); //Se imprime el siguiente cont++; } printf("\n"); return 0; }</pre> <p>Al igual que en el ejercicio dos las restricciones se ponen al lado del Ford pero al momento de designar qué número colocaremos se hace hasta el final a diferencia de los otros dos que se hace en medio el programa</p>	<pre>#include<stdio.h> int main() { int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, cont=0, *ap; ap = *matriz; //Esta sentencia es análoga a: ap = printf("Imprimir Matriz\n"); i=0; while (i<9) { if (cont == 3) //Se imprimió un renglón y se { printf("\n"); cont = 0; //Inicia conteo de elementos d } printf("%d\t",*(ap+i)); //Se imprime el sigui cont++; i++; } printf("\n"); return 0; }</pre> <p>Lo que noto esque coloca el valor de y desde un inicio igualándolo a cero</p>	<pre>#include<stdio.h> int main() { int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, cont=0, *ap; ap = *matriz; //Esta sentencia es análoga a: ap = printf("Imprimir Matriz\n"); i=0; do { if (cont == 3) //Se imprimió un renglón y se { printf("\n"); cont = 0; //Inicia conteo de elementos } printf("%d\t",*(ap+i)); //Se imprime el siguiente cont++; i++; } while (i<9); printf("\n"); return 0; }</pre> <p>Al igual que en el anterior declara i igual a cero al principio pero éste no describe el resultado sino hasta el final</p>

1.-Realiza un programa que muestre tu nombre y número de cuenta haciendo uso de 2 [arreglos](#), emplear while y for

```
#include<stdio.h>

int main()

{

int matriz[3][3] = {{3,2,0},{0,0,5},{5,7,3}};

int i, cont=0, *ap;

ap = *matriz;

printf("Coloca tu nombre empezando por apellidos");

scanf("cadena")

printf("Imprimir Matriz\n");

i=0;

do

{ if (cont == 3)

{

printf("\n");

cont = 0;

}

printf("%d\t",*(ap+i));

cont++; i++;

}

while (i<9);

printf("\n");

return 0;

}
```

2.-Modifica el programa del ejercicio 1 utilizando apuntadores

```
#include<stdio.h>
```



```

int main()

{

int a=[9] = { 1,2,3,4,5,6,7,8,9};

int *ap=lista;

printf("a=[9] = { 1,2,3,4,5,6,7,8,9};

for (indice = 0 ; indice < 9 ; indice++)

{

printf("\nCalificación del alumno %d es %d", indice+1, *(ap+indice));

}

printf("\n");

return 0;

}

```

3.- Corrige e indica que realiza el siguiente programa:

#include<stdio.h>

Int main()

{

int i,j, cont=0,n;

float M[2][2], s=0, *ap;

ap = M;

for(i=0; i<=3; i++)

{

for(j=0; j<=1;j--)

{

printf("\t\n Teclear el elemento %d %d \t", i+1,j+1);

scanf("%f",&M[i][j]);

s+=M[i][j];

```

    }

}

printf("\tLa Matriz es:\n");

for (n=0 ; n<4 ; n++)

{

    if (cont == 2)

    {

        printf("\n");

        cont = 1;

    }

    printf("%.2f\t\t", *(ap+n));

    cont++;

}

printf("\t\n\n La suma de los elementos es:%0.2f", s);

return 0;

}

```

Se le agrego el "int" por que no lo tenia

Conclusiones

La verdad la práctica me resultó bastante sencilla al momento de que la profesora nos estaba explicando cómo funcionaba pero cuando empecé a tratar de resolverlo por mi parte se me hizo difícil no sabía cómo organizarlo hasta que un compañero me dijo que lo podía poner como yo quisiera mientras que funcionará el arreglo por lo que cuando lo fui haciendo poco a poco le fui comprendiendo más por lo que me pareció una buena práctica difícil si se me hizo pero creo que estuvo muy bien la práctica

https://github.com/Chente117/practica1_fbp.git