



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 20

No. de práctica(s): No. 10 - Arreglos multidimensionales.

Integrante(s): López Olmedo Ernesto Yael

No. de lista o brigada: 27

Semestre: 2024-1

Fecha de entrega: 31 de Octubre 2023

Observaciones:

CALIFICACIÓN: _____

Introducción

Para la continuación el aprendizaje del lenguaje C y del tema de arreglos unidimensionales, ahora en la siguiente presentación de esta práctica consistirá en crear arreglos de dos dimensiones, como es que estos pueden ser concatenados, como se representan de manera gráfica, cómo estos pueden ser usados y manipulados mediante a diversas estructuras de control. Con esto se verá la comprensión por parte de los alumnos al uso de este tipo de arreglo, las condiciones de uso, sus restricciones, dando por cumplido el objetivo de la práctica.

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

-Laboratorio:

Programa 1a.c

```
1  #include <stdio.h>
2  //López Olmedo Ernesto Yael No.27
3  int main()
4  {
5      int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
6      int i, j;
7      printf("Imprimir Matriz\n");
8      for (i=0 ; i<3 ; i++)
9          //Representa al renglón del arreglo
10     {
11         for (j=0 ; j<3 ; j++)
12             //Representa a la columna del arreglo
13         {
14             printf("%d, ",matriz[i][j]);
15         }
16         printf("\n");
17     }
18     return 0;
19 }
20
```



input

1, 2, 3,
4, 5, 6,
7, 8, 9,

Figura 1.1 - Programa 1a.c

Programa 2a.c

```
1  #include <stdio.h>
2  //López Olmedo Ernesto Yael No.27
3  int main()
4  {
5      int i,j,a[5][5];
6      for (i=0 ; i<5 ; i++)
7          //Representa al renglón del arreglo
8      {
9          for (j=0 ; j<5 ; j++)
10             //Representa a la columna del arreglo
11         {
12             a[i][j]=i+j;
13             printf("\t%d, ",a[i][j]);
14         }
15     }
16 }
```

Figura 1.2 - Programa 2a.c

Programa 1b.c

```
1  #include <stdio.h>
2  //López Olmedo Ernresto Yael No.27
3  int main(){
4      int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5      int i, j;
6      printf("Imprimir Matriz\n");
7      i=0;
8      while(i<3)
9          //Representa al renglón del arreglo
10         {
11             j=0;
12             while (j<3)
13                 //Representa a la columna del arreglo
14                 {
15                     printf("%d, ",matriz[i][j]);
16                     j++;
17                 }
18             printf("\n");
19             i++;
20         }
21     return 0;
22 }
23
```

input

Imprimir Matriz

1, 2, 3,
4, 5, 6,
7, 8, 9,

Programa 2b.c

```
1  #include <stdio.h>
2  //López Olmedo Ernresto Yael No.27
3  int main(){
4      int i,j,a[5][5];
5      i=0;
6      while (i<5)
7          //Representa al renglón del arreglo
8          {
9              j=0;
```

Figura 1.4 - Programa 2b.c

Programa 1c.c

```
1 #include <stdio.h>
2 //López Olmedo Ernesto Yael No.27
3 int main(){
4     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5     int i, j;
6     printf("Imprimir Matriz\n");
7     i=0;
8     do //Representa al renglón del arreglo
9     {
10         j=0;
11         do //Representa a la columna del arreglo
12         {
13             printf("%d, ",matriz[i][j]);
14             j++;
15         }
16         while (j<3);
```

Figura 1.5 - Programa 1c.c

Programa 2c.c

```
1 #include <stdio.h>
2 //López Olmedo Ernresto Yael No.27
3 int main(){
4     int i,j,a[5][5];
5     i=0;
6     do //Representa al renglón del arreglo
7     {
8         j=0;
9         do
10         //Representa a la columna del arreglo
11         {
12             a[i][j]=i+j;
13             printf("\t%d, ",a[i][j]);
14             j++;
15         }
16         while (j<5);
17         printf("\n");
18         i++;
19     }
20     while (i<5);
21     return 0;
22 }
```

input				
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,

Figura 1.6 - Programa 2c.c

Programa 3.c

```
1 #include <stdio.h>
2 //López Olmedo Ernresto Yael No.27
3 int main(){
4     int lista[10][10];
5     // Se declara el arreglo multidimensional
6     int i,j;
7     int renglon,columna;
8     printf("\nDa el número de renglones y columnas separados con coma\n");
9     scanf("%d,%d",&renglon,&columna);
10     if(((renglon>=1) && (renglon<=10))&&((columna>=1) && (columna<=10))){
11         // Acceso a cada elemento del arreglo multidimensional usando for
12         for (i= 0 ; i <= renglon-1 ; i++){
13             for(j= 0 ; j <= columna-1 ; j++){
```

Figura 1.7 - Programa 3c.c

Programa 4a.c

```
1  #include <stdio.h>
2  //López Olmedo Ernesto Yael No.27
3  int main(){
4      int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5      int i, cont=0, *ap;
6      ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
7      printf("Imprimir Matriz\n");
8      for (i=0 ; i<9 ; i++){
9          if (cont == 3)
10             //Se imprimió un renglón y se hace un salto de línea
11             {
12                 printf("\n");
13                 cont = 0;
```

Figura 1.8 - Programa 3a.c

Programa 4b.c

```
1 #include <stdio.h>
2 //López Olmedo Ernresto Yael No.27
3 int main(){
4     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5     int i, cont=0, *ap;
6     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
7     printf("Imprimir Matriz\n");
8     i=0;
9     while (i<9){
10         if (cont == 3)
11             //Se imprimió un renglón y se hace un salto de línea
12             {
13                 printf("\n");
14                 cont = 0; //Inicia conteo de elementos del siguiente renglón
15             }
16         printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
17         cont++;
18         i++;
19     }
20     printf("\n");
21     return 0;
22 }
```

input

```
Imprimir Matriz
1      2      3
4      5      6
7      8      9
```

Figura 1.9 - Programa 3a.c

Programa 4c.c

```
1 #include <stdio.h>
2 //López Olmedo Ernresto Yael No.27
3 int main(){
4     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5     int i, cont=0, *ap;
6     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
7     printf("Imprimir Matriz\n");
8     i=0;
9     do{
10         if (cont == 3)
11             //Se imprimió un renglón y se hace un salto de línea
12             {
13                 printf("\n");
14                 cont = 0;
15                 //Inicia conteo de elementos del siguiente renglón
16             }
17         printf("%d\t",*(ap+i));
18         //Se imprime el siguiente elemento de la matriz
19         cont++;
20         i++;
21     } while (i<9);
22 }
```


Figura 1.10 - Programa 4c.c

Tarea

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() {
```

```

// Declarar arreglos

char nombre[] = "Ernesto Olmedo";

char numeroDeCuenta[] = "319331505";

// Mostrar el nombre

printf("Nombre: ");

for (int i = 0; nombre[i] != '\0'; i++) {

    printf("%c", nombre[i]);

}

printf("\n");


// Mostrar el número de cuenta

printf("No.Cuenta: ");

int i = 0;

while (numeroDeCuenta[i] != '\0') {

    printf("%c", numeroDeCuenta[i]);

    i++;

}

printf("\n");

return 0;

}

```

Terminal

```

1  #include <stdio.h>
2
3  int main() {
4      // Declarar arreglos
5      char nombre[] = "Ernesto Olmedo";
6      char numeroDeCuenta[] = "319331505";
7      // Mostrar el nombre
8      printf("Nombre: ");
9      for (int i = 0; nombre[i] != '\0'; i++) {
10         printf("%c", nombre[i]);
11     }

```

Figura 1.11 - Programa tarea 1

2.-Modifica el programa del ejercicio (Programa3.c) para aplicar el uso de
apuntadores(página 158)

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

```
int main() {
```

```
    int lista[10][10];
```

```

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))) {

    int (*ptr_lista)[10] = lista; // Declarar el puntero

    //Puntero y for

    for (int i = 0; i < renglon; i++) {

        for (int j = 0; j < columna; j++) {

            printf("\nNúmero para el elemento %d,%d del arreglo", i, j);

            scanf("%d", &ptr_lista[i][j]);

        }

    }

    printf("\nLos valores dados son: \n");

    //Punteros y for

    for (int i = 0; i < renglon; i++) {

        for (int j = 0; j < columna; j++) {

            printf("%d ", ptr_lista[i][j]);

        }

        printf("\n");

    }

} else {

    printf("Los valores dados no son válidos");

}

printf("\n");

return 0;

}

```

Terminal

```
Número para el elemento 0,0 del arreglo1
Número para el elemento 0,1 del arreglo2
Número para el elemento 0,2 del arreglo3
Número para el elemento 1,0 del arreglo4
Número para el elemento 1,1 del arreglo5
Número para el elemento 1,2 del arreglo6
Número para el elemento 2,0 del arreglo7
Número para el elemento 2,1 del arreglo8
Número para el elemento 2,2 del arreglo9
Número para el elemento 3,0 del arreglo1
Número para el elemento 3,1 del arreglo2
Número para el elemento 3,2 del arreglo3
Número para el elemento 4,0 del arreglo4
Número para el elemento 4,1 del arreglo5
Número para el elemento 4,2 del arreglo6

Los valores dados son:
1 2 3
4 5 6
7 8 9
1 2 3
4 5 6
```

Figura 1.12 - Programa tarea 2

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

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

```

Lo que este programa hace es hacer un matriz de 2x2, únicamente lo que hace es la suma de los elementos de este, la corrección es la siguiente:

```

#include <stdio.h>

int main() {

    int i, j, cont = 0;

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

    ap = &M[0][0]; //Asignación ap

    for (i = 0; i < 2; i++) { //Rango del primer bucle

        for (j = 0; j < 2; j++) { //Rango del segundo bucle

            printf("Teclear el elemento %d %d: ", i + 1, j + 1);

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

```

```

        s += M[i][j];

    }

}

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

for (i = 0; i < 2; i++) {

    for (j = 0; j < 2; j++) {

        //Corrección ingreso datos

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

    }

    printf("\n");

}

printf("La suma de los elementos es: %.2f\n", s);

return 0;

}

```

Terminal

```

1  #include <stdio.h>
2
3  int main() {
4      int i, j, cont = 0;
5      float M[2][2], s = 0, *ap;
6      ap = &M[0][0]; //Asignación ap
7
8      for (i = 0; i < 2; i++) { //Rango del primer bucle
9          for (j = 0; j < 2; j++) { //Rango del segundo bucle
10             printf("Teclear el elemento %d %d: ", i + 1, j + 1);
11             scanf("%f", &M[i][j]);
12             s += M[i][j];
13         }
14     }

```

Figura 1.13 - Programa tarea 3

4.- 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:

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

```
int main() {
```



```
int matriz1[3][3];

int matriz2[3][3];

int resultado[3][3];


// Solicitar valores para la primera matriz

printf("Ingrese los valores para la primera matriz (3x3):\n");

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

    for (int j = 0; j < 3; j++) {

        printf("Ingrese el valor para la fila %d y columna %d: ", i + 1, j + 1);

        scanf("%d", &matriz1[i][j]);

    }

}

// Solicitar valores para la segunda matriz

printf("Ingrese los valores para la segunda matriz (3x3):\n");

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

    for (int j = 0; j < 3; j++) {

        printf("Ingrese el valor para la fila %d y columna %d: ", i + 1, j + 1);

        scanf("%d", &matriz2[i][j]);

    }

}

// Multiplicar las dos matrices

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

    for (int j = 0; j < 3; j++) {

        resultado[i][j] = 0;

        for (int k = 0; k < 3; k++) {

            resultado[i][j] += matriz1[i][k] * matriz2[k][j];

        }

    }

}
```

```

    }

}

// Mostrar el resultado

printf("El resultado de la multiplicación de matrices es:\n");

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

    for (int j = 0; j < 3; j++) {

        printf("%d ", resultado[i][j]);

    }

    printf("\n");

}

return 0;

}

```

Terminal

```

Ingrese los valores para la primera matriz (3x3):
Ingrese el valor para la fila 1 y columna 1: 1
Ingrese el valor para la fila 1 y columna 2: 2
Ingrese el valor para la fila 1 y columna 3: 3
Ingrese el valor para la fila 2 y columna 1: 1
Ingrese el valor para la fila 2 y columna 2: 2
Ingrese el valor para la fila 2 y columna 3: 3
Ingrese el valor para la fila 3 y columna 1: 1
Ingrese el valor para la fila 3 y columna 2: 2
Ingrese el valor para la fila 3 y columna 3: 3
Ingrese los valores para la segunda matriz (3x3):
Ingrese el valor para la fila 1 y columna 1: 1
Ingrese el valor para la fila 1 y columna 2: 2
Ingrese el valor para la fila 1 y columna 3:
3
Ingrese el valor para la fila 2 y columna 1: 1
Ingrese el valor para la fila 2 y columna 2: 2
Ingrese el valor para la fila 2 y columna 3: 3
Ingrese el valor para la fila 3 y columna 1: 1
Ingrese el valor para la fila 3 y columna 2: 2
Ingrese el valor para la fila 3 y columna 3: 3
El resultado de la multiplicación de matrices es:
6 12 18
6 12 18
6 12 18

```

Figura 1.14 - Programa tarea 4

Conclusión

Para esta práctica se comprendió el uso de los arreglos multidimensionales, las cuales son estructuras que permiten el almacenamiento, manipulación de datos en múltiples dimensiones, representados en matrices. Se trabajó en su sintaxis, para la concatenación de estos y para conseguir el acceso a los elementos, a la memoria o lo que el usuario desee

hacer comprendiendo el concepto de almacenamiento de la información a través de filas y columnas. Al mismo tiempo se experimentó los conocimientos anteriores como las estructuras de control e iterativas con las cuales se manipulan o se extraen datos de los arreglos.

Referencias

- Templos, A.(2022).Manual de prácticas del laboratorio de Fundamentos de programación. México: UNAM.