



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 15

No. de Práctica(s): 1

Integrante(s): María Guadalupe Martínez Pavón

*No. de Equipo de
cómputo empleado:* _____

No. de Lista o Brigada: _____

Semestre: 2021-2

Fecha de entrega: 1 marzo 2021

Observaciones: Había un pequeño error en el código, ya que
faltaban algunos corchetes

CALIFICACIÓN: _____

Práctica 1: Aplicaciones de arreglos

Objetivo:

Utilizar arreglos unidimensionales y multidimensionales para dar solución a problemas computacionales.

Actividades:

- Crear arreglos unidimensionales.
- Crear arreglos multidimensionales.

Introducción:

Un arreglo es una estructura de datos almacenados con una sola variable, que tienen un tamaño definido al momento de crearse, los arreglos comienzan desde la posición 0 hasta la posición n-1, existen dos tipos de arreglos unidimensionales y multidimensionales, los arreglos unidimensionales es un tipo de datos estructurado que está formado de una colección finita y ordenada de datos del mismo tipo podemos acceder a cualquier elemento del arreglo sin tener que consultar a elementos anteriores o posteriores.

Los arreglos multidimensionales es un arreglo con más de dos dimensiones o sea es un arreglo de arreglos los elementos se encuentran organizados en varias dimensiones, los utilizamos cuando se necesita almacenar múltiples valores del mismo tipo, también permite agrupar muchos datos usando un mismo identificador.

Código "Escitala espartan"

```
#include <stdio.h>

void crearMensaje();
void descifrarMensaje();

int main(void)
{
    short opcion=0;

    while (1){
        printf("\n\t*** ESCÍTALA ESPARTANA ***\n");
        printf("¿Qué desea realizar?\n");
        printf("1) Crear mensaje cifrado.\n");
        printf("2) Descifrar mensaje.\n");
        printf("3) Salir.\n");
        scanf("%d", &opcion);
        switch(opcion){
            case 1:
                crearMensaje();
                break;
            case 2:
                descifrarMensaje();
                break;
            case 3:
                return 0;
            default:
                printf("Opción no válida.\n");
        }
    }
}
```

```

    }
    }
    return 0;
}

void crearMensaje(){
    int ren, col, i, j, k=0;
    printf("Ingresar el tamaño de la escítala:\n");
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);
    char escitala[ren][col];
    char texto[ren*col];

    printf("Escriba el texto a cifrar:\n");
    scanf("%s", texto);

    for (i=0 ; i<ren ; i++)
        for (j=0 ; j<col ; j++)
            escitala[i][j] = texto[k++];

    printf("El texto en la tira queda de la siguiente manera:\n");
    for (i=0 ; i<col ; i++)
        for (j=0 ; j<ren ; j++)

            printf("%c", escitala[j][i]);

    printf("\n");
}

void descifrarMensaje(){
    int ren, col, i, j, k=0;
    printf("Ingresar el tamaño de la escítala:\n");
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);

    char escitala[ren][col];
    char texto[ren*col];

    printf("Escriba el texto a descifrar:\n");
    scanf("%s", texto);

    for (i=0 ; i<col ; i++)
        for (j=0 ; j<ren ; j++)
            escitala[j][i] = texto[k++];

```

```

printf("El texto descifrado es:\n");
for (i=0 ; i<ren ; i++)
for (j=0 ; j<col ; j++)
printf("%c", escitala[i][j]);
}

```

Codigo en Notepad+

```

1 #include<stdio.h> //directivas del preprocesador
2 /*
3  Programa que realiza la implementación de la escitala espartana
4  Para cifrar y descifrar.
5 */
6
7 void crearMensaje(); // Definimos las funciones principales
8 void descifrarMensaje();
9 int main() { //
10     short opcion=0;
11
12     while (!){
13         printf("\n\t*** ESCRITALA ESPARTANA ***\n"); //Nos mostrara un mensaje de acuerdo a lo que se desea hacer
14         printf("¿Qué desea realizar?\n"); // los printf nos ayudaran para mostrar mensajes secuenciados
15         printf("1) Crear mensaje cifrado.\n");
16         printf("2) Descifrar mensaje.\n");
17         printf("3) Salir.\n");
18         scanf("%d", &opcion); //Nos permitira elegir una opción y ejecutarla
19         switch(opcion){ // Nos ayudara para abrir un menú y selección de función
20             case 1:
21                 crearMensaje();
22                 break;
23             case 2:
24                 descifrarMensaje();
25                 break;
26             case 3:
27                 return 0;
28             default: //En caso de tener un caracter invalido
29                 printf("Opción no válida.\n");
30         }
31     }
32     return 0; //Fin del proceso
33 }
34 void crearMensaje(){ //Dara el paso a la función llamada "Crear mensaje"
35     int ren, col, i, j, k=0; //Denomina las variables
36     printf("Ingresar el tamaño de la escitala:\n");
37     printf("\nRenglones:");
38     scanf("%i",&ren);
39     printf("\nColumnas:");
40     scanf("%i",&col);
41     char escitala[ren][col]; //Se da el arreglo
42     char texto[ren*col];
43
44     printf("Escriba el texto a cifrar:\n");
45     scanf("%s", texto); //Permitira meter el mensaje para ser codificado
46
47     for (i=0 ; i<ren ; i++){
48         for (j=0 ; j<col ; j++){
49             escitala[i][j] = texto[k++];
50         }
51     }
52     printf("El texto en la tira queda de la siguiente manera:\n");
53     for (i=0 ; i<col ; i++){
54         for (j=0 ; j<ren ; j++){
55             printf("%c", escitala[j][i]);
56         }
57     }
58     printf("\n");
59 }
60 void descifrarMensaje(){
61     int ren, col, i, j, k=0; //Declaración de las variables
62     printf("Ingresar el tamaño de la escitala:\n");
63     printf("\nRenglones:");
64     scanf("%i",&ren); //Permite escribir para guardar los datos y ocuparlos despues
65     printf("\nColumnas:");
66     scanf("%i",&col);
67
68     char escitala[ren][col]; // Declaramos el arreglo
69     char texto[ren*col]; //La función de los arreglos

```

```

59 void descifrarMensaje(){
60     int ren, col, i, j, k=0; //Declaración de las variables
61     printf("Ingresar el tamaño de la escitala:\n");
62     printf("\nRenglones:");
63     scanf("%i",&ren); //Permite escribir para guardar los datos y ocuparlos despues
64     printf("\nColumnas:");
65     scanf("%i",&col);
66
67     char escitala[ren][col]; // Declaramos el arreglo
68     char texto[ren*col]; //La función de los arreglos
69
70     printf("Escriba el texto a descifrar:\n");
71     scanf("%s", texto); //Escribir el mensaje
72
73     for (i=0 ; i<col ; i++)
74         for (j=0 ; j<ren ; j++)
75             escitala[j][i] = texto[k++]; //Ordenaran el mensaje en el arreglo
76
77     printf("El texto descifrado es:\n");
78     for (i=0 ; i<ren ; i++)
79         for (j=0 ; j<col ; j++)
80             printf("%c", escitala[i][j]); // Mostrara el mensaje final
81 } // Fin del programa

```

```

*** ESCITALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
1
Ingresar el tamaño de la escitala:

Renglones:3

Columnas:5
Escriba el texto a cifrar:
mellamomaria
El texto en la tira queda de la siguiente manera:
mmieoalm la ar

```

```

*** ESCITALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
2
Ingresar el tamaño de la escitala:

Renglones:3

Columnas:5
Escriba el texto a descifrar:
mellamomaria
El texto descifrado es:
mlor eami lmaa

```

Sudoku código

```
#include<stdio.h>

void comenzarjuego();
void salir();
void matriz();
void numero();

int main()
{
int ren, col, i, j, k=0;
int matriz [9][9]=

{{0,1,9,0,0,2,0,0,0},{4,7,0,6,9,0,0,0,1},{0,0,0,4,0,0,0,9,0},{8,9,5,5,0,7,0,0,0},{0,0,0,0,0,0,0,0,0},{0,0,0,
2,0,1,9,5,8}, {0,5,0,0,0,6,9,5,8},{6,0,0,0,2,8,0,7,9},{0,0,0,1,0,0,8,6,0}};
short opcion=0;

while (1){
printf("\n\t*** Sudoku ***\n");
printf("¿Qué desea realizar?\n");
printf("1) Comenzar juego.\n");
printf("2) Obtener la respuesta.\n");
printf("3) Salir.\n");
scanf("%d", &opcion);
switch(opcion){
case 1:
comenzarjuego();
break;
case 2:
return 0;
default:
printf("Opción no válida.\n");
}
}
return 0;

}

void comenzarjuego(){
int matriz[9][9]=
{{0,1,9,0,0,2,0,0,0},{4,7,0,6,9,0,0,0,1},{0,0,0,4,0,0,0,9,0},{8,9,5,5,0,7,0,0,0},{0,0,0,0,0,0,0,0,0},
{0,0,0,2,0,1,9,5,8}, {0,5,0,0,0,6,9,5,8},{6,0,0,0,2,8,0,7,9},{0,0,0,1,0,0,8,6,0}};
```

```

for(int i=0;i<9;i++)
{
    printf("\t");
    for (int j=0;j<9;j++)
    {
        printf("%d",matriz[i][j]);
    }
    printf("\n");
}
int ren, col, i, j, k=0,numero;
printf ("Introduzca unas coordenadas ");
scanf ("%d %d", &ren, &col);
printf("\n");

printf ("Introduce numero");
scanf ("%d", numero);

    numero=matriz[ren][col];
    printf("\nTu progreso\n");
    for (int i=0 ; i<9 ; i++)
    {
        for (int j=0 ; j<9 ; j++)
        {
            printf("%d",matriz[i][j]);
        }
        printf("\n");
    }
}

```

Codigo en NotePad

```

#include<stdio.h>

//Declaracion de funciones
void comenzarjuego();
void salir();
void matriz();
void numero();

int main()
{
    //Declaración de variables
    int ren, col, i, j, k=0;
    //Declaracion de matriz sin resolver
    int matriz [9][9]=
        {{0,1,9,0,0,2,0,0,0},{4,7,0,6,9,0,0,0,1},{0,0,0,4,0,0,0,9,0},{8,9,5,5,0,7,0,0,0},{0,0,0,0,0,0,0,0,0},{0,0,0,2,0,1,9,5,8}, {0,5,0,0,0,6,9,5,8},{6,0,0,0,2,8,0
        ,7,9},{0,0,0,1,0,0,8,6,0}};
    short opcion=0;

    while (1){
        printf("\n\t*** Sudoku ***\n");//Menú del programa
        printf("¿Qué desea realizar?\n");
        printf("1) Comenzar juego.\n");
        printf("2) Obtener la respuesta.\n");
        printf("3) Salir.\n");
        scanf("%d", &opcion);
        //Switch para ejecutar la opcion guardada en opl
        switch(opcion){
            case 1:
                comenzarjuego();
                break;
            case 2:
                return 0;
            default:
                printf("Opción no válida.\n");
        }
    }
    return 0;
}

```

```

//Funcion de comenzar juego
void comenzarjuego(){
    int matriz[9][9]=
    {{0,1,9,0,0,2,0,0,0},{4,7,0,6,9,0,0,0,1},{0,0,0,4,0,0,0,9,0},{8,9,5,5,0,7,0,0,0},{0,0,0,0,0,0,0,0,0},{0,0,0,2,0,1,9,5,8},{0,5,0,0,0,6,9,5,8},{6,0,0,0,2,8,0,7,9},
    {0,0,0,1,0,0,8,6,0}};

    for(int i=0;i<9;i++)
    {
        printf("\t");
        for (int j=0;j<9;j++)
        {
            printf("%d",matriz[i][j]);
        }
        printf("\n");
    }

    int ren, col, i, j, k=0,numero;
    //Pedir la ubicación del numero que quieren poner
    printf ("Introduzca unas coordenadas ");
    scanf ("%d %d", &ren, &col);
    printf("\n");
    //Preguntar numero
    printf ("Introduce numero");
    scanf ("%d", numero);

    numero=matriz[ren][col];
    printf("\nTu progreso\n"); //Mostrar progreso
    for (int i=0 ; i<9 ; i++)
    {
        for (int j=0 ; j<9 ; j++)
        {
            printf("%d",matriz[i][j]);
        }
        printf("\n");
    }
}

```

```

*** Sudoku ***
¿Quiere desea realizar?
1) Comenzar juego.
2) Obtener la respuesta.
3) Salir.
1

```

```

019002000
470690001
000400090
895507000
000000000
000201958
050006958
600028079
000100860

```

Introduzca unas coordenadas 8 9

Introduce numero 6

```

Tu progreso
019002000
470690001
000400090
895507000
000000000
000201958
050006958
600028079
000160860

```


Conclusión:

Al hacer esta práctica note la importancia de los arreglos ya que hacen que nuestro trabajo se vea mucho más ordenado y podamos repetir algunos valores sin necesidad de irlo buscando, pude notar muchos errores que todavía cometemos al programar en C y tuve unos pequeños errores en el programa que fui corrigiendo con los aprendizajes adquiridos el semestre pasado, también pude notar la relación con la materia ya que estos arreglos también nos sirven para almacenar datos lo cual va a ser fundamental al querer guardar los datos.

A lo largo de nuestro día ocupamos los arreglos, aunque no los veamos como es el caso de nosotros como estudiantes de ingeniería utilizamos matrices para la solución de problemas o al momento de crear cuentas en redes sociales que nos permite tener una serie de aparatos en los que podemos llenar con nuestra información y nos dan nombres aleatorios, o al tener una agenda con todos los números y querer llamar, solo introducimos el nombre y nos aparece, también cuando descargamos un juego y nos pide seleccionar nuestra edad o el nivel de dificultad en base a eso se van poniendo más funciones, los maestros podrían utilizar estos arreglos para los números de lista o la base de datos de una empresa de acuerdo con la información de las personas, como vemos son muy importantes ya que nos dan un control y una facilidad del manejo de la información, haciendo que al buscar todo sea más rápido, al mismo tiempo se da un orden en todo lo que hacemos y hace que se ve más limpio y bonito de trabajar o analizar.

Referencias:

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- Laboratorio Salas A y B. (s. f.). Laboratorio de Computación Salas A y B. Recuperado el 13 de marzo del 2021, de <http://lcp02.fi-b.unam.mx>
- Publicado por Ing. Hugo Delgado. (2020, 21 noviembre). Arrays, arreglos, cadenas o vectores en C - Ejemplos y uso. Recuperado 14 de marzo de 2021, de <https://disenowebakus.net/arrays.php>
- Arreglos Multidimensionales - Proyecto de Fundamentos. (s. f.). Recuperado 14 de marzo de 2021, de <https://sites.google.com/site/proyectedefundamentos/metodos/estructuras-de-datos-basicas/arreglos-multidimensionales>