



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): Práctica 1: Aplicación de arreglos

Integrante(s): Miranda González José Francisco

*No. de Equipo de
cómputo empleado:* /

No. de Lista o Brigada: /

Semestre: 2021-2

Fecha de entrega: 15 de Marzo del 2021

Observaciones:

CALIFICACIÓN: _____

Objetivo

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

Introducción

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

A cada elemento del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales.

Arreglos unidimensionales:

Es un tipo de dato estructurado que está formado de una colección finita y ordenada de datos del mismo tipo. Están formados por un conjunto de elementos de un mismo tipo de datos que se almacenan bajo un mismo nombre y se diferencian por la posición que tiene cada elemento dentro del arreglo de datos.

Arreglos multidimensionales

Es un tipo de dato estructurado, que está compuesto por dimensiones. Para hacer referencia a cada componente del arreglo es necesario utilizar “n” índices, uno para cada dimensión. El término dimensión representa el número de índices utilizados para referirse a un elemento particular en el arreglo.

Los arreglos se utilizan para hacer más eficiente el código de un programa.

Desarrollo

Código de la Escitala Espartana

Analizar y realizar comentarios en el código acerca de su funcionamiento.

Adjuntar capturas del código y de su ejecutable.

```

1 // CODIGO (LA ESCITALA ESPARTANA)
2 #include <stdio.h> //libreria
3
4 /* Programa que realiza la implementación de la escitala espartana
5 Para cifrar y descifrar. */
6
7 void crearMensaje(); //se define la funcion
8 void descifrarMensaje(); //se define la funcion
9
10 int main() { //funcion principal main
11     short opcion=0; // se inicializa la variable en 0
12
13     while (1) { //Bucle
14         printf("\n\t*** ESCITALA ESPARTANA ***\n\n");
15         printf("¿Que desea realizar?\n");
16         printf("1) Crear mensaje cifrado.\n");
17         printf("2) Descifrar mensaje.\n");
18         printf("3) Salir.\n");
19         printf("\n");
20         scanf("%d", &opcion); //agrega un valor a la variable opcion
21     }
22 }

```

```

22 printf("\n");
23 switch(opcion) { //segun la variable "opcion" se realizaran las siguientes acciones:
24     case 1: //si opcion es 1:
25         crearMensaje(); //manda a llamar a la funcion crearMensaje
26         break;
27     case 2: //si opcion es 2:
28         descifrarMensaje(); //manda a llamar a la funcion descifrarMensaje
29         break;
30     case 3: //si opcion es 3 saldra del programa
31         return 0;
32     default: // si opcion no es 1,2 o 3 imprimira por pantalla:
33         printf("Opcion no valida.\n");
34 }
35 }
36 return 0;
37 }
38 void crearMensaje() { // lo que se realice en la funcion se ejecutara en la principal "main"
39     int ren, col, i, j, k=0; //declaracion de variables e inicializacion de k en 0
40     printf("Ingresar el tamaño de la escitala:\n");
41     printf("\nRenGlones:");
42     scanf("%i", &ren); //asigna un valor a la variable ren (renGlones)

```

```

43 printf("\nColumnas:");
44 scanf("%i", &col); //asigna un valor a la variable col (columnas)
45
46 char escitala[ren][col]; //creacion de la matriz
47 char texto[ren*col];
48
49 printf("Escriba el texto a cifrar:\n");
50 scanf("%s", texto); //almacena el texto a escribir
51
52 for (i=0 ; i<ren ; i++){ //Bucle for definido por las filas y columnas
53     for (j=0 ; j<col ; j++){
54         escitala[i][j] = texto[k++];
55     }
56 }
57 printf("El texto en la tira queda de la siguiente manera:\n");
58 for (i=0 ; i<col ; i++){ //Bucle for para imprimir el texto
59     for (j=0 ; j<ren ; j++){
60         printf("%c", escitala[j][i]);
61     }
62 }
63 printf("\n");

```

```
Start here x Escitala.c x Sudoku.c x
64 }
65
66 void descifrarMensaje(){ // lo que se realiza en la funcion se ejecutara en la principal "main"
67     int ren, col, i, j, k=0; //declaracion de variables e inicializacion de k en 0
68     printf("Ingresar el tamaño de la escitala:\n");
69     printf("\nRenglones:");
70     scanf("%i",&ren); //asigna un valor a la variable ren (renglones)
71     printf("\nColumnas:");
72     scanf("%i",&col); //asigna un valor a la variable col (columnas)
73
74     char escitala[ren][col]; //creacion de la matriz
75     char texto[ren*col];
76
77     printf("Escriba el texto a descifrar:\n");
78     scanf("%s", texto); //almacena el texto a escribir
79
80     for (i=0 ; i<col ; i++){ //bucle for definida por las filas y columnas
81         for (j=0 ; j<ren ; j++){
82             escitala[j][i] = texto[k++];
83         }
84     }
85
86     printf("El texto descifrado es:\n");
87     for (i=0 ; i<ren ; i++){ //Bucle for para imprimir el texto
88         for (j=0 ; j<col ; j++){
89             printf("%c", escitala[i][j]);
90         }
91     }
92 }
```

```
C:\Users\manolo\Desktop\Escitala.exe
*** ESCITALA ESPARTANA ***
¿Que desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
1
Ingresar el tamaño de la escitala:
Renglones: 3
Columnas: 3
Escriba el texto a cifrar:
HolaProfe
El texto en la tira queda de la siguiente manera:
HaaooPflre
*** ESCITALA ESPARTANA ***
¿Que desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
2
Ingresar el tamaño de la escitala:
Renglones: 3
Columnas: 3
Escriba el texto a descifrar:
HaaooPflre
El texto descifrado es:
HolaProfe
*** ESCITALA ESPARTANA ***
¿Que desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
3
Process returned 0 (0x0)   execution time : 93.122 s
Press any key to continue.
```

Código del Sudoku

Realizar y resolver un sudoku

Adjuntar capturas del código y de su ejecutable

```
Start here x Sudoku.c x
1  #include <stdio.h>
2
3  int main() {
4      printf("\n\t\tSUDOKU\n\n");
5      int Sudoku[9][9]={3,0,0,0,0,0,5,4,0,
6                          4,8,6,1,3,0,0,0,0,
7                          0,0,0,0,7,6,0,3,1,
8                          0,3,2,0,0,8,0,0,4,
9                          0,6,0,0,0,0,0,2,0,
10                         0,1,0,7,2,3,0,0,0,
11                         9,4,3,0,0,0,2,1,0,
12                         6,0,7,0,0,0,0,8,0,
13                         0,0,0,0,4,2,7,6,9};
14
15      int i,j;
16      for(i=0;i<9;i++){
17          for(j=0;j<9;j++){
18              printf("[%d] ",Sudoku[i][j]);
19          }
20          printf("\n\n");
21      }
22      printf("Complete los espacios con 0 del sudoku\n\n");
```

```
Start here x Sudoku.c x
22
23      int x;
24      printf("posicion [1,2] :"); scanf("%d",&x); Sudoku[0][1]=x;
25      printf("posicion [1,3] :"); scanf("%d",&x); Sudoku[0][2]=x;
26      printf("posicion [1,4] :"); scanf("%d",&x); Sudoku[0][3]=x;
27      printf("posicion [1,5] :"); scanf("%d",&x); Sudoku[0][4]=x;
28      printf("posicion [1,6] :"); scanf("%d",&x); Sudoku[0][5]=x;
29      printf("posicion [1,9] :"); scanf("%d",&x); Sudoku[0][8]=x;
30      printf("\n");
31      printf("posicion [2,6] :"); scanf("%d",&x); Sudoku[1][5]=x;
32      printf("posicion [2,7] :"); scanf("%d",&x); Sudoku[1][6]=x;
33      printf("posicion [2,8] :"); scanf("%d",&x); Sudoku[1][7]=x;
34      printf("posicion [2,9] :"); scanf("%d",&x); Sudoku[1][8]=x;
35      printf("\n");
36      printf("posicion [3,1] :"); scanf("%d",&x); Sudoku[2][0]=x;
37      printf("posicion [3,2] :"); scanf("%d",&x); Sudoku[2][1]=x;
38      printf("posicion [3,3] :"); scanf("%d",&x); Sudoku[2][2]=x;
39      printf("posicion [3,4] :"); scanf("%d",&x); Sudoku[2][3]=x;
40      printf("posicion [3,7] :"); scanf("%d",&x); Sudoku[2][6]=x;
41      printf("\n");
42
```

```
Start here x Sudoku.c x
43     for(i=0;i<9;i++){
44         for(j=0;j<9;j++){
45             printf("[%d] ",Sudoku[i][j]);
46         }
47         printf("\n\n");
48     }
49     printf("\n");
50
51     printf("posicion [4,1] :"); scanf("%d",&x); Sudoku[3][0]=x;
52     printf("posicion [4,4] :"); scanf("%d",&x); Sudoku[3][3]=x;
53     printf("posicion [4,5] :"); scanf("%d",&x); Sudoku[3][4]=x;
54     printf("posicion [4,7] :"); scanf("%d",&x); Sudoku[3][6]=x;
55     printf("posicion [4,8] :"); scanf("%d",&x); Sudoku[3][7]=x;
56     printf("\n");
57     printf("posicion [5,1] :"); scanf("%d",&x); Sudoku[4][0]=x;
58     printf("posicion [5,3] :"); scanf("%d",&x); Sudoku[4][2]=x;
59     printf("posicion [5,4] :"); scanf("%d",&x); Sudoku[4][3]=x;
60     printf("posicion [5,5] :"); scanf("%d",&x); Sudoku[4][4]=x;
61     printf("posicion [5,6] :"); scanf("%d",&x); Sudoku[4][5]=x;
62     printf("posicion [5,7] :"); scanf("%d",&x); Sudoku[4][6]=x;
63     printf("posicion [5,9] :"); scanf("%d",&x); Sudoku[4][8]=x;
```

```
Start here x Sudoku.c x
64     printf("\n");
65     printf("posicion [6,1] :"); scanf("%d",&x); Sudoku[5][0]=x;
66     printf("posicion [6,3] :"); scanf("%d",&x); Sudoku[5][2]=x;
67     printf("posicion [6,7] :"); scanf("%d",&x); Sudoku[5][6]=x;
68     printf("posicion [6,8] :"); scanf("%d",&x); Sudoku[5][7]=x;
69     printf("posicion [6,9] :"); scanf("%d",&x); Sudoku[5][8]=x;
70     printf("\n");
71
72     for(i=0;i<9;i++){
73         for(j=0;j<9;j++){
74             printf("[%d] ",Sudoku[i][j]);
75         }
76         printf("\n\n");
77     }
78     printf("\n");
79
80     printf("posicion [7,4] :"); scanf("%d",&x); Sudoku[6][3]=x;
81     printf("posicion [7,5] :"); scanf("%d",&x); Sudoku[6][4]=x;
82     printf("posicion [7,6] :"); scanf("%d",&x); Sudoku[6][5]=x;
83     printf("posicion [7,9] :"); scanf("%d",&x); Sudoku[6][8]=x;
84     printf("\n");
```

```
Start here x Sudoku.c x
85     printf("posicion [8,2] :"); scanf("%d",&x); Sudoku[7][1]=x;
86     printf("posicion [8,4] :"); scanf("%d",&x); Sudoku[7][3]=x;
87     printf("posicion [8,5] :"); scanf("%d",&x); Sudoku[7][4]=x;
88     printf("posicion [8,6] :"); scanf("%d",&x); Sudoku[7][5]=x;
89     printf("posicion [8,7] :"); scanf("%d",&x); Sudoku[7][6]=x;
90     printf("posicion [8,9] :"); scanf("%d",&x); Sudoku[7][8]=x;
91     printf("\n");
92     printf("posicion [9,1] :"); scanf("%d",&x); Sudoku[8][0]=x;
93     printf("posicion [9,2] :"); scanf("%d",&x); Sudoku[8][1]=x;
94     printf("posicion [9,3] :"); scanf("%d",&x); Sudoku[8][2]=x;
95     printf("posicion [9,4] :"); scanf("%d",&x); Sudoku[8][3]=x;
96     printf("\n");
97
98     printf("SUDOKU RESULETO!!\n\n");
99
100    for(i=0;i<9;i++){
101        for(j=0;j<9;j++){
102            printf("[%d] ",Sudoku[i][j]);
103        }
104        printf("\n\n");
105    }
```

```

106
107
108
109
}
system("pause");
return 0;

```

```

C:\Users\manolo\Desktop\Sudoku.exe

SUDOKU

[3] [0] [0] [0] [0] [0] [5] [4] [0]
[4] [8] [6] [1] [3] [0] [0] [0] [0]
[0] [0] [0] [0] [7] [6] [0] [3] [1]
[0] [3] [2] [0] [0] [8] [0] [0] [4]
[0] [6] [0] [0] [0] [0] [0] [2] [0]
[0] [1] [0] [7] [2] [3] [0] [0] [0]
[9] [4] [3] [0] [0] [0] [2] [1] [0]
[6] [0] [7] [0] [0] [0] [0] [8] [0]
[0] [0] [0] [0] [4] [2] [7] [6] [9]

Complete los espacios con 0 del sudoku

posicion [1,2] :7
posicion [1,3] :1
posicion [1,4] :2
posicion [1,5] :8
posicion [1,6] :9
posicion [1,9] :6

posicion [2,6] :5
posicion [2,7] :9
posicion [2,8] :7
posicion [2,9] :2

posicion [3,1] :2
posicion [3,2] :9
posicion [3,3] :5
posicion [3,4] :4
posicion [3,7] :8

[3] [7] [1] [2] [8] [9] [5] [4] [6]
[4] [8] [6] [1] [3] [5] [9] [7] [2]
[2] [9] [5] [4] [7] [6] [8] [3] [1]
[0] [3] [2] [0] [0] [8] [0] [0] [4]
[0] [6] [0] [0] [0] [0] [0] [2] [0]
[0] [1] [0] [7] [2] [3] [0] [0] [0]
[9] [4] [3] [0] [0] [0] [2] [1] [0]
[6] [0] [7] [0] [0] [0] [0] [8] [0]

```

```

C:\Users\manolo\Desktop\Sudoku.exe

[0] [0] [0] [0] [4] [2] [7] [6] [9]

posicion [4,1] :7
posicion [4,4] :6
posicion [4,5] :9
posicion [4,7] :1
posicion [4,8] :5

posicion [5,1] :8
posicion [5,3] :9
posicion [5,4] :5
posicion [5,5] :1
posicion [5,6] :4
posicion [5,7] :3
posicion [5,9] :7

posicion [6,1] :5
posicion [6,3] :4
posicion [6,7] :6
posicion [6,8] :9
posicion [6,9] :8

[3] [7] [1] [2] [8] [9] [5] [4] [6]
[4] [8] [6] [1] [3] [5] [9] [7] [2]
[2] [9] [5] [4] [7] [6] [8] [3] [1]
[7] [3] [2] [6] [9] [8] [1] [5] [4]
[0] [6] [9] [5] [1] [4] [3] [2] [7]
[5] [1] [4] [7] [2] [3] [6] [9] [8]
[9] [4] [3] [0] [0] [0] [2] [1] [0]
[6] [0] [7] [0] [0] [0] [0] [8] [0]
[0] [0] [0] [0] [4] [2] [7] [6] [9]

posicion [7,4] :8
posicion [7,5] :6
posicion [7,6] :7
posicion [7,9] :5

posicion [8,2] :2
posicion [8,4] :9
posicion [8,5] :5
posicion [8,6] :1
posicion [8,7] :4
posicion [8,9] :3

posicion [9,1] :1
posicion [9,2] :5

```

```
C:\Users\manolo\Desktop\Sudoku.exe
posicion [8,2] :2
posicion [8,4] :9
posicion [8,5] :5
posicion [8,6] :1
posicion [8,7] :4
posicion [8,9] :3

posicion [9,1] :1
posicion [9,2] :5
posicion [9,3] :8
posicion [9,4] :3

SUDOKU RESUELTO!!

[3] [7] [1] [2] [8] [9] [5] [4] [6]
[4] [8] [6] [1] [3] [5] [9] [7] [2]
[2] [9] [5] [4] [7] [6] [8] [3] [1]
[7] [3] [2] [6] [9] [8] [1] [5] [4]
[8] [6] [9] [5] [1] [4] [3] [2] [7]
[5] [1] [4] [7] [2] [3] [6] [9] [8]
[9] [4] [3] [8] [6] [7] [2] [1] [5]
[6] [2] [7] [9] [5] [1] [4] [8] [3]
[1] [5] [8] [3] [4] [2] [7] [6] [9]

Presione una tecla para continuar . . .
```

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

Conclusión

Los arreglos son estructuras muy útiles cuando el objetivo es almacenar varios datos del mismo tipo, además que ayudan a hacer más eficiente el código de un programa.

Estos pueden ser representados a manera de matrices y tienen muchas aplicaciones en diferentes campos.

Actualmente, las matrices son de mucha utilidad en problemas prácticos de la vida diaria. Sobre todo, en aquellos que involucran Sistemas de Ecuaciones Lineales.

Para el campo de la informática sus usos pueden ser: resolver circuitos eléctricos, resolver sistemas de ecuaciones, analizar fallas en telecomunicaciones, almacenamiento de información óptima en sistemas, ayuda para graficar funciones cruzadas, analizar redes eléctricas, modelar sistemas entre otros.

Un uso claro que vimos en esta práctica fue la encriptación de códigos, mediante diferentes funciones el programa es capaz de hacer que un código introducido por teclado pueda quedar cifrado y a su vez es posible descifrarlo.

Bibliografía

<https://quehacerenbenidorm.com/wp-content/uploads/2019/03/Sudoku-nivel-2-RESUELTO.jpg>

<https://www.monografias.com/docs/Aplicacion-De-Matrices-En-Computacion-F3ZAPHSYMZ#:~:text=Aplicaci%C3%B3n%20de%20las%20matrices%20en,ayuda%20para%20graficar%20funciones%20cruzadas%20%2D>