



## 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 1

*Grupo:* 15

*No de Práctica(s):* #1

*Integrante(s):* RAUDALES PALMA LEONARDO DE JESÚS

*No. de Equipo de  
cómputo empleado:* NO APLICA

*No. de Lista o Brigada:* -

*Semestre:* 2021-2

*Fecha de entrega:* 15 DE MARZO 2021

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## **OBJETIVO:**

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

## **INTRODUCCIÓN:**

Un arreglo es un grupo de datos unidos que son del mismo tipo y este tiene un tamaño determinado a la hora de programarse. Cada uno de estos datos es relacionado con una posición particular dentro del arreglo y para poder acceder a este por medio del Lenguaje C se requiere de un índice donde el arreglo comienza en 0 y termina en  $n-1$  siendo  $n$  el tamaño de la dimensión.

Existen dos tipos de arreglos fundamentales, unidimensionales que como el nombre lo dice solo están conformados con una dimensión, una fila o columna. O los arreglos multidimensionales los cuales están conformados por 2 columnas u filas o más.

También están los arreglos ligados o los arreglos contiguos. El arreglo continuo es aquel que permanece con la misma dimensión de inicio a fin. Mientras que el arreglo ligado se declara bajo demanda y durante la ejecución del programa es posible poder redimensionar el arreglo.

Los arreglos tienen distintas funciones dentro del rubro de la programación y la solución de problema, por ejemplo, la implementación en un sistema de listado quizá tipo inventario o incluso para hacer programas que requieran de esa especificación de espacios de datos particular que ocupan los arreglos como una lista de calificaciones.

Uno de los métodos para codificar mensajes ocupados primeramente fue la escítala espartana que tiene quizá un parecido a un arreglo a la hora de dar un orden a ciertos caracteres de manera que el mensaje tenga sentido solo ubicando estos datos de manera correcta y ordenada dentro de una estructura mayor.

## DESARROLLO:

### Código (la escítala espartana)

```
C:\Users\Leo\Documents\EDA\Lenguaje C\EE.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?

inventario.c  i.c  EE.c

1  #include<stdio.h>
2
3  /*
4   Programa que realiza la implementación de la escítala espartana
5   para cifrar y descifrar.
6  */
7
8  // Declaración de variables para cifrar y descifrar.
9  void crearMensaje();
10 void descifrarMensaje();
11
12 // Función principal.
13 int main()
14
15 {
16     //Declaración de variables tipo caracter acentuadas.
17     char ene = 164, e = 130, si = 168, I = 214, ai = 161, o = 162, a = 160;
18     // Función de opción para el menú.
19     short opcion=0;
20
21     // Menú utilizando while y switch para las opciones.
22     while (1)
23     {
24         short opcion=0;
25         printf("\n\t*** ESCÍTALA ESPARTANA ***\n", I);
26         printf("\n\t¿c desea realizar?\n", si, e);
27         printf("1) Crear mensaje cifrado.\n");
28         printf("2) Descifrar mensaje. \n");
29         printf("3) Salir. \n\n");
30         scanf("%d", &opcion);
31         switch(opcion)
32         {
33             case 1:
34                 crearMensaje();
35                 break;
36             case 2:
37                 descifrarMensaje();
38                 break;
39             case 3:
40                 return 0;
41             default:
42                 printf("Opción no valida.\n", o);
43         }
44     }
45
46
47     return 0;
48 }
49
50
51 // Función que abarca toda la acción de cifrar un texto en escítala espartana.
52 void crearMensaje()
53 {
54     char ene = 164, ai = 161;
55     int ren, col, i, j, k=0;
56     printf("\nIngresar el tamaño de la escítala:\n", ene, ai);
```

```

57     printf("\n\nRenglones:");
58     scanf("%i", &ren);
59     printf("\nColumnas:");
60     scanf("%i", &col);
61
62     char escitala[ren][col];
63     char texto[ren*col];
64
65     printf("\nEscriba el texto a cifrar:");
66     scanf("%s", &texto);
67
68     for (i=0 ; i<ren ; i++)
69         for (j=0 ; j<col ; j++)
70             escitala [i][j] = texto [k++];
71
72     printf("\nEl texto en la tira queda de la siguiente manera:\t");
73     for (i=0 ; i<col ; i++)
74         for (j=0 ; j<ren ; j++)
75             printf("%c", escitala[j][i]);
76
77     printf("\n");
78 }
79
80 // Función que abarca toda la acción de descifrar un texto en escitala espartana.
81 void descifrarMensaje()
82 {
83     char ene = 164, ai = 161;
84     int ren, col, i, j, k=0;
85     printf("\nIngresar el tamaño de la escitala:\t", ene, ai);
86     printf("\n\nRenglones:");
87     scanf("%i", &ren);
88     printf("\nColumnas:");
89     scanf("%i", &col);
90
91     char escitala[ren][col];
92     char texto[ren*col];
93
94     printf("\nEscriba el texto a descifrar:");
95     scanf("%s", texto);
96
97     for (i=0 ; i<col ; i++)
98         for (j=0 ; j<ren ; j++)
99             escitala[j][i] = texto[k++];
100
101     for (i=0 ; i<ren ; i++)
102         for (j=0 ; j<col ; j++)
103             printf("%c", escitala[i][j]);
104 }

```

## Compilación (la escítala espartana)

```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\EDA I\Lenguaje C>gcc EE.c -o EE.exe

C:\Users\Leo\Documents\EDA I\Lenguaje C>EE.exe

*** ESCÍTALA ESPARTANA ***

¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.

1

Ingresar el tamaño de la escítala:

Renglones:2

Columnas:2

Escriba el texto a cifrar:hola

El texto en la tira queda de la siguiente manera:      hloa

*** ESCÍTALA ESPARTANA ***

¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.

2

Ingresar el tamaño de la escítala:

Renglones:2

Columnas:2

Escriba el texto a descifrar:hloa
hola

*** ESCÍTALA ESPARTANA ***

¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.

4

Opción no valida.

*** ESCÍTALA ESPARTANA ***

¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.

3

C:\Users\Leo\Documents\EDA I\Lenguaje C>_
```

## Código (Actividad SODUKU)

```
C:\Users\Leo\Documents\EDA I\Lenguaje C\sudoku.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins

inventario.c x  i.c x  EE.c x  sudoku.c x  sudokub.c x

1  #include<stdio.h>
2
3  // Declarando las variables para resolver y mostrar la solución del Sudoku.
4  void resolver();
5  void solucion();
6
7  // Función principal.
8  int main(){
9
10     //Declaración de variables tipo caracter .
11     char o=162, log = 219, e = 130, si = 168, sel = 33, se2 = 173;
12     //Variable tipo opción para el menú.
13     short selec=0;
14
15     // Menú principal.
16     printf("\n\t%c SUPER SUDOKU :D %c\n", log, log);
17     printf("\n\t%c¿Qué quieres hacer? 0u0\n", si, e);
18     printf("(1) Resolver Sudoku.\n");
19     printf("(2) Quiero ver la solución.\n", o);
20     printf("(3) Salir.\n\n");
21     scanf("%d", &selec);
22     switch(selec)
23     {
24         case 1:
25             resolver();
26             break;
27         case 2:
28             solucion();
29             main();
30             break;
31         case 3:
32             printf("\n\t!Vuelve pronto! :D\n");
33             return 0;
34
35         default:
36             printf("Opción no válida.\n", o);
37     }
38
39     return 0;
40 }
41
42 // Función que ejecuta la acción de solución del sudoku
43 void resolver()
44 {
45     char efec = 178, si = 168, u = 163, o=162, e = 130;
46     int *ren, *col, re=5, valor=9;
47     int opcion=0, op;
48
49
50
51
52     int i,j,matrix[9][9] = {{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},
53     {0,6,0,0,0,0,2,0},{0,1,0,7,2,3,0,0,0},{9,4,3,0,0,2,1,0},{6,0,7,0,0,0,0,8,0},{0,0,0,4,2,7,6}};
54     int res[9][9] = {{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},
55     {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}};
56
57     ren = &i;
58     col = &j;
59
60
61     printf("\t%cSudoku%c\n", efec, efec);
62     for (i=0 ; i<9 ; i++){
63         for (j=0 ; j<9 ; j++){
64             printf(" %d ", matrix[i][j]);
65         }
66         printf("\n");
67     }
68 }
```

```

69
70 while (opcion==1 || op==1 || matriz != res )
71 {
72
73     char efec = 178;
74     printf("\n\tCon qu c n mero quieres intentar? :)\n\n", si, e, u);
75     scanf("%i", &valor);
76
77
78     printf("\n\tEn qu c rengl n del 0 al 8 lo ponemos?\n", si, e, o);
79     scanf("%i", &ren);
80     printf("\n\tEn qu c columna de la 0 a la 8 lo ponemos?\n", si, e);
81     scanf("%i", &col);
82
83     printf("\nSe sustituy  el %i que estaba antes\npor el %i que ingresaste :D \n", o, matriz[i][j], valor);
84
85     matriz[i][j]= valor;
86     printf("\n\t cSudoku c\n\n", efec, efec);
87     for (i=0 ; i<9 ; i++){
88         for (j=0 ; j<9 ; j++){
89             printf(" %i ", matriz[i][j]);
90         }
91
92         printf("\n");
93     }
94
95     printf("\n\tDeseas continuar? 0u0\n1) Seguir \n2) Terminar \n", si);
96     scanf("%d", &op);
97
98     if (op==2)
99     {main();
100     break;}
101 }
102
103
104
105 // Funci n que realiza la acci n de mostrar la respuesta correcta al jugador.
106 void solucion()
107 {
108     char o = 162;
109     int res[9][9] = {{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},
110     {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}};
111     int i, j;
112     printf("\n Compara tus resultados con la soluci n :)\n\n\t", o);
113     for (i=0 ; i<9 ; i++){
114         for (j=0 ; j<9 ; j++){
115             printf(" %d ", res[i][j]);
116         }
117         printf("\n");
118     }
119
120
121
122

```

## Compilación (Actividad SODUKU)

```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\EDA I\Lenguaje C>gcc sudoku.c -o sudoku.exe

C:\Users\Leo\Documents\EDA I\Lenguaje C>sudoku.exe

  ■ SUPER SUDOKU :D ■

¿Qué quieres hacer? 0u0

1) Resolver Sudoku.
2) Quiero ver la solución.
3) Salir.

1

  ■ 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 0

¿Con qué número quieres intentar? :)

2

¿En qué renglón del 0 al 8 lo ponemos? 3

¿En qué columna de la 0 a la 8 lo ponemos? 4

Se substituyó el 0 que estaba antes
por el 2 que ingresaste :D

  ■ 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 2 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 0

¿Deseas continuar? 0u0
```



¿Deseas continuar? 0u0

- 1) Seguir
  - 2) Terminar
- 2

■ SUPER SUDOKU :D ■

¿Qué quieres hacer? 0u0

- 1) Resolver Sudoku.
  - 2) Quiero ver la solución.
  - 3) Salir.
- 2

Compara tus resultados con la solución ;)

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	0

■ SUPER SUDOKU :D ■

¿Qué quieres hacer? 0u0

- 1) Resolver Sudoku.
  - 2) Quiero ver la solución.
  - 3) Salir.
- 3

!Vuelve pronto! :D

C:\Users\Leo\Documents\EDA I\Lenguaje C>\_

## CONCLUSIÓN:

Los arreglos son muy importantes debido a su funcionalidad dentro del rubro de la programación, como se vio en los ejemplos se pueden ejecutar de una manera ordenada y esto es muy útil por ejemplo a la hora de programar juegos como el ajedrez o quizá un típico de batalla naval porque requieren de datos en espacios con un orden específico. Sin duda la necesidad de aprender un arreglo debido a que es un conocimiento básico en el estudio es algo que ha quedado demostrado durante la ejecución de la práctica.

## REFERENCIAS UTILIZADAS:

- ❖ Curso Básico de C - Arreglos  
<https://youtu.be/GOSOaNBqZaw>
- ❖ Sudoku Solver in C - Coding Challenge #7.1 - CarlinoGonzalez  
[https://youtu.be/f\\_5FgfvHw30](https://youtu.be/f_5FgfvHw30)