

	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): Rosario Vázquez José André

No. de Equipo de cómputo empleado: -

No. de Lista o Brigada: -

Semestre: 2021 - 2

Fecha de entrega: 15 de marzo del 2021

Observaciones:

CALIFICACIÓN:

Aplicaciones de arreglos

Objetivo.

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

Actividades:

- Crear arreglos unidimensionales.
- Crear arreglos multidimensionales.
- Realizar un sudoku.

Introducción.

Un arreglo se puede definir como un conjunto de variables que se citan y manejan con un mismo nombre, y que permite además la utilización individual de sus elementos.

En realidad funciona como cualquier variable cualquiera , excepto que en lugar de almacenar un solo valor, guarda algunos valores. Los arreglos pueden representarse en varias dimensiones: Unidimensionales, Bidimensionales, hasta multidimensionales.

Resolución.

Código de la práctica.

Notepad ++

```

1  #include<stdio.h>
2  /*
3   Programa que realiza la implementación de la escitala
4   Para cifrar y descifrar.
5  */
6  void crearMensaje();
7  void descifrarMensaje();
8  int main(){
9      short opcion=0;
10     while (1){
11         printf("\n\t*** ESCÍTALA ESPARTANA ***\n");
12         printf("¿Qué desea realizar?\n");
13         printf("1) Crear mensaje cifrado.\n");
14         printf("2) Descifrar mensaje.\n");
15         printf("3) Salir.\n");
16         scanf("%d", &opcion);
17         switch(opcion){
18             case 1:
19                 crearMensaje();
20                 break;
21             case 2:
22                 descifrarMensaje();
23                 break;
24             case 3:
25                 return 0;
26             default:
27                 printf("Opción no válida.\n");
28         }
29     }
30     return 0;
31 }
32 void crearMensaje(){
33     int ren, col, i, j, k=0;
34     printf("Ingresar el tamaño de la escitala:\n");
35     printf("\nRenglones:");
36     scanf("%i",&ren);
37     printf("\nColumnas:");
38     scanf("%i",&col);

```

```

34     printf("Ingresar el tamaño de la escitala:\n");
35     printf("\nRenglones:");
36     scanf("%i",&ren);
37     printf("\nColumnas:");
38     scanf("%i",&col);
39     char escitala[ren][col];
40     char texto[ren*col];
41     printf("Escriba el texto a cifrar:\n");
42     scanf("%s", texto);
43     for (i=0 ; i<ren ; i++)
44         for (j=0 ; j<col ; j++)
45             escitala[i][j] = texto[k++];
46     printf("El texto en la tira queda de la siguiente manera:\n");
47     for (i=0 ; i<col ; i++)
48         for (j=0 ; j<ren ; j++)
49             printf("%c", escitala[j][i]);
50     printf("\n");
51 }
52 void descifrarMensaje(){
53     int ren, col, i, j, k=0;
54     printf("Ingresar el tamaño de la escitala:\n");
55     printf("\nRenglones:");
56     scanf("%i",&ren);
57     printf("\nColumnas:");
58     scanf("%i",&col);
59     char escitala[ren][col];
60     char texto[ren*col];
61     printf("Escriba el texto a descifrar:\n");
62     scanf("%s", texto);
63     for (i=0 ; i<col ; i++)
64         for (j=0 ; j<ren ; j++)
65             escitala[j][i] = texto[k++];
66     printf("El texto descifrado es:\n");
67     for (i=0 ; i<ren ; i++)
68         for (j=0 ; j<col ; j++)
69             printf("%c", escitala[i][j]);
70 }

```

Demostrado en el sistema.

```
Renglones:4
Columnas:1
Escriba el texto a cifrar:
htyuiolphytrcdsadvghjkknmhgfdsa
El texto en la tira queda de la siguiente manera:
htyu

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

Renglones:1
Columnas:4
Escriba el texto a descifrar:
htyu
El texto descifrado es:
htyu

*** ESCITALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
```

Comentario.

El código presenta la estructura switch para poder presentar un menú en condiciones. También posee estructuras de repetición como while y for para repetir la veces que quieras el menú, para descifrar y cifrar un mensaje. El arreglo que se presenta es uno bidimensional. Estos para representar las filas y columnas, donde es fundamental descifrar y cifrar los mensajes. Por último posee funciones que son importantes para calcular el problema independientemente del programa.

Sudoku con arreglos.

Parte del código en Notepad++ (es muy largo)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int tabla[9][9];
6  int dificultad;
7
8  int main()
9  {
10     void crearSudoku();
11
12     printf("Generador de sudokus\nhttp://opengl-esp.superforo.net - por HarZe\n\n");
13     scanf("%i",&dificultad);
14     if (!dificultad) dificultad = 40;
15     else if (dificultad<5) dificultad = 5;
16     else if (dificultad>75) dificultad = 75;
17     printf("\nDificultad seleccionada: %i\nSudoku:\n",dificultad);
18
19     crearSudoku();
20
21     system("PAUSE");
22     return 0;
23 }
24
25 void crearSudoku()
26 {
27     int i,j,aux;
28     srand(time(0)); /*Establece que el origen de los numeros rand() seran el n
29     tabla[0][0] = (rand()%(9))+1; /*Para obtener un entero entre 0 y 8, sacamos el n
30     do aux = (rand()%(9)) + 1;
31     while(aux == tabla[0][0]); /*Se busca un numero aleatorio que NO sea igual
32     tabla[0][1] = aux;
33     do aux = (rand()%(9))+1;
34     while(aux == tabla[0][0] || aux == tabla[0][1]);/*Se busca un numero aleat
35     tabla[0][2] = aux;
36     do aux = (rand()%(9))+1;
37     while(aux == tabla[0][0] || aux == tabla[0][1] || aux == tabla[0][2]);

```

Demostrado en el sistema.

```

Introduce una dificultad (5-75):6

Dificultad seleccionada: 6
Sudoku:
5#7      831      629
629      547      831
83#      629      547

75#      1#3      962
962      75#      1#3
#8#      962      754

475      #18      2#6
296      475      318
3#8      2#6      475

Presione una tecla para continuar . . .

C:\Users\andre\OneDrive\Escritorio\Lenguaje C>

```

```
C:\Users\andre\OneDrive\Escritorio\Lenguaje C>sudokuc.exe
Generador de sudokus

Introduce una dificultad (5-75):54
Dificultad seleccionada: 54
Sudoku:
##5  ##  ##3
#2#  #7#  ###
1##  #23  6##

56#  ##8  392
3##  ###  41#
##8  ##2  ###

7#6  8#1  ###
23#  #5#  8##
###  ###  ###

Presione una tecla para continuar . . .
```

Código comentado.

Para realizar los sudokus me inspire en un trabajo de un blog, donde explican como hacer y te dan una base muy concreta, lamentablemente no entendi como se puede rellenar; pero cree con ayuda del blog y tomado como base un generador de sudokus por medio de arreglos bidimensionales, con estructuras de repetición y de selección, para mostrara la plantilla de sudoku que se genera.

Conclusión.

Los arreglos son sumamente complejos y extensos con una cantidad inmensa de aplicaciones. En lo personal considero que debo trabajar más este apartado ya que me cuesta todavía comprenderlo al 100%; pero ahora gracias a la práctica pude mejorar mi comprensión con los arreglos.

Los arreglos que se me ocurren son muchos bidimensionales como por ejemplo:

- Un calendario posee la estructura de un arreglo bidimensional donde uno es capaz de comprender el día y año.
- El tablero del ajedrez es otro arreglo bidimensional, ya que posee filas y columnas.
- Las matrices son arreglos bidimensionales debido a su estructura y que son fundamentales en las matemáticas.

Referencias.

Guanilo, M. P. H. (2010, 23 julio). *Sudoku en C*. Taringa!

https://www.taringa.net/+hazlo_tu_mismo/sudoku-en-c_12yli0

Rancel, M. R. (2014, 15 septiembre). Arreglos unidimensionales y bidimensionales. Aprender a programar.

[https://www.aprenderaprogramar.com/index.php?option=com_content
&view=article&id=259:concepto-y-ejemplos-de-arrays-o-arreglos-multid
imensionales-bidimensionales-tridimensional-etc-cu00129a&catid=28&
Itemid=59](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=259:concepto-y-ejemplos-de-arrays-o-arreglos-multidimensionales-bidimensionales-tridimensional-etc-cu00129a&catid=28&Itemid=59)