

Objetivo

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

Introducción

Un arreglo es un conglomerado de información del mismo tipo, guardada de forma contigua y cuya dimensión debe ser definida al momento de su declaración. Los arreglos pueden ser uni, di, tri o hasta multidimensionales. Una característica de los arreglos es la posición dada a cada una de los datos contenidos en el. Para acceder a un dato específico contenido en un arreglo debe utilizarse un índice.

Desarrollo

Código de escítala espartana

```
1  #include<stdio.h>
2  /*
3   Programa que realiza la implementación de la escítala espartana
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*** ESC%cTALA ESPARTANA ***\n", 214);
12 printf("\t¿Qu%c desea realizar?\n", 130);
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 escítala:\n");
35 printf("\nRenglones:");
36 scanf("%i",&ren);
37 printf("\nColumnas:");
38 scanf("%i",&col);
39 char escítala[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 escítala[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", escítala[j][i]);
50 printf("\n");
51 }
52 void descifrarMensaje(){
53 int ren, col, i, j, k=0;
54 printf("Ingresar el tamaño de la escítala:\n");
55 printf("\nRenglones:");
56 scanf("%i",&ren);
57 printf("\nColumnas:");
58 scanf("%i",&col);
59 }
```

```

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 escítala:\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 }
71

```

3

```

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

```

```

*** ESCITALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
1
Ingresar el tamaño de la escítala:
Renglones:2
Columnas:4
Escriba el texto a cifrar:

```

Ejercicio Sudoku

Ya que el programa está formado por 155 líneas de código, adjuntaré como evidencia tanto algunas capturas del programa así como capturas del programa en ejecución.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <windows.h>
4  int arr[9][9];
5  void getsudoku();
6  void getvalues();
7  int main ()
8  {
9      printf("\t\t\tBienvenido a mi sudoku!\n\n");
10     getvalues();
11     getsudoku();
12     printf("\n\n");
13 }
14
15
16 void getvalues()
17 {
18     arr[0][0]=1;
19     arr[0][1]=6;
20     arr[0][2]=2;
21     arr[0][3]=5;
22     arr[0][4]=9;
23     arr[0][5]=8;
24     arr[0][6]=7;
25     arr[0][7]=4;
26     arr[0][8]=3;
27     arr[1][0]=4;
28     arr[1][1]=7;
29     arr[1][2]=9;
30     arr[1][3]=1;
```

```
30     arr[1][3]=1;
31     arr[1][4]=3;
32     arr[1][5]=2;
33     arr[1][6]=6;
34     arr[1][7]=5;
35     arr[1][8]=8;
36     arr[2][0]=5;
37     arr[2][1]=3;
38     arr[2][2]=8;
39     arr[2][3]=7;
40     arr[2][4]=6;
41     arr[2][5]=4;
42     arr[2][6]=2;
43     arr[2][7]=9;
44     arr[2][8]=1;
45     arr[3][0]=9;
46     arr[3][1]=1;
47     arr[3][2]=3;
48     arr[3][3]=4;
49     arr[3][4]=2;
50     arr[3][5]=7;
51     arr[3][6]=5;
52     arr[3][7]=8;
53     arr[3][8]=6;
54     arr[4][0]=6;
55     arr[4][1]=8;
56     arr[4][2]=7;
57     arr[4][3]=9;
58     arr[4][4]=1;
59     arr[4][5]=5;
```

```

100 void getsudoku()
101 {
102     int i, j, r, c, num, f, w;
103     f=0;
104     w=0;
105     do {
106         for(i=0; i<9; i++)
107         {
108             if(i==3 || i==6)
109             {
110                 printf("\n");
111             }
112             for(j=0; j<9; j++)
113             {
114                 if(j==2 || j==5 || j==8)
115                 {
116                     printf("[%d] \t", arr[i][j]);
117                 }
118                 else if((i==0 && j==1)|| (i==0 && j==2)|| (i==0 && j==4)|| (i==0 && j==5)||
119 (i==0 && j==6)|| (i==1 && j==2)|| (i==1 && j==4)|| (i==1 && j==6)
120 || (i==2 && j==0)|| (i==2 && j==1)|| (i==2 && j==2)|| (i==2 && j==3)||
121 (i==2 && j==4)|| (i==2 && j==5)|| (i==3 && j==1)||
122 (i==3 && j==4)|| (i==3 && j==5)|| (i==4 && j==1)|| (i==4 && j==2)||
123 (i==4 && j==3)|| (i==4 && j==4)|| (i==4 && j==6)||
124 (i==5 && j==0)|| (i==5 && j==2)|| (i==5 && j==3)|| (i==5 && j==4)||
125 (i==5 && j==5)|| (i==5 && j==8)|| (i==6 && j==1)|| (i==6 && j==4)||
126 (i==6 && j==5)|| (i==7 && j==0)|| (i==7 && j==4)|| (i==7 && j==7)||
127 (i==8 && j==1)|| (i==8 && j==2)|| (i==8 && j==3)|| (i==8 && j==5)||
128 (i==8 && j==6))//end of if parentheses
129 { //start of if

```

```

126     (i==6 && j==5)|| (i==7 && j==0)|| (i==7 && j==4)|| (i==7 && j==7)||
127     (i==8 && j==1)|| (i==8 && j==2)|| (i==8 && j==3)|| (i==8 && j==5)||
128     (i==8 && j==6))//end of if parentheses
129     { //start of if
130         printf("[ ]\t");
131     } //end of if*/
132     else
133     {
134         printf("[%d]\t", arr[i][j]);
135     }
136     if(j==8)
137     {
138         printf("\n");
139     }
140     } //end j for loop
141 } //end i for loop
142 printf("\n\t\tIngresa una fila (del 0 al 8):\n");
143 scanf("%d", &r);
144 printf("\t\tIngresa una columna (del 0 al 8):\n");
145 scanf("%d", &c);
146 printf("\t\tIngresa n°cmerno:\n", 163);
147 scanf("%d", &num);
148 system("cls");
149 f=1;
150 } while(f=0);
151 printf("\n\t\tCompletaste el sudoku!!\n");
152 Sleep(5000);

```

04. 3

Bienvenido a mi sudoku!

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

Ingresa una fila (del 0 al 8):

04. 3

Bienvenido a mi sudoku!

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

Ingresa una fila (del 0 al 8):

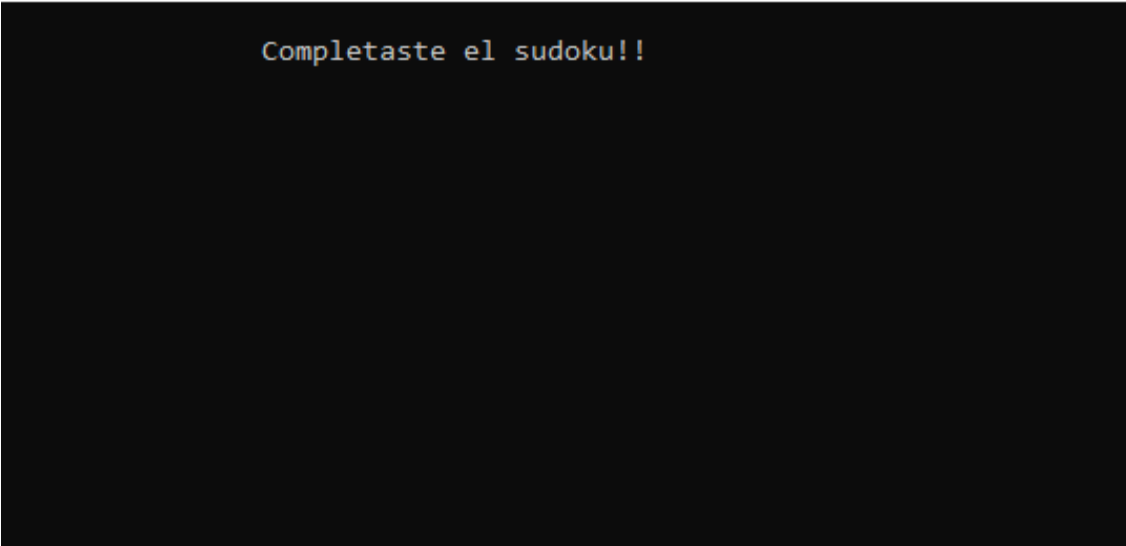
5

Ingresa una columna (del 0 al 8):

6

Ingresa número:

8



```
Completaste el sudoku!!
```

Conclusión

Gracias a esta práctica aprendí tanto características como nuevas aplicaciones de los arreglos que hasta el momento desconocía. El código de la escitala espartana presentado en la presentación me fue de gran utilidad, ya que gracias a él aprecie una aplicación de los arreglos en la vida real que puede ser muy importante (la criptografía).