



## 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):* 01

*Integrante(s):* García Gallegos Luis

*No. de Equipo de  
cómputo empleado:* NA

*No. de Lista o Brigada:* -----

*Semestre:* 2021-2

*Fecha de entrega:* 15/03/2021

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## **Objetivo:**

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

## **Introducción:**

Un arreglo es un tipo de dato estructurado que almacena una sola variable un conjunto limitado de datos o elemento del mismo tipo. Así mismo, es un conjunto de localidades de memoria contiguas donde la dirección más baja corresponde al primer elemento y la dirección más alta al último. El nombre del arreglo apunta a la dirección del primer elemento. Los datos se llaman elementos del arreglo y su posición inicia desde cero por lo tanto el primer elemento se encuentra en la posición  $i-1$ . El tipo de datos que almacena un arreglo puede ser cualquiera.

Un arreglo posee: una lista finita de números de  $n$  elementos del mismo tipo, almacena los elementos en memoria contigua, tener un único nombre de variable que representa a todos los elementos y estos se diferencian por un índice o subíndice.

Arreglos Unidimensionales: es un conjunto de elementos del mismo tipo almacenados en memoria contigua en un vector o lista.

Arreglos Bidimensionales: es un conjunto de elementos del mismo tipo almacenados en memoria contigua en una matriz o tabla. Para acceder a cada elemento se requiere de dos índices o subíndices declarados en dos partes de corchetes, el primero se refiere al tamaño de filas y el segundo al tamaño de columnas.

# Desarrollo:

## *-ESCÍTALA ESPARTANA-*

```
#include<stdio.h>
/*
 Programa que realiza la implementación de la escitala espartana
 Para cifrar y descifrar.
 */
//Funcion para obtener el texto y cifrarlo
void crearMensaje();

//Funcion para descifrar el mensaje
void descifrarMensaje();
int main(){
    short opcion=0;
    while (1){
        //Menu
        printf("\n\t*** ESCITALA ESPARTANA ***\n");
        printf("Que desea realizar?\n");
        printf("1) Crear mensaje cifrado.\n");
        printf("2) Descifrar mensaje.\n");
        printf("3) Salir.\n");
        scanf("%d", &opcion);

        //Te envia a la función correspondiente para realizar lo que el usuario desea
        switch(opcion){
            case 1:
                crearMensaje();
                break;
            case 2:
                descifrarMensaje();
                break;
            case 3:
                return 0;
            default:
                printf("Opción no válida.\n");
        }
    }
    return 0;
}
void crearMensaje(){
```

```
}
void crearMensaje(){
    int ren, col, i, j, k=0;

    //Obtiene los datos para crear dos arreglos uno para el texto(unidimensional) y otro para pasar el texto a una matriz(bidimensional)
    printf("Ingresar el tamaño de la escitala:\n");
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);

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

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

    //Pasa el texto a la matriz
    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");

    //Imprime el texto cifrado
    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;

    //Crea dos arreglos uno para obtener el texto cifrado y otro para pasar el texto a la matriz
    -- INSERT --
```

```

}
void descifrarMensaje(){
    int ren, col, i, j, k=0;

    //Crea dos arreglos uno para obtener el texto cifrado y otro para pasar el texto a la matriz
    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];

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

    //Ingresa el texto cifrado a una matriz
    for (i=0 ; i<col ; i++){
        for (j=0 ; j<ren ; j++){
            escitala[j][i] = texto[k++];
        }
    }
    printf("El texto descifrado es:\n");

    //Imprime el texto descifrado
    for (i=0 ; i<ren ; i++){
        for (j=0 ; j<col ; j++){
            printf("%c", escitala[i][j]);
        }
    }
}

```

\*\*\* ESCITALA ESPARTANA \*\*\*

Que desea realizar?

- 1) Crear mensaje cifrado.
- 2) Descifrar mensaje.
- 3) Salir.

1

Ingresar el tamao de la escotala:

Renglones:6

Columnas:5

Escriba el texto a cifrar:

noolvideospasaralatintoreriaxxx

El texto en la tira queda de la siguiente manera:

niaaltiodsaoaoeatrxlsriexvpanrx

\*\*\* ESCITALA ESPARTANA \*\*\*

Que desea realizar?

- 1) Crear mensaje cifrado.
- 2) Descifrar mensaje.
- 3) Salir.

2

Ingresar el tamao de la escotala:

Renglones:6

Columnas:5

Escriba el texto a descifrar:

niaaltiodsaoaoeotrxlsriexvpanrx

El texto descifrado es:

noolvideospasoralatintoreriaxxx

\*\*\* ESCITALA ESPARTANA \*\*\*

Que desea realizar?

- 1) Crear mensaje cifrado.
- 2) Descifrar mensaje.
- 3) Salir.

3

[root@localhost ~]#

## -SUDOKU-

```
1  #include <iostream>
2  #include <string.h>
3  #include <conio.h>
4
5  using namespace std;
6
7  void asignarBloques(char [9][9]);
8  void dibujarArea(int[9][9]);
9  bool comprobar(int[9][9]);
10 bool buscarElemento(int[9][9],char[9][9],int,int,int,char);
11
12
13 int main()
14 {
15
16     int area[9][9]={0, 0, 6, 0, 0, 9, 0, 0, 0,
17                    8, 3, 5, 4, 0, 1, 6, 0, 2,
18                    0, 2, 7, 0, 3, 6, 1, 5, 4,
19                    6, 5, 0, 7, 4, 0, 8, 0, 9,
20                    2, 0, 4, 0, 0, 0, 7, 0, 0,
21                    0, 0, 0, 0, 8, 3, 0, 4, 5,
22                    0, 6, 2, 0, 9, 0, 0, 7, 0,
23                    5, 7, 0, 0, 6, 0, 0, 3, 0,
24                    4, 1, 8, 3, 5, 0, 9, 2, 6};
25
26     int fila=0,columna=0,valor=0, op=0;
27     char bloques[9][9];
28     int fila=0,columna=0,valor=0, op=0;
29     char bloques[9][9];
30     char idBloque;
31     bool check = true, check1 = false;
32
33     ///ASIGNAMOS BLOQUES
34     asignarBloques(bloques);
35
36     do{
37         ///DIBUJAMOS AREA EN CONSOLA
38         system("cls");
39         dibujarArea(area);
40
41         ///PEDIMOS DATOS DE INGRESO
42         cout << "\n\tQue desea hacer?";
43         cout << "\n\t1) Ingresar valor";
44         cout << "\n\t2) Eliminar valor\n\t";
45         cin >>op;
46
47         switch(op){
48
49             case 1:
50                 ///ASIGNAMOS LOS INDICES AL ARREGLO
51                 cout << "\n\tIngrese los datos para ingresar el elemento";
52                 cout << "\n\tFila: ";
53                 cout << "\n\tIngrese los datos para ingresar el elemento";
54                 cout << "\n\tFila: ";
55                 cin >> fila;
56                 cout << "\n\tColumna: ";
57                 cin >> columna;
58                 ///ASIGNAMOS EL VALOR QUE IRA EN EL ARREGLO
59                 do{
60                     cout << "\n\tIngrese el valor: ";
61                     cin >> valor;
62                     if(valor<0 || valor>9){
63                         cout<< "\n\tEl valor no esta en el rango, favor de volverlo a ingresar";
64                     }
65                 }while(valor<0 || valor>9);
66                 ///OBTENEMOS EL BLOQUE
67                 idBloque = bloques[fila-1][columna-1];
68
69                 ///COMPROBAMOS QUE SEA UN VALOR QUE NO SE REPITA
70                 check1 = buscarElemento(area,bloques,fila,columna,valor,idBloque);
71
72                 if(check1){
73                     cout << "\n\tNo se agrego elemento. Elemento repetido";
74                 }else{
75                     area[fila-1][columna-1]=valor;
76                     cout << "\n\tSe agrego elemento.";
77                 }
78             }
79         }
80     }
81 }
```

```

73     }
74
75     break;
76
77     case 2:
78         ///ELIMINAMOS VALOR
79         cout << "\n\tQue elemento vas a eliminar?";
80         cout << "\n\tFila: ";
81         cin >> fila;
82         cout << "\n\tColumna: ";
83         cin >> columna;
84
85         area[fila-1][columna-1]=0;
86         cout<< "\n\tSe ha eliminado el elemento";
87         break;
88
89     default:
90         cout<< "!!!\n\tOpcion invalida!!!";
91         break;
92
93     }
94
95     getch();
96
97     ///COMPROBAMOS QUE EL AREA NO ESTE LLENA
98     ///COMPROBAMOS QUE EL AREA NO ESTE LLENA
99     check = comprobar(area);
100
101 }while(check);
102
103 cout << "\n\tFELICIDADES!! HAS COMPLETADO EL SUDOKU!!";
104
105
106 getch();
107 }
108
109 void asignarBloques(char bloques[9][9]){
110
111     for(int i=0; i < 9;i++){
112
113         for(int j=0; j < 9; j++){
114
115             if(i>=0 && i<=2 ){
116
117                 if(j==0 || j==1 || j==2){
118                     bloques[i][j]='a';
119                 }else if(j==3 || j==4 || j==5){
120                     bloques[i][j]='b';
121                 }else{
122                     bloques[i][j]='c';
123                 }else{
124                     bloques[i][j]='c';
125                 }
126
127             if(i>=3 && i<=5 ){
128
129                 if(j==0 || j==1 || j==2){
130                     bloques[i][j]='d';
131                 }else if(j==3 || j==4 || j==5){
132                     bloques[i][j]='e';
133                 }else{
134                     bloques[i][j]='f';
135                 }
136
137             if(i>=6 && i<=8 ){
138
139                 if(j==0 || j==1 || j==2){
140                     bloques[i][j]='g';
141                 }else if(j==3 || j==4 || j==5){
142                     bloques[i][j]='h';
143                 }else{
144                     bloques[i][j]='i';
145                 }
146

```

```

145     }
146 }
147 }
148
149 }
150 }
151 }
152 }
153
154 void dibujarArea(int area[9][9]){
155
156     cout<< "\t\t\t\t\t SUDOKU";
157     cout<< "\n";
158     for(int i=0; i<10; i++){
159
160         for(int j=0; j < 10; j++){
161
162             if(j==4 || j==7 || j==1){
163                 cout<< "\t";
164             }
165
166             if(i==0 && j>0)
167                 cout << " " << j;
168             else if(j==0 && i>0)
169                 cout << "\t " << i;
170             else if(i!=0 && j!=0)
171                 cout << "\t " << i;
172             else if(i!=0 && j!=0)
173                 cout << " " << area[i-1][j-1];
174             else
175                 cout << "\t";
176
177         }
178         cout << "\n";
179         if(i==3 || i==6 || i==9){
180             cout<< "\n";
181         }
182     }
183 }
184
185 bool comprobar(int area[9][9]){
186     ///DEVUELVE TRUE SI ENCUENTRA CERO
187     bool check= false;
188
189     for(int i=0; i<9; i++){
190
191         for(int j=0; j < 9; j++){
192
193             if(area[i][j]==0){
194                 check= true;
195             }
196         }
197     }
198
199     return check;
200 }
201
202 bool buscarElemento(int area[9][9], char bloques[9][9], int fila, int columna, int valor, char idBloque){
203
204     bool check=false;
205
206     ///BUSCAMOS EN FILAS
207     for(int i=0; i < 9; i++){
208
209         ///DEVUELVE TRUE SI SE REPITE EN FILAS
210         if(area[fila-1][i]==valor)
211         {
212             check = true;
213             cout << "\n\tADVERTENCIA: el valor que desea ingresar se repite de la fila " << fila << ", colum
214             return check;
215         }
216     }
217
218 }

```



```

220 //BUSCAMOS EN COLUMNAS
221 for(int i=0; i < 9;i++){
222
223 //DEVUELVE TRUE SI SE REPITE EN COLUMNAS
224 if(area[i][columna-1]==valor){
225     check = true;
226     cout << "\n\tADVERTENCIA: el valor que desea ingresar se repite de la fila " << i+1 << ", columna " << columna;
227     return check;
228 }
229
230 }
231
232 //BUSCAMOS EN EL BLOQUE CORRESPONDIENTE
233 for(int i=0; i < 9;i++){
234
235     for(int j=0; j < 9;j++){
236         //DEVUELVE TRUE SI SE REPITE EN EL MISMO BLOQUE
237         if(bloques[i][j]==idBloque && area[i][j]==valor ){
238             check= true;
239             cout << "\n\tADVERTENCIA: el valor que desea ingresar ya es parte del mismo bloque. Fila " << i+1 << ", columna " << j+1;
240             return check;
241         }
242     }
243 }
244
245 }
246
247 return check;
248 }
249 }
250
251

```

## SUDOKU

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

Que desea hacer?

- 1) Ingresar valor
- 2) Eliminar valor

	SUDOKU								
	1	2	3	4	5	6	7	8	9
1	0	0	6	0	0	9	0	0	0
2	8	3	5	4	0	1	6	0	2
3	0	2	7	0	3	6	1	5	4
4	6	5	0	7	4	0	8	0	9
5	2	0	4	0	0	0	7	0	0
6	0	0	0	0	8	3	0	4	5
7	0	6	2	0	9	0	0	7	0
8	5	7	0	0	6	0	0	3	0
9	4	1	8	3	5	0	9	2	6

Que desea hacer?  
 1) Ingresar valor  
 2) Eliminar valor

Ingrese los datos para ingresar el elemento  
 Fila: 1

Columna: 1

Ingrese el valor: 1

Se agrego elemento.

	SUDOKU								
	1	2	3	4	5	6	7	8	9
1	1	0	6	0	0	9	0	0	0
2	8	3	5	4	0	1	6	0	2
3	0	2	7	0	3	6	1	5	4
4	6	5	0	7	4	0	8	0	9
5	2	0	4	0	0	0	7	0	0
6	0	0	0	0	8	3	0	4	5
7	0	6	2	0	9	0	0	7	0
8	5	7	0	0	6	0	0	3	0
9	4	1	8	3	5	0	9	2	6

Que desea hacer?  
 1) Ingresar valor  
 2) Eliminar valor

	1	2	3	SUDOKU			4	5	6	7	8	9
1	1	0	6	0	0	9	0	0	0	0	0	0
2	8	3	5	4	0	1	6	0	2	0	0	0
3	0	2	7	0	3	6	1	5	4	0	0	0
4	6	5	0	7	4	0	8	0	9	0	0	0
5	2	0	4	0	0	0	7	0	0	0	0	0
6	0	0	0	0	8	3	0	4	5	0	0	0
7	0	6	2	0	9	0	0	7	0	0	0	0
8	5	7	0	0	6	0	0	3	0	0	0	0
9	4	1	8	3	5	0	9	2	6	0	0	0

Que desea hacer?

1) Ingresar valor

2) Eliminar valor

1

Ingrese los datos para ingresar el elemento

Fila: 1

Columna: 2

Ingrese el valor: 7

ADVERTENCIA: el valor que desea ingresar se repite de la fila 8, columna 2  
No se agrego elemento. Elemento repetido

	1	2	3	SUDOKU			4	5	6	7	8	9
1	1	0	6	0	0	9	0	0	0	0	0	0
2	8	3	5	4	0	1	6	0	2	0	0	0
3	0	2	7	0	3	6	1	5	4	0	0	0
4	6	5	0	7	4	0	8	0	9	0	0	0
5	2	0	4	0	0	0	7	0	0	0	0	0
6	0	0	0	0	8	3	0	4	5	0	0	0
7	0	6	2	0	9	0	0	7	0	0	0	0
8	5	7	0	0	6	0	0	3	0	0	0	0
9	4	1	8	3	5	0	9	2	6	0	0	0

Que desea hacer?

1) Ingresar valor

2) Eliminar valor

1

Ingrese los datos para ingresar el elemento

Fila: 3

Columna: 1

Ingrese el valor: 4

ADVERTENCIA: el valor que desea ingresar se repite de la fila 3, columna 9  
No se agrego elemento. Elemento repetido\_

```
SUDOKU
      1  2  3      4  5  6      7  8  9
1      1  0  6      0  0  9      0  0  0
2      8  3  5      4  0  1      6  0  2
3      0  2  7      0  3  6      1  5  4

4      6  5  0      7  4  0      8  0  9
5      2  0  4      0  0  0      7  0  0
6      0  0  0      0  8  3      0  4  5

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

Que desea hacer?
1) Ingresar valor
2) Eliminar valor
1

Ingrese los datos para ingresar el elemento
Fila: 1

Columna: 2

Ingrese el valor: 8

ADVERTENCIA: el valor que desea ingresar ya es parte del mismo bloque. Fila 2, columna 1
No se agrego elemento. Elemento repetido
```

## **Conclusión:**

Al realizar todos los ejercicios me percate de la gran importancia que tiene los arreglos, no solo en la programación, desde tiempos antiguos se usaban para la realización de cifrados de textos (criptografía), hasta en la actualidad con los juegos (sudoku, ajedrez, etc.).

Aplicaciones de los arreglos:

- El cine para asignar los lugares a las personas e ir descartando lugares para tener un mejor control
- Para las matrices
- Para vectores
- En la escuela para llevar un control de calificaciones y al alumno que le pertenece

## **Bibliografía:**

- Corona Nakamura María Adriana, Ancona Valdez María de los Ángeles. (2011). Diseño de Algoritmos y su Codificación en Lenguaje C. México: Mc Graw Hill.
- Dr. Cairó Battistutti Osvaldo.. (2005). Metodología de la programación. México: Alfaomega.