	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: Estructuras de Datos y Algoritmos I (1227)

Grupo: 15

No de Práctica(s): Práctica 1 – Aplicaciones de arreglos

Integrante(s): Cadena Luna Iván Adrián

No. de Equipo de cómputo empleado: —

No. de Lista o Brigada: —

Semestre: 2021-2

Fecha de entrega: 16 de Marzo de 2021

Observaciones:

CALIFICACIÓN: _____

➢ OBJETIVO

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

➢ INTRODUCCIÓN

Una estructura de uso típico en todos los lenguajes de programación es el arreglo o array, el cual es un tipo estructurado de dato capaz de almacenar una colección de datos del mismo tipo. A diferencia de las variables comunes, los arreglos nos permiten guardar varias variables y acceder a ellas de manera independiente, es como tener una variable con distintos compartimentos donde podemos introducir datos distintos. Para ello utilizamos un índice que nos permite especificar el compartimiento o posición a la que nos estamos refiriendo.

El arreglo es una estructura multidimensional. En una dimensión puede ser visto como un vector, necesita de un índice (la posición) para recorrer sus elementos. Un arreglo de dos dimensiones puede representarse como una matriz, necesita de dos índices (fila y columna) para acceder a uno de sus elementos. Los arreglos son entidades estáticas debido a que se declaran de un cierto tamaño y conservan este todo a lo largo de la ejecución del programa en el cual fue declarado.

➢ DESARROLLO

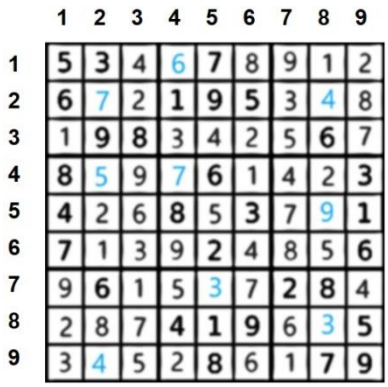
- Código (la escítala espartana).

```c
#include<stdio.h>

/*Programa que realiza la implementación de la escítala espartana
Para cifrar y descifrar.*/

void crearMensaje();
void descifrarMensaje();
int main(){
    short opcion=0;
    while (1){
        printf("\n\t*** ESCÍTALA ESPARTANA ***\n");
        printf("¿Qué desea realizar?\n");
        printf("1) Crear mensaje cifrado.\n");
        printf("2) Descifrar mensaje.\n");
        printf("3) Salir.\n");
        scanf("%d", &opcion);
         switch(opcion){
            case 1:
                crearMensaje(); //función creada para cifrar el mensaje
                break;
            case 2:
                descifrarMensaje(); ///función creada para descifrar el mensaje
                break;
            case 3:
                return 0;
            default:
                printf("Opción no válida.\n");}
    }
    return 0;
}
```

```c
void crearMensaje(){
    int ren, col, i, j, k=0;
    printf("Ingresar el tamaño de la escítala:\n");
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);
    char escitala[ren][col]; //arreglo de 'x' filas y 'y' columnas
    char texto[ren*col];
    printf("Escriba el texto a cifrar:\n");
    scanf("%s", texto); //el usuario indica el mensaje a cifrar
    for (i=0 ; i<ren ; i++)
        for (j=0 ; j<col ; j++)
            escitala[i][j]=texto[k++]; //el mensaje se guarda dentro del arreglo escitala
    printf("El texto en la tira queda de la siguiente manera:\n");
    for (i=0 ; i<col ; i++)
        for (j=0 ; j<ren ; j++)
            printf("%c", escitala[j][i]); //se transpone el arreglo para que el mensaje quede cifrado
    printf("\n");
}
void descifrarMensaje(){
    int ren, col, i, j, k=0;
    printf("Ingresar el tamaño de la escitala:\n"); /*para descifrar el mensaje se debe
                                                      indicar el número de filas y columnas que tiene el mensaje*/
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);
    char escitala[ren][col]; //arreglo de 'x' filas y 'y' columnas
    char texto[ren*col];
    printf("Escriba el texto a descifrar:\n"); //posteriormente ingresar el mensaje cifrado
    scanf("%s", texto);
    for (i=0 ; i<col ; i++)
        for (j=0 ; j<ren ; j++)
            escitala[j][i] = texto[k++];
    printf("El texto descifrado es:\n"); //una vez ingresada la información, el programa imprime el mensaje original
    for (i=0 ; i<ren ; i++)
        for (j=0 ; j<col ; j++)
            printf("%c", escitala[i][j]);
}
```

```
        *** ESC=TALA ESPARTANA ***
┐QuÚ desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
1
Ingresar el tamaıo de la escÝtala:

Renglones:2

Columnas:2
Escriba el texto a cifrar:
hola
El texto en la tira queda de la siguiente manera:
hloa

        *** ESC=TALA ESPARTANA ***
┐QuÚ desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
2
Ingresar el tamaıo de la escÝtala:

Renglones:2

Columnas:2
Escriba el texto a descifrar:
hloa
El texto descifrado es:
hola
```

- Buscar un sudoku en una revista, periódico, app o internet y desplegarlo en la pantalla con la ayuda de un arreglo bidimensional.
  Indicar al usuario qué casilla llenar con coordenadas y actualizar la matriz desplegada (puede ser consecutiva o limpiar pantalla y volver a escribir).



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

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

       6       0       2       1       9       5       3       0       8

       1       9       8       3       4       2       5       6       7

       8       0       9       0       6       1       4       2       3

       4       2       6       8       5       3       7       0       1

       7       1       3       9       2       4       8       5       6

       9       6       1       0       3       7       2       8       4

       2       8       7       4       1       9       6       0       5

       3       0       5       2       8       6       1       7       9

Ingrese el numero que va en la coordenada (2,2): 7
Ingrese el numero que va en la coordenada (1,4): 6
Ingrese el numero que va en la coordenada (2,8): 4
Ingrese el numero que va en la coordenada (3,2): 5
Ingrese el numero que va en la coordenada (4,4): 7
Ingrese el numero que va en la coordenada (5,8): 9
Ingrese el numero que va en la coordenada (9,2): 4
Ingrese el numero que va en la coordenada (7,4): 5
Ingrese el numero que va en la coordenada (8,8): 3


Sudoku completado:


       5       3       4       6       7       8       9       1       2

       6       7       2       1       9       5       3       4       8

       1       9       8       3       4       2       5       6       7

       8       5       9       7       6       1       4       2       3

       4       2       6       8       5       3       7       9       1

       7       1       3       9       2       4       8       5       6

       9       6       1       5       3       7       2       8       4

       2       8       7       4       1       9       6       3       5

       3       4       5       2       8       6       1       7       9

Presione una tecla para continuar . . .
```

```cpp
#include <iostream>
#include <stdio.h>
#include <stdlib.h>

using namespace std;

main(){
    int dato;
    int table[9][9]= {{5,3,4,0,7,8,9,1,2},
                      {6,0,2,1,9,5,3,0,8},
                      {1,9,8,3,4,2,5,6,7},
                      {8,0,9,0,6,1,4,2,3},
                      {4,2,6,8,5,3,7,0,1},
                      {7,1,3,9,2,4,8,5,6},
                      {9,6,1,0,3,7,2,8,4},
                      {2,8,7,4,1,9,6,0,5},
                      {3,0,5,2,8,6,1,7,9},
                      };

    for(int i=0;i<9;i++){
        for(int j=0;j<9;j++){
            cout<<"\t"<<table[i][j];
        }
        printf("\n\n");
    }

    printf("Ingrese el numero que va en la coordenada (2,2): ");
    scanf("%d", &dato);
    table[1][1]=dato;
    printf("Ingrese el numero que va en la coordenada (1,4): ");
    scanf("%d", &dato);
    table[0][3]=dato;
    printf("Ingrese el numero que va en la coordenada (2,8): ");
    scanf("%d", &dato);
    table[1][7]=dato;
    printf("Ingrese el numero que va en la coordenada (3,2): ");
    scanf("%d", &dato);
    table[3][1]=dato;
    printf("Ingrese el numero que va en la coordenada (4,4): ");
    scanf("%d", &dato);
    table[3][3]=dato;
    printf("Ingrese el numero que va en la coordenada (5,8): ");
    scanf("%d", &dato);
    table[4][7]=dato;
    printf("Ingrese el numero que va en la coordenada (9,2): ");
    scanf("%d", &dato);
    table[8][1]=dato;
    printf("Ingrese el numero que va en la coordenada (7,4): ");
    scanf("%d", &dato);
    table[6][3]=dato;
    printf("Ingrese el numero que va en la coordenada (8,8): ");
    scanf("%d", &dato);
    table[7][7]=dato;

    if(table[1][1]==7 && table[0][3]==6 && table[1][7]==4 && table[3][1]==5
    && table[3][3]==7 && table[4][7]==9 && table[8][1]==4 && table[6][3]==5 && table[7][7]==3){
        printf("\n\nSudoku completado:\n\n\n");
        for(int i=0;i<9;i++){
            for(int j=0;j<9;j++){
                cout<<"\t"<<table[i][j];
            }
            printf("\n\n");
        }
    }
    else{
        printf("\n\nIntentelo de nuevo\n\n");
    }

system("PAUSE");
return 0;
}
```

➢ CONCLUSIÓN

Entre las que ya hemos visto en el desarrollo de esta práctica. Los arreglos también tienen otro tipo de aplicaciones:

- **El manejo de listas.** Por ejemplo, una con los nombres de un grupo de alumnos y otra con una calificación para cada uno de los alumnos. Primero se deberán leer los nombres de los alumnos y la calificación que corresponde a cada uno de ellos y después habrá que desplegar dos columnas: una con los nombres de los alumnos y la otra con sus respectivas calificaciones.
  La lectura luciría de manera similar a la siguiente:

```
Número de alumnos: 30

Nombre [0] :

Calificación[0] :

Nombre [1] :

Calificación[1] :

Nombre [2] :

Calificación[2] :

...

...

Nombre [29] :

Calificación[29] :
```

- **Vectores.** En Física, los vectores sirven para representar velocidades, aceleraciones, etc. Podemos representar a un vector como un segmento de recta dirigida que tiene magnitud, orientación y sentido. En un espacio tridimensional, un vector puede expresarse por medio de tres componentes sobre los ejes cartesianos. Cada componente puede, a su vez, expresarse en función de los vectores unitarios i, j, k ,que se ubican en los ejes x, y, z, respectivamente.
  Un segmento de programa sería por ejemplo:

```
v1 = 20i + 15j + 35k

v2 = -5i + 40j + 25k

s = v1+v2 = (20-5)i +(15+40)j+(35+25)k

s = 15i + 55j + 60k          //suma de vectores
```

```
double v1[ ] = new double[3];
double v2[ ] = new double[3];
double s[ ] = new double[3];
v1[0] = 20;
v1[1] = 15;
v1[2] = 35;
v2[0] = -5;
v2[1] = 40;
v2[2] = 25;

for(int i=0; i < 3 ; i++)
{
s[ i ] = v1[ i ] + v2[ i ] ;
}
```

- **Posición de caracteres.** Por ejemplo, si se necesita manejar en un arreglo la posición todos los caracteres escritos en una enciclopedia. El arreglo deberá tener las siguientes dimensiones:
  -Una para manejar el renglón en la página,
  -otra para la columna en la página,
  -otra para la hoja en el volumen y
  -otra para el volumen en la enciclopedia.
  Por ejemplo, si:
  -Cada página tiene 30 renglones y 80 columnas.
  -Cada volumen tiene 500 páginas.
  -La enciclopedia tiene 18 volúmenes.
  El siguiente fragmento de programa implementaría la solución a este problema:

```
char car;
// Declara y crea el arreglo.
char[ , , , ] enciclopedia = new char [18,500,30,80] ;
// Lee, desde el teclado, cada carácter para la enciclopedia.
for(int v = 0 ; v < 18 ; v++) // Volumen
for(int p = 0 ; p < 500 ; p++) // Página
for(int r = 0 ; r < 30 ; r++) // Renglón
for(int c = 0 ; c < 80 ; c++) // Columna
{
     Car = (char)Console.Read( );
     enciclopedia[v,p,r,c] = car ;


}
```

En conclusión, en verdad entendemos que los arreglos son herramientas muy útiles para la programación. Calcular el promedio de calificaciones, hacer una lista de nombres de x personas, almacenar el inventario de una tienda, son algunos de los muchísimos ejemplos de problemas que podemos resolver utilizando arreglos. La información obtenida además del trabajo realizado en la práctica me hizo reflexionar y darme una mejor idea de la crucial importancia de los arreglos para resolver problemas de toda índole, no sólo matemáticos, también en el ámbito de los videojuegos o de codificación como vimos en los retos presentados, que se valieron del posicionamiento de los datos. Al utilizar arreglos, podemos evitar declarar el valor de tantas variables del mismo tipo y de esta manera lograr que nuestro programa sea más rápido, eficiente y versátil.

➢ BIBLIOGRAFÍA CONSULTADA

Elena de Lobos, M. (2005). *Estructuras de datos y arreglos*. Consultado el 15 de marzo de 2021, de http://www.mailxmail.com/curso-aprende-programar/estructuras-datos-arreglos

Desconocido. (2018). *Multidimensionales: conceptos básicos, operaciones y aplicaciones*. Consultado el 15 de marzo de 2021, de http://tutoriales-isc.blogspot.com/2018/05/52-multidimensionales-conceptos-basicos.html

Desconocido. (2018). *Unidimensionales: conceptos básicos, operaciones y aplicaciones*. Consultado el 15 de marzo de 2021, de http://tutoriales-isc.blogspot.com/2018/05/51-unidimensionales-conceptos-basicos.html