



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martinez Quintana

Asignatura: Estructura de datos y algoritmos I

Grupo: 15

No de Práctica(s): 1

Integrante(s): López Cruz Marino

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: No aplica

Semestre: 2021-2

Fecha de entrega: 15-Marzo-2021

Observaciones:

CALIFICACIÓN: _____

Practica 1. Aplicaciones de Arreglos.

Objetivo: Utilizar apuntadores en lenguaje C para acceder a las localidades de memoria tanto de datos primitivos como de arreglos.

Actividades:

- Crear apuntadores.
- Leer y modificar datos a través de apuntadores.
- 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)
- Llenar el sudoku y mostrar algunas capturas de pantalla en su reporte.
- En las conclusiones, además de las personales, indicar otros 3 tipos de aplicaciones de los arreglos (en la vida cotidiana, en tu campo de conocimiento y otro).

Introducción

Se le conoce como apuntador a una variable que tiene como contenido la dirección de memoria de otra variable. Gracias a que los apuntadores trabajan directamente con la memoria, es fácil acceder a la información a través de ellos.

La forma de declarar un apuntador es la que se muestra a continuación:

```
Tipodedato*apuntador,variable;
```

Donde se debe de especificar el tipo de dato, el nombre que se le dará al apuntador y el mismo tipo de dato de la variable que va a apuntar.

Para asignarle un valor, la forma correcta es la siguiente:

```
Apuntador=&variable;
```

Los apuntadores solo pueden apuntar variables con el mismo tipo de datos con el cual fueron declarados.

Se puede acceder (leer/modificar) mediante el puntero al valor que está almacenado en la dirección de memoria (la que está almacenada en el puntero por supuesto) utilizando el operador *

Para acceder a los miembros de clase de un objeto a través de un puntero se utiliza el operador flecha -> en lugar del operador punto, obviamente el acceso con operador flecha sigue respetando los niveles de acceso establecidos en la definición de la clase.

Traduciendo a lenguaje C el algoritmo de la escala Escítala Espartana.

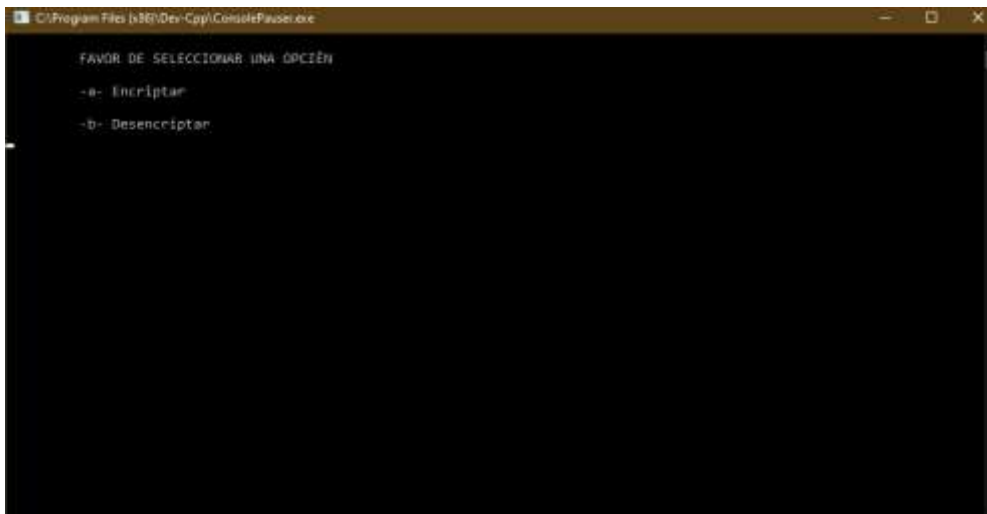
```
#include<stdio.h>
#include<math.h>
#include <stdlib.h>
#define p printf
#define s scanf
int main(){
int i,j;
char A[5][5],menu,R;
do
{
printf("\n\t FAVOR DE SELECCIONAR UNA OPCIÓN\n");
printf("\n\t -a- Encriptar\n");
printf("\n\t -b- Desencriptar\n");
scanf("%s",&menu);
switch(menu)
{
case 'a':
p("\nTu mensaje puede contener 25 letras");
for(i=1;i<=5;i++)
for(j=1;j<=5;j++)
{
fflush(stdin);
p("\n Dame una letra de tu mensaje: ");
s("%c",&A[i][j]);}
p("\nTu mensaje es: ");
for(i=1;i<=5;i++)
for(j=1;j<=5;j++)
p("\n%c",A[i][j]);
p("\nY cifrado queda: ");
for(i=1;i<=5;i++)
for(j=1;j<=5;j++)
p("\n%c",A[j][i]);
break;
case 'b':
```

```

        p("\nTu mensaje puede contener 25 letras");
for(i=1;i<=5;i++)
    for(j=1;j<=5;j++)
        {
            fflush(stdin);
            p("\n Dame una letra de tu mensaje: ");
            s("%c",&A[i][j]);}
p("\nTu mensaje es: ");
for(i=1;i<=5;i++)
    for(j=1;j<=5;j++)
        p("\n%c",A[i][j]);
p("\nY descifrado queda: ");
for(i=1;i<=5;i++)
    for(j=1;j<=5;j++)
        p("\n%c",A[j][i]);

break;
default:printf("\n\t OPCION NO VALIDA\n");
break;
} //cierre de switch
printf("\n\t DESEA VER EL MENU NUEVAMENTE Escriba s o S\n");
scanf("%s",&R);
}while(R=='S' || R=='s');//cierre de do-while
return 0;
}

```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe

FAVOR DE SELECCIONAR UNA OPCION
-a- Encriptar
-b- Desencriptar

a
Tu mensaje puede contener 25 letras
Dame una letra de tu mensaje:
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe

Dame una letra de tu mensaje: t
Dame una letra de tu mensaje: a
Dame una letra de tu mensaje: r
Dame una letra de tu mensaje:
Dame una letra de tu mensaje: e
Dame una letra de tu mensaje: s
Dame una letra de tu mensaje:
Dame una letra de tu mensaje: d
Dame una letra de tu mensaje: i
Dame una letra de tu mensaje: v
Dame una letra de tu mensaje: a
Dame una letra de tu mensaje: r
Dame una letra de tu mensaje: t
Dame una letra de tu mensaje: i
Dame una letra de tu mensaje: d
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe

Tu mensaje es:
E
n
c
r
i
p
t
a
r

a
v
e
r
t
i
d
o

Y cifrado queda:
E
```

Sudoku

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
6      0      1      0      0      4      0      8      0
0      4      9      7      3      0      0      0      0
8      2      3      0      0      0      0      4      9
0      0      4      0      0      0      0      9      6
0      0      7      0      0      0      2      0      0
9      8      0      0      0      0      3      0      0
4      1      0      0      0      0      9      6      5
0      0      0      0      4      1      8      3      0
0      5      0      2      0      0      4      0      7

-----
Process exited with return value 0
Press any key to continue . . .
```

Activar Wind

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
0      0      4      0      1      7      0      8      6
0      0      0      0      4      0      2      0      0
5      8      1      2      0      0      0      0      9
0      0      5      0      3      0      0      0      0
4      0      7      0      0      0      6      2      0
0      1      0      7      0      0      0      9      0
0      0      0      0      2      1      8      0      0
0      0      6      0      0      0      9      0      0
0      4      0      9      0      6      0      3      5

Ahora debes resolver el juego
Dame el valor que quieres ingresar para la casilla [1][1]= 9
Dame el valor que quieres ingresar para la casilla [1][2]= 2
Dame el valor que quieres ingresar para la casilla [1][4]= 5
Dame el valor que quieres ingresar para la casilla [1][7]= 3
Dame el valor que quieres ingresar para la casilla [2][1]=
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe

Dame el valor que quieres ingresar para la casilla [8][5]= 8
Dame el valor que quieres ingresar para la casilla [8][6]= 4
Dame el valor que quieres ingresar para la casilla [8][8]= 2
Dame el valor que quieres ingresar para la casilla [8][9]= 2
Dame el valor que quieres ingresar para la casilla [9][1]= 8
Dame el valor que quieres ingresar para la casilla [9][3]= 7
Dame el valor que quieres ingresar para la casilla [9][5]= 1
Dame el valor que quieres ingresar para la casilla [9][7]= 1
9      2      4      5      1      7      3      8      6
7      6      3      8      4      9      2      5      1
5      8      1      2      6      3      4      7      9
8      9      5      6      3      2      7      1      4
4      3      7      9      5      8      6      2      6
2      1      8      7      4      5      3      9      3
5      9      4      6      2      1      8      7      1
7      3      6      5      8      4      9      2      2
8      4      7      9      1      6      1      3      5

-----
Process exited with return value 0
Press any key to continue . . .
```

Conclusiones:

- Los arreglos y apuntadores tienen un campo de aplicaciones enorme en el ámbito computacional, teniendo la capacidad de resolver y plantear en interfases digitales, problemas complejos, que permiten estudiar fenómenos o desarrollar nuevas tecnologías.
- Los juegos como el sudoku son un claro ejemplo de lo laborioso mentalmente que puede llegar a ser un problema, sin embargo, la computadora programada de forma correcta tiene la capacidad para resolverlo sin mayores complicaciones.
- Es labor del ingeniero tener la capacidad de análisis que permita abstraer el proceso de resolución para programar a una computadora con el fin de resolver el problema.
- Dentro de las 3 aplicaciones que mencionare podemos hacer un enmarcado de los siguientes puntos:
 1. En la vida cotidiana los arreglos son de utilidad para la organización de datos que puedan ser útiles tales como tablas de conteo financiero o de actividades programadas con relaciones entre si.
 2. En mi campo de conocimiento, son una base fundamental para lograr avances en la automatización de sistemas complejos como el algoritmo diseñado para el autopilot de vehículos eléctricos.
 3. En el campo musical, a partir de una serie de arreglos se pueden definir las frecuencias y longitudes de onda ideales para la construcción o adaptación de espacios, que posean una acústica que permita grabaciones de alta calidad auditiva.