	Sistema de gestión de calidad
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de programación

Grupo: 3

No. de Práctica(s): #11

Integrante(s): María Guadalupe Martínez Pavón

No. de Equipo de cómputo empleado: No aplica

No. de Lista o

Semestre: 1

Fecha de entrega: 11-01-2021

Observaciones:

CALIFICACIÓN: _____

Práctica 11: Arreglos unidimensionales y multidimensionales

Objetivo: Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

Actividades:

♣ Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.

♣ Resolver un problema que requiera el uso de un arreglo de dos dimensiones, a través de un programa en lenguaje C.

♣ Manipular arreglos a través de índices y apuntadores.

Introducción:

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse. A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices. Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.

Arreglos unidimensionales

Código (arreglo unidimensional while)

```c
#include <stdio.h>
/*
      Este programa genera un arreglo unidimensional de 5 elementos y los
      accede a cada elemento del arreglo a través de un ciclo while.
*/
int main (){
    #define TAMANO 5
    int lista[TAMANO] = {10, 8, 5, 8, 7};

    int indice = 0;

    printf("\tLista\n");
    while (indice < 5 ){
    printf("\nCalificación del alumno %d es %d", indice+1, lista[indice]);
    indice += 1; // análogo a indice = indice + 1;
    }

    printf("\n");

    return 0;
}
```

```
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\lupit>cd OneDrive

C:\Users\lupit\OneDrive>cd Escritorio

C:\Users\lupit\OneDrive\Escritorio>cd "Lenguaje C"

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C>cd Ejemplos

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>gcc Awhile.c -o Awhile.exe

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>Awhile.exe
        Lista

Calificaci├n del alumno 1 es 10
Calificaci├n del alumno 2 es 8
Calificaci├n del alumno 3 es 5
Calificaci├n del alumno 4 es 8
Calificaci├n del alumno 5 es 7

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>
```

Código (arreglo unidimensional for)

```c
#include <stdio.h>
/*
      Este programa genera un arreglo unidimensional de 5 elementos y
      accede a cada elemento del arreglo a través de un ciclo for.
*/
int main (){
      #define TAMANO 5
      int lista[TAMANO] = {10, 8, 5, 8, 7};

      printf("\tLista\n");
      for (int indice = 0 ; indice < 5 ; indice++){
      printf("\nCalificación del alumno %d es %d", indice+1, lista[indice]);
     }

      printf("\n");

      return 0;
}
```

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\lupit>cd OneDrive

C:\Users\lupit\OneDrive>cd Escritorio

C:\Users\lupit\OneDrive\Escritorio>cd "Lenguaje C"

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C>cd Ejemplos

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>gcc Afor.c -o Afor.exe

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>Afor.exe
        Lista

Calificaci├│n del alumno 1 es 10
Calificaci├│n del alumno 2 es 8
Calificaci├│n del alumno 3 es 5
Calificaci├│n del alumno 4 es 8
Calificaci├│n del alumno 5 es 7

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>
```

Apuntadores

Código (apuntadores)

```c
#include <stdio.h>
/*
      Este programa crea un apuntador de tipo carácter.
*/
int main () {
      char *ap, c = 'a';
      ap = &c;

      printf("Carácter: %c\n",*ap);
      printf("Código ASCII: %d\n",*ap);
      printf("Dirección de memoria: %d\n",ap);

      return 0;
}
```

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\lupit>cd OneDrive

C:\Users\lupit\OneDrive>cd Escritorio

C:\Users\lupit\OneDrive\Escritorio>cd "Lenguaje C"

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C>cd Ejemplos

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>gcc Apuntadores.c -o Apuntadores.exe

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>Apuntadores.exe
Car├ícter: a
C├│digo ASCII: 97
Direcci├│n de memoria: 6422299

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>
```

Código (apuntadores)

```c
#include<stdio.h>
/*
    Este programa accede a las localidades de memoria de distintas variables a
    través de un apuntador.
*/
int main () {
    int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
    int *apEnt;
    apEnt = &a;

    printf("a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
    printf("apEnt = &a\n");

    b = *apEnt;
    printf("b = *apEnt \t-> b = %i\n", b);

    b = *apEnt +1;
    printf("b = *apEnt + 1 \t-> b = %i\n", b);

    *apEnt = 0;
    printf("*apEnt = 0 \t-> a = %i\n", a);

    apEnt = &c[0];
    printf("apEnt = &c[0] \t-> apEnt = %i\n", *apEnt);

    return 0;
}
```

```
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\lupit>cd OneDrive

C:\Users\lupit\OneDrive>cd Escritorio

C:\Users\lupit\OneDrive\Escritorio>cd "Lenguaje C"

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C>cd Ejemplos

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>gcc Aapuntadores.c -o Aapuntadores.exe

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>Aapuntadores.exe
a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}
apEnt = &a
b = *apEnt      -> b = 5
b = *apEnt + 1  -> b = 6
*apEnt = 0      -> a = 0
apEnt = &c[0]   -> apEnt = 5

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>
```

Código (apuntadores en ciclo for)

```c
#include <stdio.h>
/*
    Este programa genera un arreglo unidimensional de 5 elementos y
accede a cada elemento del arreglo a través de un apuntador
utilizando un ciclo for.
*/
int main (){
        #define TAMANO 5
        int lista[TAMANO] = {10, 8, 5, 8, 7};
        int *ap = lista;
        printf("\tLista\n");
        for (int indice = 0 ; indice < 5 ; indice++){
        printf("\nCalificación del alumno %d es %d", indice+1, *(ap+indice));
        }

        printf("\n");

  return 0;
}
```

```
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\lupit>cd OneDrive

C:\Users\lupit\OneDrive>cd Escritorio

C:\Users\lupit\OneDrive\Escritorio>cd "Lenguaje C"

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C>cd Ejemplos

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>gcc apuntadoresfor.c -o apuntadoresfor.exe

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>apuntadoresfor.exe
        Lista

Calificaci├n del alumno 1 es 10
Calificaci├n del alumno 2 es 8
Calificaci├n del alumno 3 es 5
Calificaci├n del alumno 4 es 8
Calificaci├n del alumno 5 es 7

C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>
```

Código (apuntadores en cadenas)

```c
#include <stdio.h>
/*
   Este programa muestra el manejo de cadenas en lenguaje C.
*/
int main(){
      char palabra[20];
      int i=0;
      printf("Ingrese una palabra: ");
      scanf("%s", palabra);
      printf("La palabra ingresada es: %s\n", palabra);
      for (i = 0 ; i < 20 ; i++){
      printf("%c\n", palabra[i]);
      }
  return 0;
}
```

```
C:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>apuntadorescadena.exe
Ingrese una palabra: Azul
La palabra ingresada es: Azul
A
z
u
l
```

Arreglos multidimensionales

Código (arreglos multidimensionales)

```c
#include<stdio.h>
/*   Este programa genera un arreglo de dos dimensiones (arreglo
multidimensional) y accede a sus elementos a través de dos ciclos
for, uno anidado dentro de otro.
*/
int main(){
      int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
      int i, j;
      printf("Imprimir Matriz\n");
      for (i=0 ; i<3 ; i++){
      for (j=0 ; j<3 ; j++){
      printf("%d, ",matriz[i][j]);
      }
      printf("\n");
     }
  return 0;
}
```

```
:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>gcc arreglosmultidimensionales.c -o arreglosmultidimensionales.ex

:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>arreglosmultidimensionales.exe
mprimir Matriz
, 2, 3,
, 5, 6,
, 8, 9,

:\Users\lupit\OneDrive\Escritorio\Lenguaje C\Ejemplos>
```

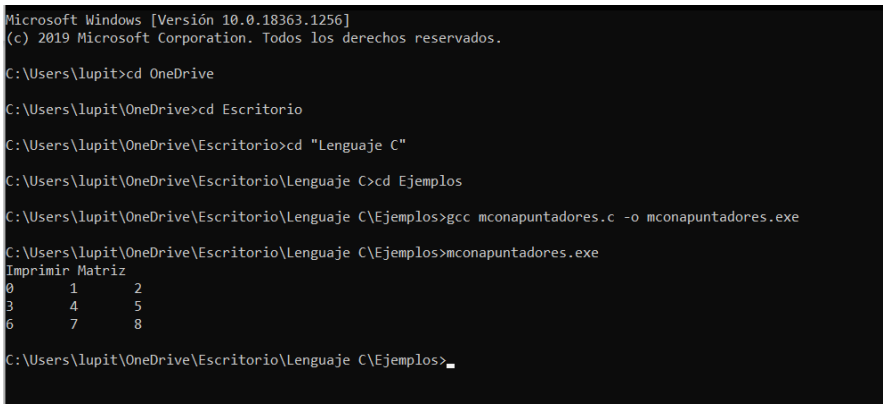
Código (arreglos multidimensionales con apuntadores)

```c
#include<stdio.h>
/*   Este programa genera un arreglo de dos dimensiones (arreglo
multidimensional) y accede a sus elementos a través de un apuntador utilizando
un ciclo for.
*/
int main(){
      int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
      int i, cont=0, *ap;
      ap = &i;

      printf("Imprimir Matriz\n");
      for (i=0 ; i<9 ; i++){
      if (cont == 3){

      printf("\n");
      cont = 0;
      }
      printf("%d\t",*(ap+i));
      cont++;
      }
      printf("\n");

      return 0;
}
```



Conclusión

Este tipo de arreglos nos van a permitir tener un orden mejor para identifica y acodar nuestras condiciones, considero que gracias a esto un trabajo se puede ver mas ordenado y una estética mejor, a parte que nos vas a dar más visualización de variantes u cualquier otro tipo, supongo que con este tipo de códigos se hacen para muchas cosas, pero mas en las bases de datos, para poder tener un orden y control, poder encontrar y saber un dato más rápido.

Como otra conclusión hubo un error en el ultimo ejemplo y con la depuración pude encontrar en donde sin embargo tuve que investigar cual era el error

Bibliografía

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.