

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de programación

Grupo: 3 sección B

No de Práctica(s): 11

Integrante(s): Frías Hernández Camille Emille Román

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

No. de Lista o Brigada: 15

Semestre: Primer Semestre

Fecha de entrega: 7/01/2021

Observaciones:

CALIFICACIÓN:

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.

Por otro lado, los apuntadores agilizan por nanosegundos la búsqueda de una variable, ya que al tener localizada la dirección donde está guardada la variable facilita su búsqueda.

Los arreglos unidimensionales son conjuntos de variables del mismo tipo. Su sintaxis es la siguiente:

tipoDeDato nombre [tamaño]

Un apuntador es una variable que contiene la dirección de una variable, es decir, hace referencia a la localidad de memoria de otra variable. Debido a que los apuntadores trabajan directamente con la memoria, a través de ellos se accede con rapidez a un dato. La sintaxis para declarar un apuntador y para asignarle la dirección de memoria de otra variable es, respectivamente:

TipoDeDato *apuntador, variable;

apuntador = &variable;

Los arreglos multidimensionales

Lenguaje C permite crear arreglos de varias dimensiones con la siguiente sintaxis:

tipoDato nombre [tamaño] [tamaño]...[tamaño];

Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimensión (el número de dimensiones está determinado por el número de corchetes).

Los tipos de dato que puede tolerar un arreglo multidimensional son: entero, real, carácter o estructura.

Actividad:

```

#include <stdio.h>
/*Este programa genera un arreglo unidimensional de 5 elementos y los accede a cada elemento del arreglo.*/
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;
}

```

```

C:\Users\Ansotec\Desktop\fundamentos de programación\Cosas de C y talvez algo más>"Arreglo unidimensional.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\Ansotec\Desktop\fundamentos de programación\Cosas de C y talvez algo más>_

```

```

#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.*/
intmain()
{
    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");
    }
}

```

```

C:\Users\Ansotec\Desktop\fundamentos de programación\Cosas de C y talvez algo más>ArregloMulti.exe
Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,

C:\Users\Ansotec\Desktop\fundamentos de programación\Cosas de C y talvez algo más>

```

```
#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;
}
```

```
C:\Users\Ansotec\Desktop\fundamentos de programación\Cosas de C y talvez algo más>Apuntadores.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\Ansotec\Desktop\fundamentos de programación\Cosas de C y talvez algo más>
```

Conclusión:

Los arreglos tanto multidimensionales como unidimensionales son de vital importancia para poder reducir las veces que ponemos variables o frases dentro del código reduciendo consecuentemente a este de manera significativa, por otro lado los apuntadores si bien hacen que el código sea más largo logran un mayor desempeño dentro del programa lo cual puede optimizar su rendimiento a la hora de correrlo.