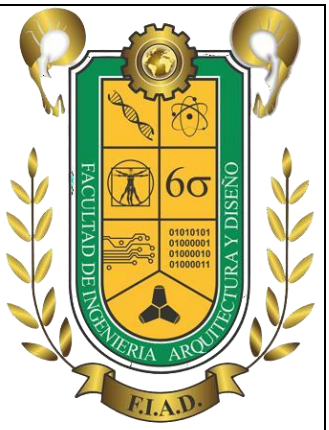

Universidad Autónoma deBaja California

Facultad de Ingeniería, Arquitectura yDiseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 8:

Arreglos

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 3 de Octubre del 2023

Introducción:

Los arreglos son estructuras de datos fundamentales en el lenguaje de programación C++, que permiten almacenar un conjunto de elementos del mismo tipo de manera contigua en la memoria. Estos elementos pueden ser variables de cualquier tipo de datos, como números enteros, números flotantes, caracteres, objetos personalizados, etc. Los arreglos proporcionan una forma eficiente de trabajar con conjuntos de datos homogéneos y acceder a sus elementos mediante índices**.**

Los arreglos se declaran especificando el tipo de datos de los elementos que contendrán, seguido de un nombre y, opcionalmente, el tamaño del arreglo entre corchetes. Los elementos dentro de un arreglo se almacenan de manera secuencial en la memoria, lo que significa que pueden accederse de manera rápida y eficiente mediante un índice numérico.

Competencia:

Dominio en arreglos en c y sus funciones, con la elaboración de la práctica se espera desarrollar problemas complejos y de mayor lógica que utilizan todas esas funciones sin la necesidad de buscarlo por internet ya que de esta manera se aprende más optimizado y practico

Fundamentos:

En el siguiente capítulo se presentan los arreglos y las cadenas. Las cadenas se consideran como un arreglo de tipo

char.

5.1 Arreglos unidimensionales y multidimensionales

Los arreglos son una colección de variables del mismo tipo que se referencian utilizando un nombre común. Un arreglo consta de posiciones de memoria contigua. La dirección más baja corresponde al primer elemento y la más alta al último. Un arreglo puede tener una o varias dimensiones. Para acceder a un elemento en particular de un arreglo se usa un índice.

El formato para declarar un arreglo unidimensional es:

tipo nombre_arr [tamaño]

Por ejemplo, para declarar un arreglo de enteros llamado lista num con diez elementos se hace de la siguiente forma:

int listanum[10];

En C, todos los arreglos usan cero como índice para el primer elemento. Por tanto, el ejemplo anterior declara un arreglo de enteros con diez elementos desde listanum[0] hasta listanum[9].

La forma como pueden ser accesados los elementos de un arreglo, es de la siguiente forma:

listanum[2] = 15; /* Asigna 15 al 3er elemento del arreglo listanum*/

num = listanum[2]; /* Asigna el contenido del 3er elemento a la variable num */

El lenguaje C no realiza comprobación de contornos en los arreglos. En el caso de que sobrepase el final durante una operación de asignación, entonces se asignan valores a otra variable o a un trozo del código, esto es, si se dimensiona un arreglo de tamaño N, se puede referenciar el arreglo por encima de N sin provocar ningún mensaje de error en tiempo de compilación o ejecución, incluso aunque probablemente se provoque el fallo del programa.

Como programador se es responsable de asegurar que todos los arreglos sean lo suficientemente grandes para guardar lo que pondrá en ellos el programa.

C permite arreglos con más de una dimensión , el formato general es:

tipo nombre_arr [tam1][tam2] ... [tamN];

Por ejemplo un arreglo de enteros bidimensionales se escribirá como:

```
int tabladenums[50][50];
```

Observar que para declarar cada dimensión lleva sus propios paréntesis cuadrados.

Para acceder los elementos se procede de forma similar al ejemplo del arreglo unidimensional, esto es,

```
tabladenums[2][3] = 15; /* Asigna 15 al elemento de la 3a fila y la 4a columna*/

num = tabladenums[25][16];
```

A continuación se muestra un ejemplo que asigna al primer elemento de un arreglo bidimensional cero, al siguiente 1,

y así sucesivamente.

```
main()

{

int t,i,num[3][4];

for(t=0; t<3; ++t)

for(i=0; i<4; ++i)

num[t][i]=(t*4)+i*1;

for(t=0; t<3; ++t)

{

for(i=0; i<4; ++i)

printf("num[%d][%d]=%d ", t,i,num[t][i]);

printf("\n");

}

}
```

En C se permite la inicialización de arreglos, debiendo seguir el siguiente formato:

tipo nombre_arr[tam1][tam2] ... [tamN] = {lista-valores};

Por ejemplo:

int i[10] = {1,2,3,4,5,6,7,8,9,10};

int num[3][4]={0,1,2,3,4,5,6,7,8,9,10,11};

Procedimiento:

ACTIVIDAD 8

Realiza programa en C el programa deberá tener el siguiente menú.

MENÚ
1.- LLENAR VECTOR 1 (MANUALMENTE)
2.- LLENAR VECTOR 2 ALEATORIAMENTE
3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)
4.- IMPRIMIR VECTORES
5.- LLENA MATRIZ 4 X 4
6.- IMPRIMIR MATRIZ
0.- SALIR

NOTA: EL PROGRAMA DEBERÁ REPETIRSE CUANTAS VECES LO DESEE EL USUARIO
NOTA 2: EL **VECTOR 1** DE **10** POSICIONES, NÚMEROS DEL **30 AL 70**
NOTA 3: EL **VECTOR 2** DE **10** POSICIONES CON NÚMEROS GENERADOS ALEATORIAMENTE DEL **1 AL 20 (SIN REPETIR)**
NOTA 4: EL **VECTOR 3** DE **20** POSICIONES, CON LOS DATOS DEL ARREGLO1 Y ARREGLO2
NOTA 5: MATRIZ 4 X 4 LLENARLA CON LOS DATOS DEL VECTOR1 Y VECTOR2,

Conclusiones:

Se logró aprender el uso correcto de arreglos, se aprendió el uso correcto al ingresar los datos como las matrices y los vectores

Se aprendió a usar las diversas operaciones para tener un programa mejor estructurado y eficaz

Anexo:

https://github.com/BrayanArM98/Estructurada/blob/main/ACTIVIDAD%208%20ESTRUCTURADA.pdf