Arreglo Unidimensional

es un tipo de datos estructurado que está formado de una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales. El tipo de acceso a los arreglos unidimensionales es el acceso directo, es decir, podemos acceder a cualquier elemento del arreglo sin tener que consultar a elementos anteriores o posteriores, esto mediante el uso de un índice para cada elemento del arreglo que nos da su posición relativa

Declaracion de arreglos unidimensionales

La declaración de arreglos sigue la estructura general de las declaraciones, es decir consta de un especificador de clase de almacenamiento opcional, un especificador de tipo de dato, el identificador del arreglo, el operador u operadores de arreglo con su respectiva dimensión y el inicializador es opcional.

Para declarar un arreglo unidimensional en C++ se utiliza la siguiente línea de código: tipo_dato identificador[tamaño];

inicialización

La inicialización de arreglos se realiza por medio del conjunto de valores iniciales de los distintos elementos del arreglo, agrupado por medio de llaves. La asignación de valores al arreglo unidimensional es en orden de secuencia con respecto al valor del índice [0] a [N].

Asignación

Iniciamos con el nombre de nuestro arreglo en el apartado que dice arreglo, entre los corchetes colocaremos que parte del array utilizaremos

En la parte “valor” asignaremos el valor que le daremos a ese elemento quedando de la siguiente manera: 1 arreglo[1]=100;

Hay que tomar en cuenta : Para recorrer un vector es necesario usar un ciclo. El ciclo clásicamente usado para recorrer un arreglo es un ciclo for.

Arreglos bidimensionales

Un arreglo de una dimensión es usado para representar elementos en una lista o secuencia de valores. En algunos casos la relación entre elementos del arreglo no pueden representarse en forma de lista.

Un arreglo de 2 dimensiones es usado para representar elementos en una tabla con columnas y filas, se puede acceder a cualquier elemento dentro del arreglo indicando el indice de la columna y la fila del arreglo.

inicialización

Inicialización Una matriz o arreglo bidimensional se puede inicializar de este modo: int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; Con la anterior asignación se crea en memoria una matriz

También podemos utilizar una estructura for dentro de otra estructura for para inicializar los valores de un arreglo de dos dimensiones como se muestra a continuación:

Leer desde teclado una matriz de números enteros de dimensión 3x3. #include <stdio.h> void main() { const int TAM=3; int matriz[TAM][TAM]; for(int i=0; i<TAM ; i++) { for(int j=0; j<TAM; j++) { cout<<”Ingrese el elemento [“<<i<<”,“<<j<<”] “; cin>>matriz[I][j]; } } }

Recorrido de arreglos bidimensionales

Recorrido de secuencias en arrays Todas las posiciones ocupadas: Tamaño del array = longitud de la secuencia N elementos en un array de N posiciones: Recorrer el array desde la primera posición hasta la última Posiciones libres al final del array: Tamaño del array > longitud de la secuencia Con centinela: Recorrer el array hasta encontrar el valor centinela Con contador de elementos: Recorrer el array hasta el índice contador

Declaracion

Declaración Un arreglo bidimensional se define así: int arreglo[10][10]; float matriz-[10][10]; también podemos utilizar constantes para definir la dimensión del arreglo de dos dimensiones: const int N = 10; int arreglo[N][N];

Funciones con arreglos

Unidimensional C ++ no permite pasar una matriz completa como un argumento a una función. Sin embargo, puedes pasar un puntero a una matriz especificando el nombre de la matriz sin un índice. Para pasar una matriz de una sola dimensión como argumento en una función, se debe declarar el parámetro formal en una de las siguientes tres formas y los tres métodos de declaración producirán resultados similares porque cada uno le dice al compilador que recibirá un puntero entero. Parámetro como puntero. void MiFuncion(int *parametro) { //código de la función } Parámetro como un Array de tamaño fijo void MiFuncion(int* parametro[10]) { //código de la función }

Funciones con arreglos (cont)

Bidimensional Cuando hay que pasar una matriz bidimensional como argumento a una función, la declaración de parámetros formales de esta debe incluir el número de columnas, ya que la función receptora debe conoce la estructura interna de la matriz, para poder para acceder a sus elementos, y esto solo es posible informándole de su tipo y dimensiones. En el caso que nos ocupa las dimensiones son dos, por lo que la definición de la función llamada sería algo así: func (int dias[2][12]) {...} Observe que en la expresión anterior está incluida toda la información necesaria: número de filas, número de columnas y tamaño de cada elemento (un int). Desde luego la función receptora necesita conocer también la dirección de inicio del almacenamiento, pero ya hemos señalado que "el identificador de una matriz puede ser utilizado como un puntero a su primer elemento", con lo que si mentalmente sustit-uimos dias por un puntero al número 31 (primer elemento) de la primera matriz, la información pasada es completa.

By **Gael Octavio Leon Jimenez** (PrroGamer)
cheatography.com/prrogamer/

Published 8th December, 2020.
Last updated 8th December, 2020.