

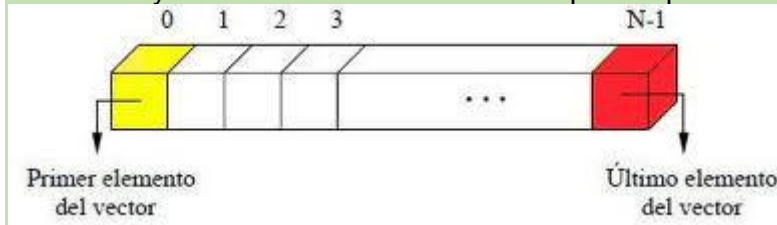
LEARN2GROW



LEARN2GROW
Build Your own Future

Definición de arreglos

Un arreglo es en resumen un conjunto de datos finito y del mismo tipo. En realidad funciona como cualquier variable cualquiera, excepto que en lugar de almacenar un solo valor, guarda algunos valores. Los arreglos pueden representarse en varias dimensiones: Unidimensionales, Bidimensionales, hasta n Dimensionales. En esta unidad se estudiarán los de una y dos dimensiones. Iniciaremos por los primeros.



Con esta imagen es posible observar que un arreglo unidimensional es una estructura, dividida en celdas, donde cada una de ellas representa un espacio de almacenamiento. Para acceder a sus posiciones se utilizan los pequeños números que en la figura aparecen la parte superior, se les denomina índices, y representan las posiciones. El primer índice en realidad depende del software, y generalmente es 0 o 1. En el caso del lenguaje de programación C y Java, este índice inicial es 0.

La imagen también permite demostrar que es FINITO; así es posible observar que tiene un inicio, en la posición 0 y un final en la posición N-1, donde N es el tamaño del arreglo. Es importante recordar que el tamaño del arreglo, representa la cantidad de memoria que se reservará para usar esta estructura, por lo que NO podrá variarse durante el tiempo de ejecución de un programa.

tipos de arreglos

Por el momento vamos ver los tipos de arreglos y hablaremos de un solo un tipo, ya que en el siguiente apartado estaremos hablando de los otros tipos de arreglos

1. Arreglos unidimensionales (Una dimensión)
2. Arreglos multidimensionales. (Dos dimensiones)
3. Arreglo con múltiple subíndices. (Múltiples dimensiones)

Arreglo de una sola dimensión

Arreglo Unidimensional

Finito: Contiene N elementos.
Homogéneo: del mismo tipo.
Ordenado: con una posición.
Referenciado: con un índice.

PRIMERO	SEGUNDO	TERCERO	CUARTO	QUINTO	SEXTO	SÉPTIMO	...	ENÉSIMO
C_1	C_2	C_3	C_4	C_5	C_6	C_7	...	C_N
i_0	i_1	i_2	i_3	i_4	i_5	i_6	...	C_{N-1}

Arreglos Unidimensionales

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. Están formados por un conjunto de elementos de un mismo tipo de datos que se almacenan bajo un mismo nombre, y se diferencian por la posición que tiene cada elemento dentro del arreglo de datos. Al declarar un arreglo, se debe inicializar sus elementos antes de utilizarlos. Para declarar un arreglo tiene que indicar su tipo, un nombre único y la cantidad de elementos que va a contener.

Un vector es una array unidimensional.

ejemplo: Para declarar un vector de 10 enteros:

Definir vector como Entero;

Dimension vector[10];

Para acceder a cada uno de los elementos del vector utilizamos un índice. el primer elemento se accede con el índice 0. Podemos trabajar individualmente con cada uno de los elementos:

vector[0]<-10;

Escribir vector[0];

El acceso a un elemento que no existe producirá un error, por ejemplo:

vector[10]<-10;

Este error ocurre ya que en la posición 10 del vector no existe ya que, es cierto pusimos entre corchetes el numero 10 porque almacenara 10 valores, pero hay que recordar que en este caso los arreglos se están trabajando desde cero y no desde 1.

Entonces queda claro que en este caso, si queremos recorrer a ultima posición de mi arreglo seria:

vector[9]<-10;

y en esa ultima posición estamos dándole el valor de 10

Recorrido de un arreglo

Vamos a inicializar todos los elementos de un vector. Para ello vamos a recorrer el vector e inicializar cada elemento con un valor ,por ejemplo lo vamos a inicializar a 0. Para recorrer un vector utilizamos un bucle Para:

Para i<-0 hasta 9 Hacer

array[i]<-0;

FinPara

En este ejemplo le decimos al ciclo para que empiece desde 0 y hasta 10 entonces después ponemos el nombre del arreglos y entre corchetes le ponemos la letra [i] ya que esta letra es quien recorrerá las posiciones del arreglo

Podríamos recorrer el vector para mostrar el valor de los elementos:

Para i<-0 hasta 9 Hacer

Escribir array[i];

FinPara