



Guia #1: Arrays unidimensionales

Objetivo: Conocer el concepto, declaración y uso de los arrays en el lenguaje C y comprender su utilidad.

Un arreglo es una colección ordenada de elementos del mismo tipo. Los arreglos permiten referirse a una colección de valores mediante un solo nombre. Para acceder a los elementos individuales de un arreglo se utiliza uno o más índices que deben ser un número entero o una expresión de tipo entero.

Los arreglos pueden ser de cualquier tipo: int, float, double, char, o se puede tener elementos que sea estructuras más complejas. Cada elemento del arreglo almacena un valor del tipo básico.

Los arreglos de una dimensión o vectores solo utilizan un índice para localizar cada uno de sus elementos componentes. Para referirse a un elemento del arreglo se utilizan corchetes ([]) para encerrar el índice del elemento del arreglo. En el lenguaje C los índices varían entre cero y un máximo igual al número de elementos del arreglo menos uno.

DECLARACION:

Tipo nombre[tamaño];

El tamaño es cualquier constante entera positiva mayor que cero. Ejemplos:

int a[23]; - declara un arreglo de 23 elementos de tipo entero.

float m[120]; - declara un arreglo de 120 elementos de tipo flotante.

char c[256]; - declara un arreglo de 256 elementos de tipo carácter.

Nota

En C puede dejar corchetes vacíos, sólo cuando se asignan valores al array tal como:

```
int numeros[ ] = { 1, 2, 3, 4, 5 }
```

El compilador asigna automáticamente cinco elementos a cuenta

```
int z[10] = {2,4,6,8,10,12,14,16,18,20};
```

```
double w[10] = {12.5,-45.3,8.7};
```

Los arreglos de caracteres pueden iniciarse con cadenas de caracteres, como se muestra.

```
char s[ ] = "hola";
```

o

```
char s[10] = { 'h', 'o', 'l', 'a' };
```

```
ejemplo1.c
1  #include<stdio.h>
2  int main()
3  {
4      //tipo nombre tamano ={valor}
5      char hola[]={ "Hola Mundo"}; //No le indicamos el valor del tamano
6      printf("%c", hola[0]);
7      return 0;
8  }
9  }
```

Tamaño de un array

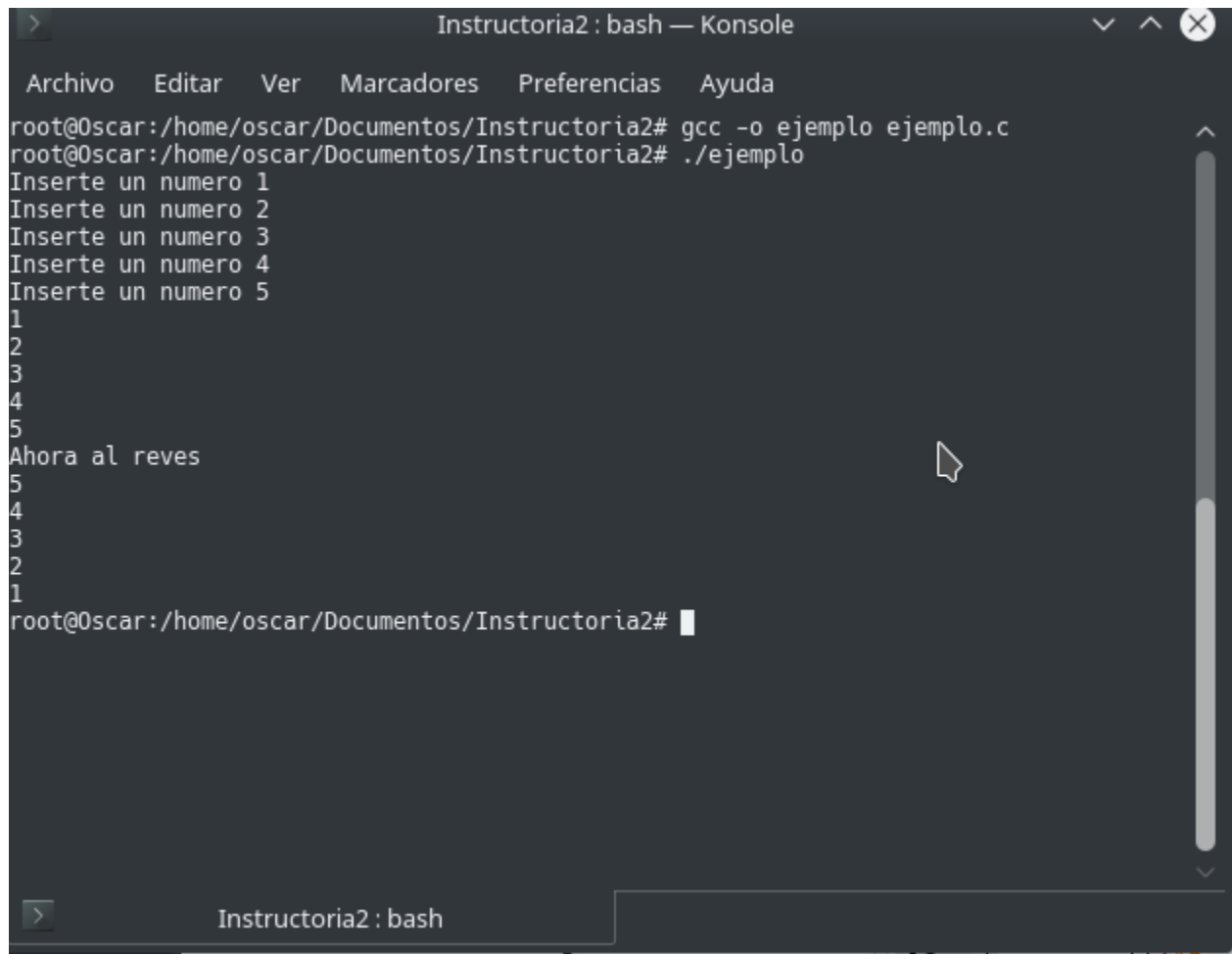
El operador sizeof() devuelve el numero de bytes necesarios para contener su argumento. Si se usa para solicitar el tamaño de un array, esta funcion devuelve el numero de bytes reservados para el array completo. Por ejemplo se tiene un array de enteros con 100 elementos llamado edades al utilizar la siguiente sentencia `n = sizeof(edades);`

El valor de n seria de 200.

Ejemplo1. Un programa que pida al usuario 5 números reales y luego los muestre en el orden contrario al que se introdujeron.

```
ejemplo.c
1  /*Un programa que pida al usuario 5 números reales y luego los muestre en el orden
2  contrario al que se introdujeron.*/
3  #include<stdio.h>
4  #define tamano 5
5  int main(){
6      int numeros[tamano];
7      int i;
8      //LLENADO DEL ARRAY HACIENDO USO DE UN FOR
9      for(i=0;i<=tamano-1;i++){
10         printf("Inserte un numero ");
11         scanf("%i", &numeros[i]);
12     }
13     //IMPRIMIR EL ARRAY EN EL ORDEN QUE FUE INGRESADO
14     for(i=0;i<=tamano-1;i++){
15         printf("%d \n", numeros[i]);
16     }
17     printf("Ahora al reves \n");
18     //IMPRIMIR EL ARRAY AL REVES DE COMO FUE INGRESADO
19     for(i=tamano-1;i>=0;i--){
20         printf("%d \n", numeros[i]);
21     }
22
23     return 0;
24 }
25
```

Ejemplo de como se veria la ejecucion.



```
Instructoria2 : bash — Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
root@oscar:/home/oscar/Documentos/Instructoria2# gcc -o ejemplo ejemplo.c
root@oscar:/home/oscar/Documentos/Instructoria2# ./ejemplo
Inserte un numero 1
Inserte un numero 2
Inserte un numero 3
Inserte un numero 4
Inserte un numero 5
1
2
3
4
5
Ahora al reves
5
4
3
2
1
root@oscar:/home/oscar/Documentos/Instructoria2#
```

Ejemplo 2. Crear un programa que sume todos los elementos de un array.

```
ejemplo2.c
1 //Crear un programa que sume todos los elementos de un array.
2 #include<stdio.h>
3 //METODO SUMA DEBE SER DECLARADO EN LA CABECERA PARA QUE TENGA AMBITO GLOBAL
4 int suma=0;
5
6 int main(){
7     int tamano;
8     printf("Escriba cuantos elementos guardara el array: ");
9     scanf("%d", &tamano);
10    int numeros[tamano];
11    //LENADO DEL VECTOR
12    printf("Ingrese los numeros: \n");
13    for(int i=0;i<tamano;i++){
14        scanf("%d", &numeros[i]);
15    }
16    //METODO PARA SUMAR LOS VALORES
17    for(int i=0;i<tamano;i++){
18        suma=suma+numeros[i];
19    }
20    printf("El resultado es: ");
21    printf("%d\n", suma);
22    return 0;
23 }
24
```

```
oscar : bash — Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
root@oscar:/home/oscar/Documentos/Instructoria2# gcc -o ejemplo2 ejemplo2.c
root@oscar:/home/oscar/Documentos/Instructoria2# ./ejemplo2
Escriba cuantos elementos guardara el array: 5
Ingrese los numeros:
1
2
3
4
5
El resultado es: 15
root@oscar:/home/oscar/Documentos/Instructoria2#
```

EJERCICIOS PROPUESTOS

1.Modificar el Ejemplo #1 para que el usuario pueda ingresar los datos que quiera y que igualmente se impriman en reverso.

2.Programa que halle la cantidad de números pares e impares en un vector de 20 posiciones