PROGRAMACIÓN ICOR312

MSC. FRANKLIN SÁNCHEZ

franklin.sanchez@epn.edu.ec

ARREGLOS

Arreglos

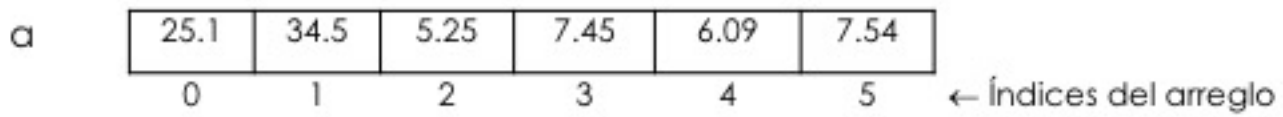
- Muchos programas requieren tipos de datos más complejos que los datos elementales.
- Estos tipos de datos son los tipos estructurados y están compuestos por varios elementos de igual o mismo tipo.
- Un arreglo es un conjunto ordenado de datos del mismo tipo.
- Los datos dentro del array se llaman elementos o términos
- Su posición dentro del arreglo se enumera consecutivamente iniciando desde cero 0, 1, 2, 3... etc. [1]
- El tipo de elementos almacenados en un array puede ser cualquiera de los tipos de dato válido de C

Arreglos

- Los arreglos son una manera conveniente de agrupar muchas variables bajo un mismo nombre.
- Los arreglos pueden ser unidimensionales como una lista o vector , bidimensionales como una tabla o matriz, tridimensionales como Cubo Rugby o un edificio de apartamentos; de hecho, pueden tener cualquier dimensión arbitraria. [2]

Arreglos

- Es siguiente ejemplo muestra un arreglo unidimesional de 6 elementos de nombre “a”:



- En el array del ejemplo el primer elemento del arreglo “a”, a[0] contiene 25.1, el segundo elemento a[1] contiene 34.5 y así sucesivamente.
- También se puede observar que los índices de un array siempre tienen como límite inferior 0 y como límite superior el tamaño (número de elementos) del array menos 1.

Declaración de un vector

- Al igual que con cualquier tipo de variable los arreglos deben ser declarados antes de ser utilizados, la declaración es igual a la de cualquier variable, excepto que se debe indicar el tamaño o longitud del array. [1]
- La sintaxis para declarar un vector es la siguiente:

```
/* Declaración de una variable de tipo vector */
tipo_elemento nom_variable[dimension];

/* Declaración e inicialización de una variable de
   tipo vector */
tipo_elementos nom_variable[dimension]={val_primer_elemento,
                                        val_segundo_elemento,
                                        ...};
```

Declaración de un vector

- Ejemplos de declaración de un vector

```
int cont[4];          /* Declara la variable cont como un
                         vector formado por 4 enteros */

#define DIM 80
char frase[DIM];      /* Declara la variable frase como un
                         vector de caracteres de dimensión DIM,
                         donde DIM está definido como una
                         constante con valor 80 */

int v[4]={0,0,0,0};   /* Declara e inicializa la variable
                         v como un vector formado por 4
                         enteros, inicializados todos a 0 */
```

Inicialización de un vector

- Los array deben ser inicializados antes de ser utilizados, tal como sucede con cualquier variable. Para asignar valores a cada elemento del array, se puede escribir:

```
numeros[0] = 10;
numeros[1] = 20;
numeros[2] = 10;
...
```

- Otro método utilizado es inicializar el array completo en una sola sentencia. En este caso el tamaño del array se puede determinar automáticamente por el número de elementos en la inicialización.

```
int numeros [6] = {10, 20, 30, 40, 50, 60};
int numeros [ ] = {3, 4, 5};  /*Declara un arreglo de tres elementos*/
```

Inicialización de un vector

- Para asignar valores a arrays de mayor tamaño, se utiliza la lazos for o while/do-while. Por ejemplo, para inicializar todos los valores de un array a cero se puede usar la siguiente sentencia:

```
for(i=0; i<6; i++){
    numeros[i] = 0;
}
```

como el valor del subíndice i varía de 0 a 5, cada elemento del array se establece en cero.

- C no valida que los valores del arreglo estén dentro del rango declarado

Ejemplo

El siguiente programa lee desde el terminal ocho números, a continuación visualiza cada unos de los elementos del arreglo y la suma total de ellos.

```
#include<stdio.h>
#define NUM 8

int main (){
    int nums[NUM];
    int i, total=0;
    /*Leo los numeros*/
    for(i=0; i<NUM; i++){
        printf("Por favor, introduzca el numero: ");
        scanf("%d",&nums[i]);
    }
    /*Imprimo los numeros*/
    printf("\nLista de numeros: ");
    for(i=0; i<NUM; i++){
        printf("%d ", nums[i]);
        total += nums[i];
    }

    printf("\nLa suma de los numeros es: %d", total);
    return 0;
}
```

Uso de arreglos como parámetros

- En C los nombres de los arrays contienen la una referencia a la dirección de memoria donde inicia el array
- Cuando se pasa un array como parámentro se está pasando la referencia al array, a esto se conoce como paso de argumentos por referencia
- Como ejemplo crearemos el protoitpo de una función que sume los elementos de un arreglo, llamada "sumaDeDatos".

```
int sumaDeDatos(int datos[],int n);
```

- En el prototipo se pasan dos argumentos: int datos[ ] pasa la referencia al vector, e int n pasa el número de elementos del arreglo.

Ejemplo

El siguiente programa lee desde desde consola 5 números usando la función `leer()` y los guarda en una arreglo, luego imprime los datos usando la función `imprimir()`.

```
#include<stdio.h>
#define NUM 5

void leer (int arreglo[],int n);
void imprimir (int arreglo[],int n);

int main (){
    int nums[NUM], i, total=0;
    leer(nums,NUM);
    imprimir(nums,NUM);
    return 0;
}

void leer (int arreglo[],int n){
    int i;
    for(i=0; i<n; i++){
        printf("Por favor, introduzca un numero: ");
        scanf("%d", &arreglo[i]);
    }
}

void imprimir (int arreglo[],int n){
    int i;
    printf("\nLista de numeros: ");
    for(i=0; i<n; i++){
        printf("%d ", arreglo[i]);
    }
}
```

Bibliografía

- [1] Luis Joyanes Aguilar and Ignacio Zahonero Martínez, *Programación en C, Metodología, algoritmos y estructuras de datos*, Segunda Edición. Mc Graw Hill.
- [2] Mark Burgess, *The GNU C Programming Tutorial*, Ed. 4.1. Faculty of Engineering, Oslo College.