

**¿Que es un arreglo?**

Un arreglo es un tipo de dato estructurado que permite guardar colecciones de elementos del mismo tipo.

**Declaración de arreglos**

Para declarar un arreglo se utiliza el siguiente formato:

```
tipo nombre_arreglo [tamaño];
```

donde tipo es el tipo de los datos que almacenará el arreglo. Es importante mencionar que se pueden declarar arreglos de los tipos

primitivos de C++, o bien de tipos definidos por el usuario.

Tamaño representa la cantidad de casillas que se reservan para el arreglo. En C++ todos los arreglos empiezan en el subíndice 0

y llegan al subíndice tamaño-1.

Por ejemplo:

```
int arr[6]; // arreglo de 6 elementos enteros, los subíndices van del 0 al 5
```

```
char cad[10]; // arreglo de 10 elementos de tipo caracter, los subíndices van del 0 al 9.
```

**Otra forma de declarar arreglos:**

También es posible declarar un arreglo declarando primero un tipo con typedef de la siguiente forma:

```
typedef tipo nombre_tipo[tamaño]; // declaración del tipo
```

```
nombre_tipo Nombre_variable; // declaración de la variable de tipo arreglo
```

Por ejemplo:

```
typedef int arr8[8];  
arr8 lista;
```

En este caso se declara el tipo de dato arr8 que es un arreglo de 8 elementos enteros, cuyos subíndices van del valor 0 al 7. En el

segundo renglón de la declaración se declara el arreglo del tipo arr8.

### **Uso de los elementos del arreglo**

Para usar los elementos individuales de un arreglo se usa el siguiente formato:

arreglo[subíndice]

Como un elemento de un arreglo es un dato, se puede usar como cualquier variable de ese tipo; Enseguida se pueden ver algunos ejemplos:

```
int arr[4];
arr[3] = 8;
cin>>arr[2];
cout<<arr[3];
arr[0] = arr[1] + arr[2];
int k = 2;
arr[k+1] = 20;
```

### **Ejemplo:**

```
#include <iostream.h>

int main()
{
    int k, arreglo[20];

    // coloca en cada casilla el valor de su subíndice
    for (k = 0; k < 20; k++)
        arreglo[k] = k;

    // muestra el contenido del arreglo
    for (k = 0; k < 20; k++)
        cout<<"Casilla "<<k<<" "<<arreglo[k]<<endl;
    return 0;
}
```

### **Inicializar arreglos en la declaración**

En C++ es posible inicializar un arreglo al declararlo; esto se hace colocando un operador de asignación y después entre llaves la

lista de valores para el arreglo separados por comas, veamos los siguientes ejemplos:

```
double arreglo[3] = { 23.5, 54.22, 78};
```

```
char cadena[4] = {?a?, ?b?, ?c?, ?d?};
```

Ahora veamos el siguiente ejemplo:

```
int valores[20] = {2, 4, 6};
```

En este ejemplo se muestra el caso de no definir valores para todos los elementos del arreglo, en este caso los elementos

faltantes se inicializan con 0 (cero) o con el caracter nulo (\0) dependiendo del tipo de dato del arreglo.

También es posible tener este caso:

```
int arreglo[ ] = { 2, 4, 6, 8};
```

En este caso, la cantidad de datos determina el tamaño del arreglo que se está declarando.

### Ejemplo completo:

```
// Ejemplo de arreglos de una dimensión
// Método de ordenamiento de burbuja
#include <iostream.h>

int main()
{
    const int tamArr = 10;
    int lista[tamArr] = {22, 7, 14, 9, 2, 80, 40, 67, 51, 38};
    int tmp;

    // muestra el arreglo original
    for (int i = 0; i < tamArr; i++)
        cout<<lista[i]<<"\t"; //muestra el número y un tab

    //método de ordenamiento de burbuja
    // para controlar la cantidad de pasadas
    for(int pasada = 0; pasada < tamArr-1; pasada++)
        // para controlar el número de comparaciones por pasada
        for (int k = 0; k < tamArr-1; k++)
            if (lista[k] > lista[k+1])
            {
                tmp = lista[k];
                lista[k] = lista[k+1];
                lista[k+1] = tmp;
            }

    // muestra el arreglo ordenado
    cout<<"Los datos ordenados son"<<endl;
    for (int j = 0; j < tamArr; j++)
        cout<<lista[j]<<"\t";

    return 0;
}
```

## Ejercicio

Copia los ejemplos que se incluyen en el material y pruébalos.

Realiza un programa que pida al usuario una lista de números de tipo double y encuentre el menor y su posición y las muestre en la pantalla. [ver solución](#)

## Ligas sugeridas

<http://www.cplusplus.com/doc/tutorial/>

<http://www.cs.wustl.edu/~schmidt/C++/>

[\*Regresar\*](#)

[\*Siguiente módulo\*](#)