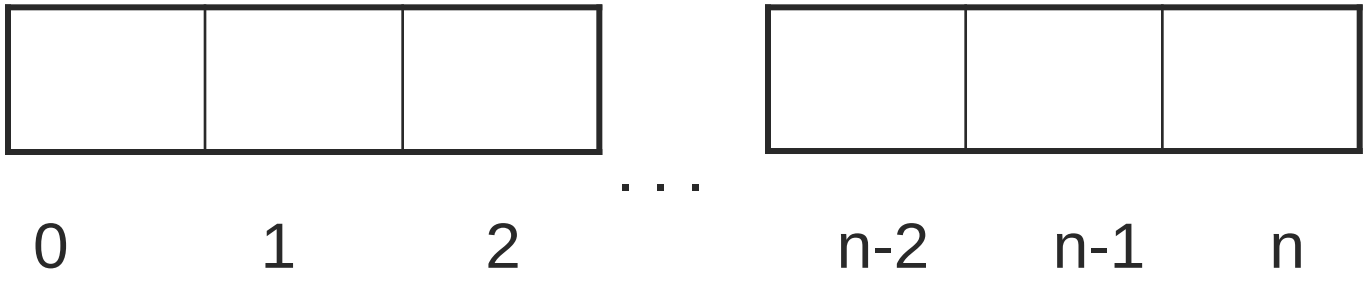
Curso Básico de C
Clase Nº 6

Arreglos Unidimensionales: Vectores
Arreglos Multidimensionales: Matrices
Manejo de Cadenas

Prof. Patricia Guerrero

Arreglos

Un arreglo es un identificador que referencia un conjunto de datos del mismo tipo.



En C los arreglos comienzan en la posición 0 (cero).

El acceso a un arreglo es directo, a través de un índice, que representa un valor entero y positivo.

Arreglos Unidimensionales

- Los arreglos unidimensionales también reciben el nombre de **vectores**. Sólo utilizan un índice para referenciar cada uno de los elementos.

Su declaración se realiza así:

tipo nombre [tamaño];

Cualquiera de los tipos primitivos, o un tipo definido por el usuario

Indica la cantidad de elementos del vector

Arreglos Unidimensionales

- **Ejemplo:**

```
#include <stdio.h>
main()
{
      int vector[10], i;
      for (i=0; i<10; i++)
            vector[i] = i*i;
      for (i=0; i<10; i++)
            printf(" %d", vector[i]);
}
```

Arreglos Unidimensionales

Un vector puede inicializarse al momento de su declaración:

tipo nombre[ ] = { valor1, valor2, ... };

No es necesario especificar el tamaño

Ejemplos:

```
int vector[ ] = {1, 2, 3, 4, 5, 6, 7, 8};
char vector[ ] = "programador";
char vector[ ] = {'p', 'r', 'o', 'g', 'r', 'a', 'm', 'a', 'd', 'o', 'r'};
```

Arreglos Unidimensionales

C no realiza ninguna comprobación de los límites de los arreglos.

Ejemplo:

```
int A[5];          /* A[0], A[1], …, A[4] elemento válidos */

A[8] = 25;         /* A[8] elemento inexistente */
```

Arreglos Unidimensionales

En C no se puede asignar un arreglo completo a otro en una sola sentencia.

```
char A[10], B[10];
.
.
.
B = A;    /* Esto no se puede hacer */
```

Para hacerlo correctamente debe copiarse elemento por elemento, cada uno por separado

Arreglos Unidimensionales

- **Cadenas:**

Una cadena se define como un arreglo unidimensional de caracteres, con un caracter de terminación nulo **'\0'.**

```
char cad[80];
int i;
printf("Introduzca una cadena: \n");
gets(cad);
for(i=0; cad[i]; i++)    //Equivalente a printf("%s", cad)
     printf("%c", cad[i]);
```

Arreglos Unidimensionales

- **Cadenas:**

La biblioteca estándar de C suministra funciones para el manejo de cadenas. Entre las más importantes están:

strcpy(), strcat(), strcmp() y strlen()

Estas funciones requieren el archivo de cabecera **string.h**

Arreglos Unidimensionales

- **Cadenas:**

strcpy(destino, origen);

Copia la cadena origen en destino.

```
char str[20];
strcpy(str, “hola”);
printf(“%s”, str);
```

Copia la cadena **“hola”** en **str**

strcpy() no realiza comprobación de límites

Arreglos Unidimensionales

- **Cadenas:**

strcat(uno, dos);

Concatena la cadena dos a uno.

No realiza comprobación de límites

```
char str[20];
strcpy(str, “Buenos”);
strcat(str, “Dias”);
printf(“%s”, str);
```

Arreglos Unidimensionales

- **Cadenas:**

strcmp(cad1, cad2);

Las cadenas se comparan por orden lexicográfico

Devuelve **cero** si las cadenas son iguales

Compara las cadenas **cad1** y **cad2**.

Devuelve un valor negativo si cad1 es menor que cad2, y un valor positivo si cad1 es mayor que cad2.

Arreglos Unidimensionales

- **Cadenas:**

strlen(str);

No toma en cuenta el caracter de terminación nulo

Devuelve la longitud en caracteres de la cadena **str**.

```
char cadena[] = “programador”;
int longitud = strlen(cadena);
printf(“%d”, longi);
```

Arreglos Multidimensionales

- Los arreglos multidimensionales también reciben el nombre de **matrices.** Representan un arreglo de dos o más dimensiones.

- Su declaración se realiza así:

tipo nombre [tamaño 1][tamaño 2] ... ;

- Una matriz bidimensional se podría representar gráficamente como una tabla con filas y columnas.

Arreglos Multidimensionales

- **Ejemplo:**

```
#include <stdio.h>
main()
{
    int x, i, numeros[3][4];
    for (x = 0; x < 3; x++)      //Lectura de la matriz
        for (i = 0; i < 4; i++)
            scanf("%d",&numeros[x][i]);
    for (x = 0; x < 3; x++)      //Escritura de la matriz
    {
        for (i = 0; i < 4; i++)
            printf("%d ",numeros[x][i]);
        printf("\n");
    }
}
```

Arreglos Multidimensionales

Una matriz también puede ser inicializada al momento de su declaración.

int numeros[4][3]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

numeros[0][0]=1	numeros[0][1]=2	numeros[0][2]=3
numeros[1][0]=4	numeros[1][1]=5	numeros[1][2]=6
numeros[2][0]=7	numeros[2][1]=8	numeros[2][2]=9
numeros[3][0]=10	numeros[3][1]=11	numeros[3][2]=12

Arreglos Multidimensionales

También se pueden inicializar cadenas de texto:

```
char dias[7][10] = { "lunes", "martes", …, "viernes",
                                    "sábado", "domingo" };
```

Para referenciar cada palabra basta con el primer índice:

```
printf("%s", dias[i]);
```

Arreglos Multidimensionales

- **Cadenas:**

C permite crear una tabla de cadenas bidimensional.

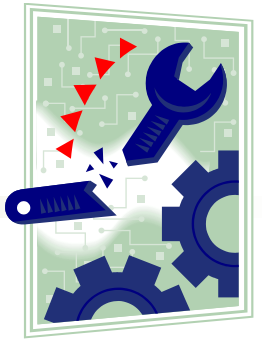
char nombres[10][40];

Para acceder a una cadena sólo debe especificarse el primer índice

Especifica una tabla que puede contener **10 cadenas**, cada una de hasta **40 caracteres**

gets(nombres[2]); //Para leer una cadena
printf(“%s”, nombres[4]); //Para mostrar una cadena

Arreglos



Ejercicios:

1. Dado un arreglo de números enteros determinar si es una Mochila Perfecta, es decir, si para cada elemento del arreglo se cumple que su valor es mayor a la suma de todos los anteriores.

2. Escriba un programa que introduzca una cadena y que después la muestre en pantalla de atrás hacia delante.

3. Escriba un programa que inicialice un arreglo de 10x3 de modo que el primer elemento de cada fila contenga un número, el segundo elemento su cuadrado y el tercero su cubo. Empiece con uno y termine en 10.

Arreglos

Ejercicios:

4. Escriba un programa que cree una tabla de cadenas que contenga las palabras en inglés para los números del 0 al 9. Permita al usuario introducir un dígito y que el programa muestre la palabra equivalente.

5. En estadística, la moda de un grupo de números es aquel que aparece con más frecuencia. Escriba un programa, que permita al usuario introducir una lista de 20 números, y después busque y muestre la moda.

6. Escriba un programa que actúe como diccionario electrónico. Si el usuario introduce una palabra, el programa debe mostrar su significado. Utilice un arreglo de caracteres tridimensional.