LENGUAJE C

Tema 4 – Vectores, Matrices y Cadenas de caracteres.

VECTORES UNIDIMENSIONALES

- Los vectores (array o arreglo) unidimensionales son secuencias de valores del mismo tipo que se almacenan en localidades contiguas de memoria, según el orden del índice.

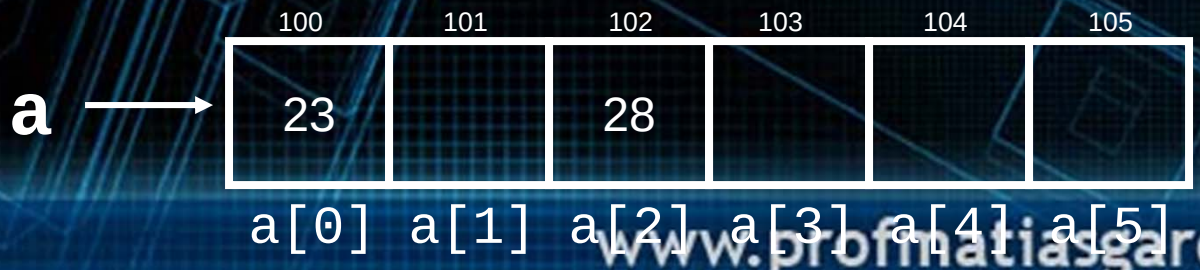
<tipo dato> <identificador>[tamaño];

- Ejemplo:

```
int valores[10];
float datos[5]={1.3, 2.8, 4.89, 0.0, 5.7};
Int numeros[]={1, 5 , 9};
```

- A diferencia de otros lenguajes los arrays en C comienzan por el elemento 0 y terminan en el n-1.

```
int a[6];
a[0]=23;
a[2]=a[0]+5;
for(i=0;i<6;i++) printf("%d",a[i]);
```



VECTORES UNIDIMENSIONALES

- Un vector se identifica por su nombre, pero para el compilador este equivale a la dirección de memoria del primer elemento del vector, es decir:
- Ejemplo:

```
int num[50];
/*Para el compilador:
num  es igual a &num[0]
La dirección del elemento 0 */
```

- Para acceder a un elemento de un vector se debe especificar el nombre del vector, seguido de la posición (el índice) que ocupa dicho elemento dentro del vector entre corchetes.
- El almacenamiento de los elementos de un vector se determina en tiempo de compilación.
- La cantidad de memoria en bytes viene dada por:

```
bytes_totales = sizeof(tipo) * tamaño;
```

Operaciones con vectores

- Los vectores pasados como argumento a funciones siempre se pasan por referencia. (¡Ojo! no sus elementos. Si se quiere pasar un único elemento por referencia hay que usar el &)

```
Sumar(vector);  /* llamada a la función pasando un vector por referencia.*/
```

- Para copiar el contenido de un vector a otro debe copiarse cada elemento uno a uno. No se puede asignar un vector a otro. Lo siguiente es ilegal:

```
int a[10], b[10];
a=b;//error ilegal
```

- En vez, se debe hacer asignaciones por cada elemento:

```
int i;
for(i=0;i<10;i++)a[i]=b[i];
```

- Para acceder a cada uno de los elementos se pueden usar expresiones, siempre que su resultado no exceda de los limites del array.

```
int tabla[8];
int x;
tabla[0]=28;
x=2;
tabla[(3*x)+1] = x*10;
```

MATRICES

- Un vector puede tener N dimensiones, dependiendo de las limitaciones de la memoria, a estos se los llama matrices y su declaración es la siguiente:

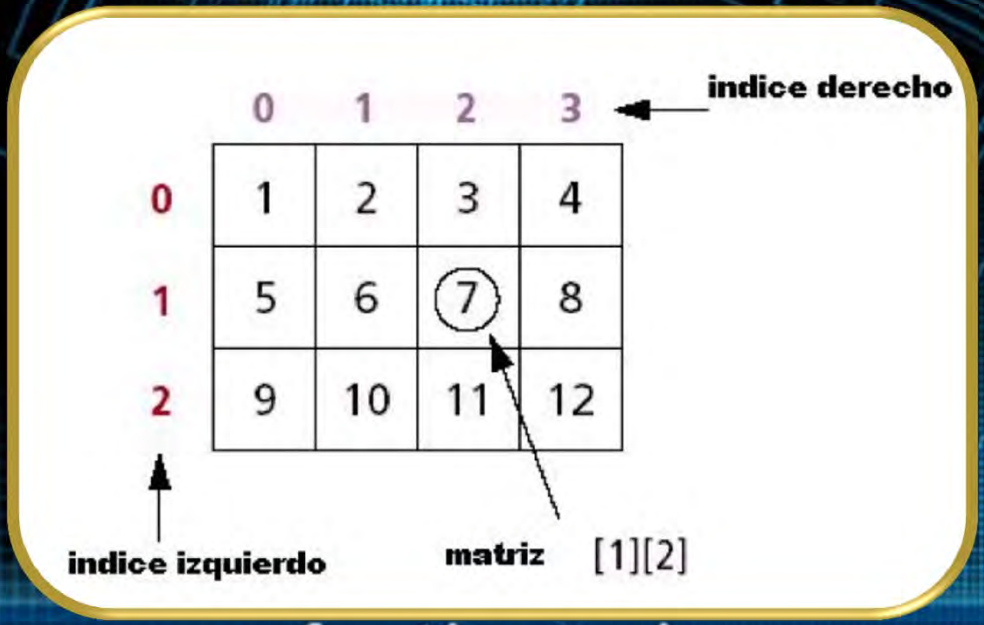
<tipo dato> <identificador>[dim1] [dim2]...[dimN];

- Ejemplo:

```
double cubo[3][3][3];
```

- La siguiente formula determina el numero de bytes que necesita ser asignada: Bytes=fila*col*numero_bytes_tipo;

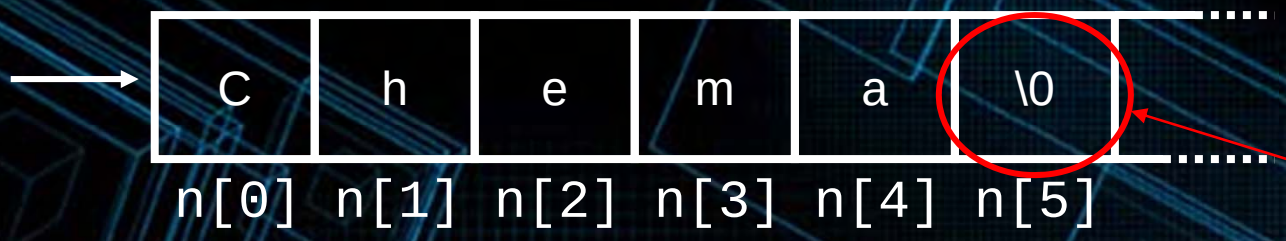
```
#include<stdio.h>
int main()
{
   int fila,col,matriz[3][4];
   for(fila=0;fila<3;fila++)
      for(col=0;col<4;col++)
      matriz[fila][col]=fila*col;
   return 0;
}
```



CADENAS DE CARACTERES

- Una cadena de caracteres o “String” se manipula en C, mediante vectores unidimensionales. Los vectores de caracteres terminan con el carácter nulo ‘\0’
- La única diferencia con los vectores numéricos es que se requiere un carácter adicional para indicar cuando finaliza la cadena de caracteres.
- Ejemplo:

```
char nombre[31]; /* Uso sólo 30, la posición 31 es \0 */
char x[20],n[50]=“Chema”;
/*equivalente a char n[50]={‘C’,’h’,’e’,’m’,’a’,’\0’}*/
```



FUNCIONES PARA CADENAS DE CARACTERES

(Su declaración está en la cabecera string.h)

- strlen devuelve la longitud de cadena (sin incluir el '\0').

```
longitud = strlen(cadena);
```

- strcpy copia la cadena origen en la cadena destino.

```
strcpy(destino, origen);
```

- strcat añade la cadena origen a la cadena destino.

```
strcat(destino, origen);
```

- strcmp compara dos cadenas y devuelve un número que será cero si son iguales, positivo si el primer carácter es diferente mayor (alfabéticamente) en cadena1 o negativo si este primer carácter es mayor en cadena2.

```
resultado = strcmp(cadena1, cadena2);
```

- La asignación de strings por medio del operador (=) sólo es posible en la declaración.

```
char n[50]="Chema";  /* Correcto */
n="Otro nombre";     /* Error: no es declaración */
Strcpy(n, "otro nombre"); /* Correcto */
```

Funciones para cadenas de caracteres

- Recordemos que podemos utilizar gets para ingresar cadenas de caracteres y puts para poder imprimirlas incluyendo en ambas funciones los espacios y saltos de línea.
- Una matriz de caracteres es un vector de vectores de caracteres, es decir un vector de cadenas.
- Ejemplos:

```
char meses[12][11]={"Enero", "Febrero",
   "Marzo“, "Abril", "Mayo", "Junio",
   "Julio", "Agosto“, "Septiembre",
   "Octubre", “Noviembre“, "Diciembre"};
char nombres[100][50];
printf("Dame el primer nombre: ");
gets(nombres[0]);
```

FUNCION STRCHR

```
#include <stdio.h>
#include <string.h>
int main()
{
   char direccion[50];
   int caracter = ‘M’;
   char *punt; /* Apuntara al
valor devuelto por strchr */
   strcpy(direccion, “La Madrid
125”);
   punt = strchr(direccion,
caracter);
   if (punt != NULL)
       printf(“El carácter %c
esta en la cadena %s\n”, *punt,
direccion);
   else
       printf(“El carácter %c
no se ha encontrado.\n”,
caracter);
   return 0;}
```

WEBGRAFÍA & LICENCIA:

- Textos tomados, corregidos y modificados de diferentes páginas de Internet, tutoriales y documentos, entre los que destaco el libro: *C/C++ Curso de programación*, 2da Ed, Javier Ceballos, Alfaomega Ra-Ma.
- Este documento se encuentra bajo Licencia Creative Commons 2.5 Argentina (BY-NC-SA), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del Prof. Matías E. García y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- Autor:

Matías E. García

Prof. & Tec. en Informática Aplicada
www.profmatiasgarcia.com.ar
info@profmatiasgarcia.com.ar