

Lenguaje C

Manejo de Arreglos

Algoritmos y Programación I
Lic. Gustavo Bianchi

1

Arreglos en C

Conceptos y Características

Qué es un arreglo en C?

Colección finita y ordenada de datos

homogéneos → Todos del mismo tipo

Estructura estática (en C)

→ Se define en tiempo de diseño
No puede ser redimensionado

Acceso directo a un elemento del arreglo

→ A través de su
posición en el arreglo

Clasificación

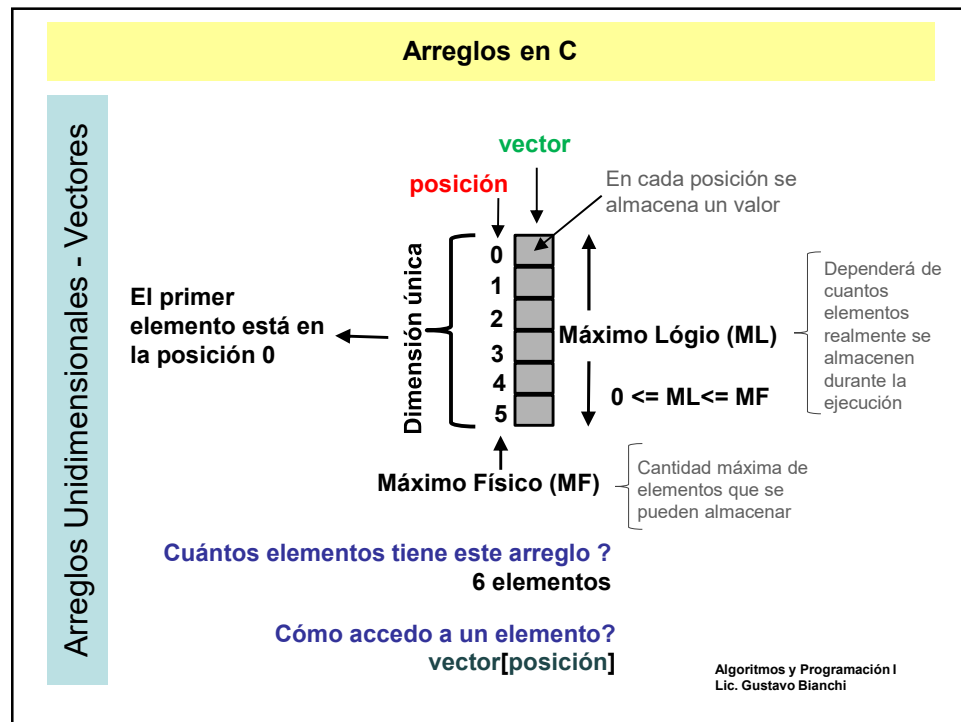
Unidimensionales → Vectores - Listas

Bidimensionales → Matrices - Tablas

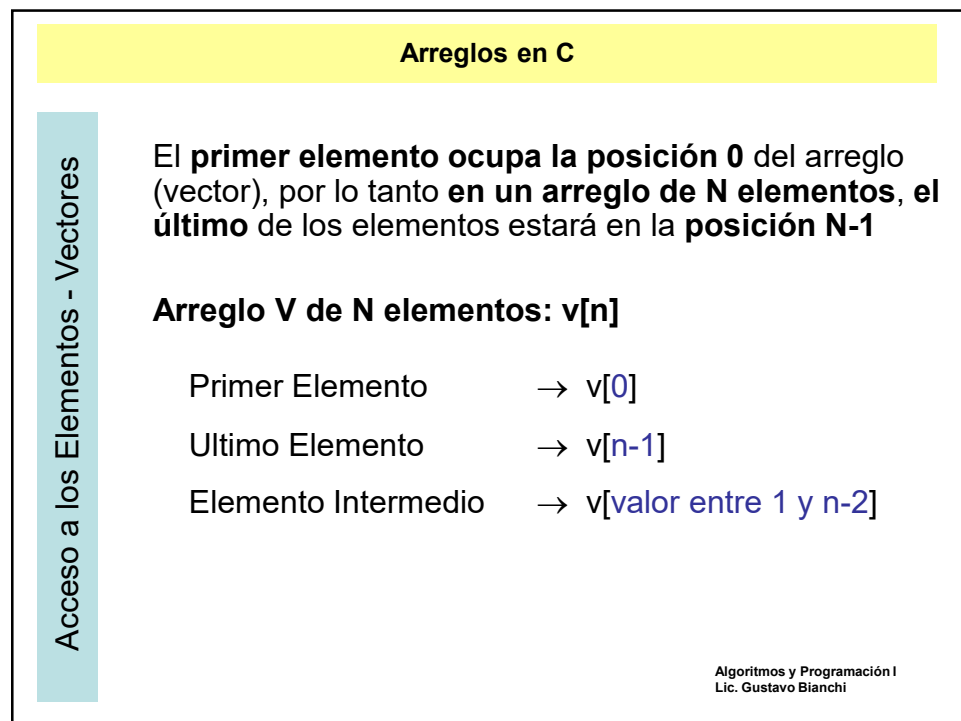
Multidimensionales → Dimension N

Algoritmos y Programación I
Lic. Gustavo Bianchi

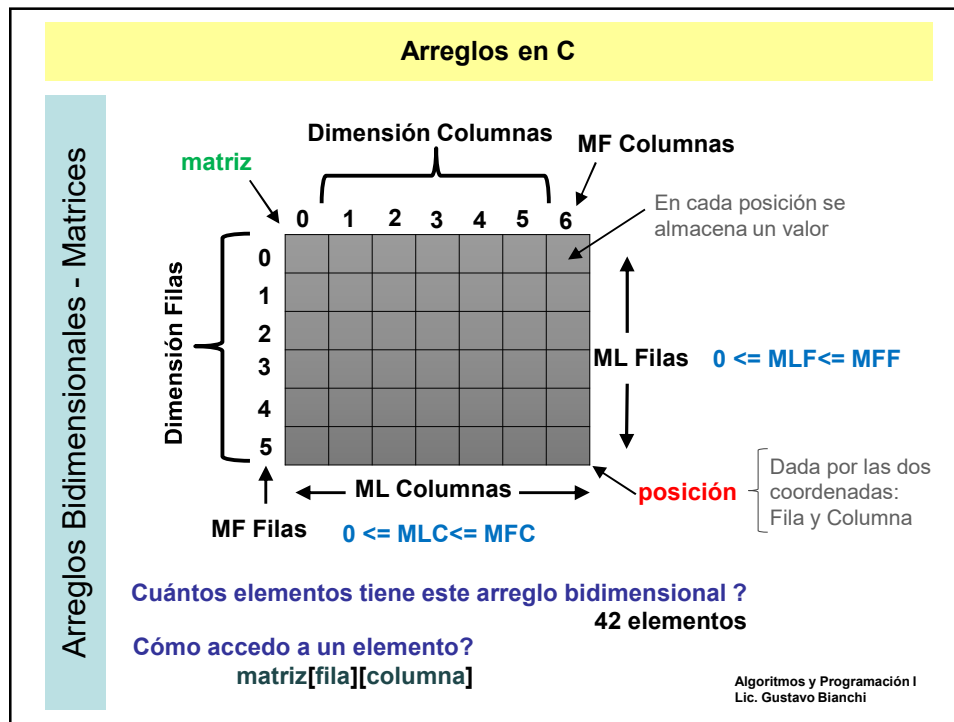
2



3



4



5

Arreglos en C

Acceso a los Elementos - Matrices

El **primer elemento** ocupa la **posición 0,0** del arreglo (matriz), por lo tanto **en un arreglo bidimensional de NxM elementos, el último** de los elementos estará en la **posición [N-1 , M-1]**

Arreglo M de NxM elementos: `m[n][m]`

Primer Elemento → `m[0][0]`

Ultimo Elemento → `m[n-1][m-1]`

Elemento Intermedio → `m[valor entre 1 y n-2][valor entre 1 y m-2]`

Algoritmos y Programación I
Lic. Gustavo Bianchi

6

Arreglos en C

Declaración

Cómo se declaran?

Debemos **indicar, tipo** de los elementos, un **nombre** y la **cantidad** de elementos en cada dimensión

tipo nombre_arreglo[cantidad_elementos]

Arreglos Unidimensionales

arreglo de 100 elementos enteros

int v_nros[100]

arreglo de 100 elementos reales

float v_reales[100]

arreglo de 25 elementos tipo caracter

char v_chars[25]

arreglo declarado por extensión

int v_pares[] = {2,4,6,8,10}

Arreglos bidimensionales

Matriz de 5 filas por 4 columnas de enteros

int m_enteros[5][4]

Matriz de 2 filas por 3 columnas de enteros

int m_enteros[3][3] = {{1,2,3},{4,5,6}}

Algoritmos y Programación I
Lic. Gustavo Bianchi

7

Arreglos en C

Restricciones

- **Los parámetros de tipo arreglo serán pasados siempre por referencia**, sin necesidad de indicarlo con el símbolo &.
- **No se pueden asignar ni comparar arreglos directamente**, hay que asignar o comparar sus elementos uno a uno.

vec_1 = vec_2 → for (i=0; i<100; i++) vec_1[i] = vec_2[i]

NO

Algoritmos y Programación I
Lic. Gustavo Bianchi

8

Arreglos en C

Declaración de tipos de datos

Es conveniente declarar tipos de datos para nuestros arreglos.

La sintaxis para declarar un tipo es la siguiente:

typedef tipo_dato nombre_del_tipo

Beneficio:
Cada vez que hagamos un cambio en la declaración del tipo, el mismo se reflejará automáticamente en todas aquellas variables que se hayan declarado con ese tipo.

Nota:
La declaración de tipos, no aplica solamente a los arreglos, es una buena práctica que se puede aplicar a cualquier tipo de dato de los vistos.

Algoritmos y Programación I
Lic. Gustavo Bianchi

9

Arreglos en C

Declaración de tipos de datos

Ejemplos:

typedef float tv_reales[30]

{

Nuevo tipo para declarar un vector de 30 números reales

typedef int tm_enteros[5][7]

{

Nuevo tipo para declarar una matriz de 5 filas por 7 columnas de valores enteros

Entonces si ahora queremos declarar un vector de 30 números reales utilizaríamos el tipo tv_reales:

tv_reales v_nros_reales

De igual modo si quisiéramos declarar una matriz de 5 filas por 7 columnas de números enteros:

tm_enteros m_nros_enteros

Algoritmos y Programación I
Lic. Gustavo Bianchi

10

Arreglos en C

Declaración de constantes para el MF

También es una buena práctica **declarar constantes** para los **Máximos Físicos de los arreglos**.

Veamos un ejemplo usando la declaración de una constante para el MF, y la declaración de un tipo para el arreglo.

Ejemplo:

```
# define MF 10                //maximo fisico del arreglo
typedef int t_vec[MF];        //arreglo de valores enteros
```

Entonces, más adelante en el programa podríamos utilizar el tipo y la constante en distintos casos:

```
int cargar_datos(t_vec v, int mf)
void mostrar_datos(t_vec v, int ml)
ml = cargar_datos(v_nros, MF);
```

} Uso del tipo t_vec en la firma de las funciones para declarar parámetros
 Uso de la constante MF como parámetro actual

Algoritmos y Programación I
Lic. Gustavo Bianchi

11

Arreglos en C

Mostrando el contenido de un arreglo

Si es necesario **recorrer todo el arreglo**, por ejemplo para mostrar los valores, es conveniente usar un **ciclo for**

Caso 1: muestra los valores almacenados en el vector v

```
for (i=0; i<=ultima_pos; i++)
    printf("%i \n", v[i]);
```

Caso 2: muestra los valores almacenados en la matriz mat, de dimensiones fil x col

```
for (i=0; i<fil; i++){
    for (j=0; j<col; j++)
        printf("%4i ", mat[i][j]);
    printf("\n");
}
```

Algoritmos y Programación I
Lic. Gustavo Bianchi

12