Matrices

Material Original: Prof. Flor Narciso
Modificaciones: Prof. Andrés Arcia
Departamento de Computación
Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes

Tipos de Arreglos

- Vectores (arreglos unidimensionales - 1D)
- Matrices (arreglos bidimensionales - 2D)
- Multidimensionales (tres o mas dimensiones)

Notación Algoritmica de los Arreglos multidimensionales

tipo_dato nombre[dim_1, dim_2, ..., dim_n]

Las matrices (arreglos de dos dimensiones) son un caso particular de los arreglos multidimensionales.

tipo_dato nombre[dim_1, dim_2]

Matriz (Arreglo Bidimensional)

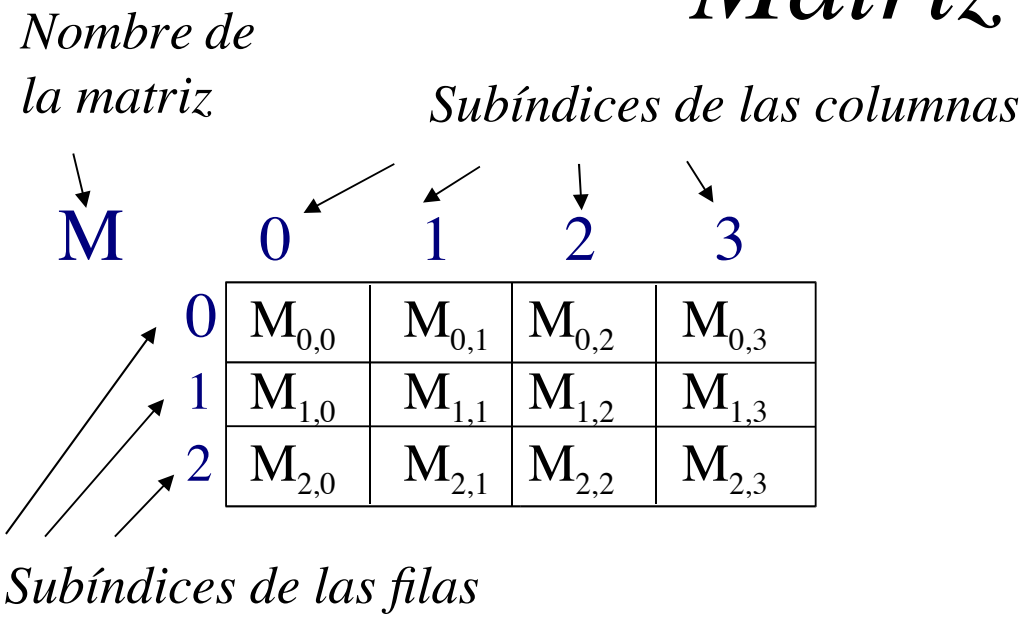
Grupo de localidades consecutivas de memoria relacionadas por el hecho que tienen el mismo nombre y tipo (matrices de enteros, matrices de reales, matrices de caracteres, etc.).

Cada localidad (o grupo de localidades) representa un elemento de la matriz.

Cada elemento de la matriz es accedido mediante el *nombre* de la matriz y dos *subíndices (fila, columna)*, uno que representa la posición numérica (entero no negativo) de dicho elemento en dentro de una fila y el otro que representa la posición numérica (entero no negativo) de dicho elemento dentro de una columna.

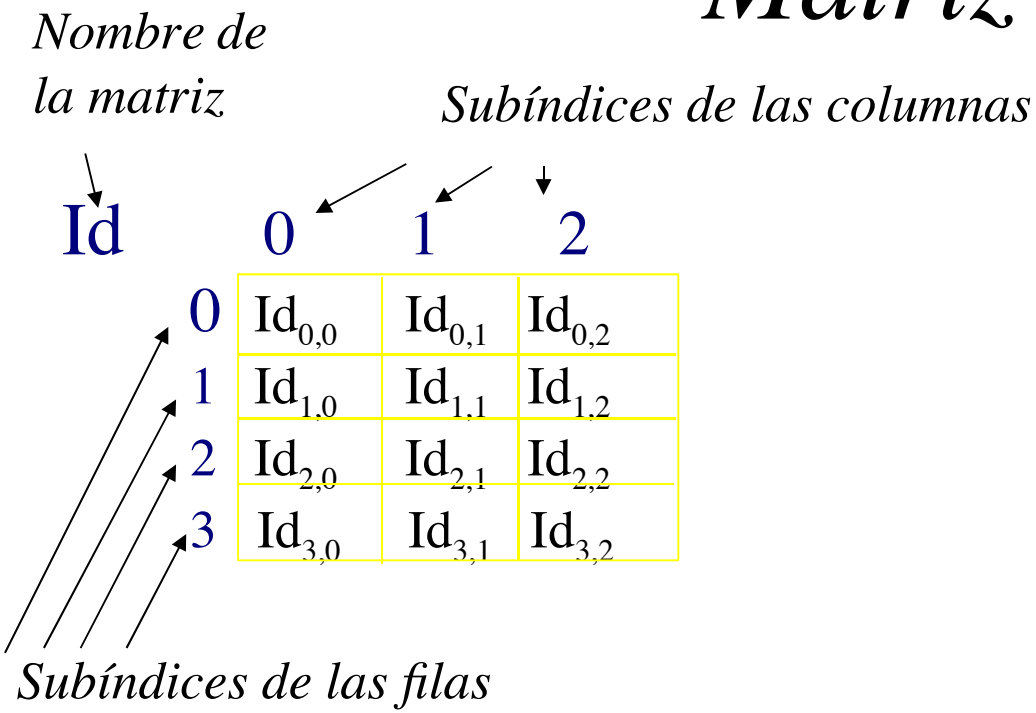
```
nombre_matriz[fila, columna]
```

Representación Gráfica de una Matriz



M_{3x4}: Matriz de tres (3) filas y cuatro (4) columnas cuyo nombre es M

Representación Gráfica de una Matriz



Id_{4x3}: Matriz de cuatro (4) filas y tres (3) columnas cuyo nombre es Id

Declaración de una Matriz: Notación Algorítmica

```
tipo_dato  nombre_matriz[número_filas, número_columnas]
```

Ejemplos en notación algoritmica:

`entero A[12, 4]` Matriz `A` de números enteros de 12 filas y 4 columnas

`caracter cdn[8, 2]` Matriz `cdn` de caracteres de 8 filas y 2 columnas

`real b[100,100]` Matriz `b` de números reales de 100 filas y 100 columnas

Declaración de una Matriz: Notación en C

Ejemplos:

```
int A[12][4];
char cdn[8][2];
float b[100][100], x[27][27];

int matrix[3][6] = {{16, 21, 8, 3, -7, 9},
                    {-3, 11, 0, 5, 9, 7},
                    {13, 7, -64, 19, 14, 2}}
```

	0	1	2	3	4	5
0	16	21	8	3	-7	9
1	-3	11	0	5	9	7
2	13	7	-64	19	14	2

Declaración de una Matriz: Notación en C

Ejemplos:

```
int m1[3][4] = {{0, 1, 2},
                {-1, -2, -3},
                {3, 4, 5}}
```

	0	1	2	3
0	0	1	2	0
1	-1	-2	-3	0
2	3	4	5	0

Esta definición asigna valores solo a los tres primeros elementos de cada fila, el cuarto elemento es inicializado con el valor cero (0).

Rellenado de un vector n-dim

Para rellenar un vector n-dimensional es necesario comprender la disposición de los llaves.
Las llaves van dispuestas de forma concéntrica, empezando desde la más externa hasta llegar a la dimensión más interna ó la de más a la derecha en la declaración.

Ejemplo:

Para un vector n-dim:

tipo vector[i][j][k] = {

0 { { a_{01}, a_{02}, …, a_{0k-1} }, { a_{11}, a_{12}, …, a_{1k-1}}, … {a_{j1}, a_{j2},…,a_{jk-1}} }

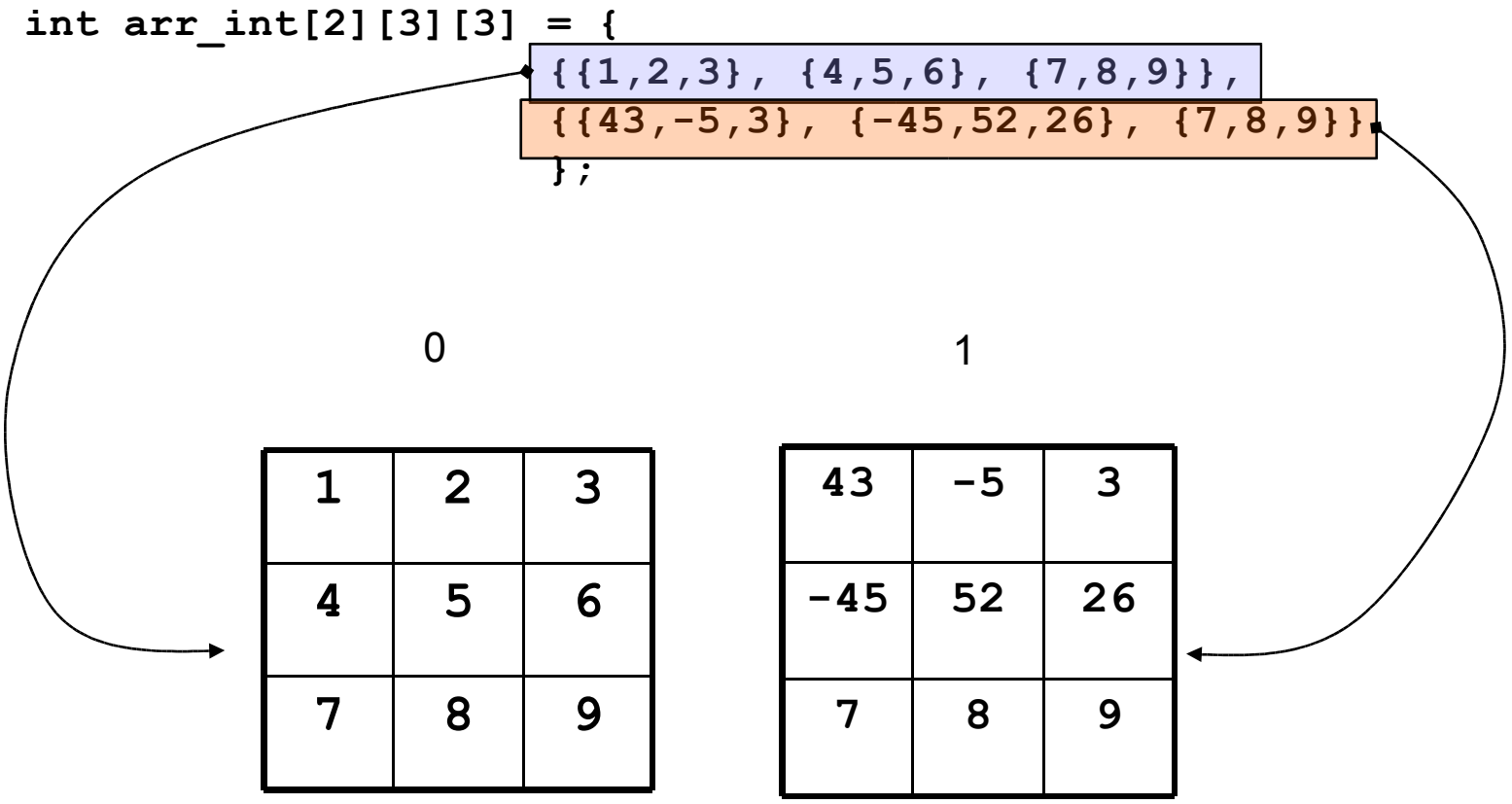
1 { { a_{01}, a_{02}, …, a_{0k-1} }, { a_{11}, a_{12}, …, a_{1k-1}}, … {a_{j1}, a_{j2},…,a_{jk-1}} }

…

i-1 { { a_{01}, a_{02}, …, a_{0k-1} }, { a_{11}, a_{12}, …, a_{1k-1}}, … {a_{j1}, a_{j2},…,a_{jk-1}} }

}

Ejemplo (vector n-dim)



Matrices: Acceso en C

Cada elemento de la matriz es accedido mediante el nombre de la matriz y la posición numérica de dicho elemento dentro de la matriz (subíndices).

nombre_matriz[fila][columna]

Ejemplo:

matrix

	0	1	2	3
0	16	21	8	3
1	-3	11	0	5
2	13	7	-64	19

matrix[0][0] = 16 matrix[0][1] = 21 matrix[0][2] = 8 matrix[0][3] = 3
matrix[1][0] = -3 matrix[1][1] = 11 matrix[1][2] = 0 matrix[1][3] = 5
matrix[2][0] = 13 matrix[2][1] = 7 matrix[2][2] = -64 matrix[2][3] = 19

Matrices: Dos Subíndices

Posición de un elemento dentro de la matriz:

- El primer subíndice corresponde a la posición del elemento con respecto a las filas, cuyo valor puede variar de 0 a n-1, donde n es el número de filas.
- El segundo subíndice corresponde a la posición del elemento con respecto a las columnas, cuyo valor puede variar de 0 a m-1, donde m es el número de columnas.

Cada subíndice puede ser una constante entera, una variable entera o una expresión entera (valor mayor o igual que cero).

Matrices: Instrucciones Válidas

```
A[1][1] = 3;
X = A[i][j+1];
printf("%i",A[2][j]);
printf("%i",A[0][1]);
A[1][1] = A[2][1]; // un par de variables enteras
suma(A[1][1], b, c); // suma recibe 3 parametros enteros
printf("%i %i %i",A[i][j,A[i+1][j+1],A[i+2][j+2]);
b = A[0][3] / 2;
```

Notese que cualquier elemento de una matriz puede usarse como una variable simple.

Matrices: Inicialización

- Inicializar los elementos de una matriz de 7 filas y 3 columnas de elementos enteros en cero.

```
float matriz[7][3];
int i,j;
// Declaracion de la matriz y subindices
……….
for (i = 0; i < 7; i++)
          for (j = 0; j < 3; j++)
              matriz[i][j] = 1.0;       // Inicializacion de cada
                                        // elemento de la matriz en 0
```

Todos los elementos de la matriz tendrán el valor uno (1.0)

Matrices: Inicialización

- Inicializar los elementos de una matriz de 2 filas y 2 columnas de tipo carácter con valores introducidos por el usuario.

```
char C[2][2];        // Declaracion de la matriz
int ind1, ind2;      // Declaracion de los subindices

……….

for (ind1 = 0; ind1 < 2; ind1++)
     for (ind2 = 0; ind2 < 2; ind2++)
{
     printf("Introduzca un caracter\n");
     getchar(C[ind1][ind2]);
}
```

Matrices: Ejemplo 1

Escribir los elementos de una matriz

```
void EscribirMatriz( )
{
   int n[3][3] = {{32, 27, 64}, {18, 95, 14},
                  {90, 70, 60}}, i, j;

   for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
           printf("Elemento (%i,%i) = %i",i+1,j+1,n[i][j]);
}
```

Matrices: Ejemplo 2

Almacenar números en una matriz, escribir los elementos de la matriz, calcular y escribir la suma de los elementos de cada fila, calcular y escribir la suma de los elementos de cada columna.

	0	1	2
0	A_{00}	A_{01}	A_{02}
1	A_{10}	A_{11}	A_{12}
2	A_{20}	A_{21}	A_{22}
3	A_{30}	A_{31}	A_{32}

	sumaFilas
0	$A_{00} + A_{01} + A_{02}$
1	$A_{10} + A_{11} + A_{12}$
2	$A_{20} + A_{21} + A_{22}$
3	$A_{30} + A_{31} + A_{32}$

0	$A_{00}+A_{10}+A_{20}+A_{30}$	sumaColumnas
1	$A_{01}+A_{11}+A_{21}+A_{31}$	
2	$A_{02}+A_{12}+A_{22}+A_{32}$	

Matrices: Ejemplo 2

```
#include <iostream.h>
#define MAXFILA 20
#define MAXCOL  30

void leerDatos(int ma[MAXFILA]
   [MAXCOL],
   int &n, int &m)
{
   int fila, col;

   printf("Numero de filas?");
   scanf("%i",&n);
   printf("Numero de
   columnas?\n");
   scanf("%i",&m);
```

```
   for (fila=0; fila<n; fila++)
      for (col=0; col<m; col++) {
      printf("Valor ?\n");
       scanf("%i",&ma[fila][col]);
   }
}

void sumarFilas(int ma[][], int n,
   int m, int vsf[]) {
   int fila, col;

   for (fila=0; fila<n; fila++)
      for (col=0; col<m; col++)
       vsf[fila] += a[fila][col];
}
```

Matrices: Ejemplo 2

```
void sumarCol(int ma[MAXFILA]
    [MAXCOL], int &n,
                int &m, int vsc[]) {
   int fila, col;

   for (col=0; col<m; col++)
      for (fila=0; fila<n; fila++)
        vsc[col] += ma[fila][col];
}
```

```
void main () {
int matriz[MAXFILA]{MAXCOL],
     vectorSumaF[MAXFILA] = {0},
     vectorSumaCol[MAXCOL] = {0},
     numFilas, numCol;

     leerDatos(matriz, numFilas,
   numCol);
     sumarFilas(matriz, numFilas,
           numCol, vectorSumaF);
     sumarCol(matriz, numFilas,
   numCol, vectorSumaCol);
}
```

Matrices: Ejemplo 3

Cada semana, el gerente de una tienda local de artefactos domésticos registra las ventas de los artículos individuales que hay en existencia. A final del mes, estos resúmenes semanales se envían a la oficina central, donde se analizan. Un ejemplo de un mes típico se muestra en la siguiente tabla.

Matrices: Ejemplo 3

		Artefacto		
		Lavadoras	Secadoras	Cocinas
Semana	1	5	4	8
	2	7	7	10
	3	5	3	7
	4	8	10	15

Calcular:

El número total de artefactos vendidos cada semana.
El número total de cada tipo de artefacto vendido en el mes.

Matrices: Ejemplo 3

```
void LeerDatos(int m[MAXF][MAXC], int filas, int columnas) {
   int i, j;

   for (i = 0; i < filas; i++)
      for (j = 0; j < columnas; j++) {
          switch (j) {
            case 0: printf("Lavadoras vendidas en la semana %i\n",i+1);
                    break;
            case 1: printf("Secadoras vendidas en la semana %i\n",i+1);
                   break;
            case 2: printf("Cocinas vendidas en la semana %i\n",i+1);
                    break;
           }
         scanf("%i", &m[i][j]);
       }
}
```

Matrices: Ejemplo 3

Tarea: Codificar el resto del programa.

Arreglos Multidimensionales

double registros [100][66][255];

Arreglo con 100 x 66 x 255 = 1.683.000 elementos

Ejemplo:

```
int tridi[10][20][30] = {{ {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 1, 2} },
                         { {2, 4, 5, 6}, {7, 8, 9, 0}, {13, 14, 15, 16} }
                         };
```

tridi[0][0][0] = 1	tridi[0][0][1] = 2	tridi[0][0][2] = 3	tridi[0][0][3] = 4
tridi[0][1][0] = 5	tridi[0][1][1] = 6	tridi[0][1][2] =7	tridi[0][1][3] = 8
tridi[0][2][0] = 9	tridi[0][2][1] = 10	tridi[0][2][2] = 1	tridi[0][2][3] = 2

Matrices: Ejercicios

Describir la salida producida por el siguiente programa:

```
#include <stdio.h>

#define FILAS 3
#define COLUMNAS 4

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

main() {
   int a, b, c;
   for (a = 0; a < FILAS; a++) {
         c = 999;
         for (b = 0; b < COLUMNAS, b++)
              if (z[a][b] < c)
                   c = z[a][b];
         printf("%i \n",c);
     }
}
```

Matrices: Ejercicios

La transpuesta de una matriz se obtiene intercambiando simplemente las filas por columnas y viceversa. Diseñar un programa utilizando funciones para leer una matriz y obtener su transpuesta.

Un examen final tiene 100 preguntas de selección múltiple. Cada pregunta tiene cinco respuestas a elegir, de las cuales sólo una es correcta. Los resultados de la información concerniente al estudiante pueden representarse de la siguiente forma:

RESPUESTAS: Contiene la respuestas correctas del examen codificadas del 1 al 5

EXAMENES: Matriz cuyas filas son las respuestas de las 100 preguntas de selección múltiple dadas por n estudiantes.

Calcular la nota de cada estudiante.

Matrices: Ejercicios

Escribir un programa que lea dos matrices de enteros y calcule la suma de los elementos correspondientes, esto es,

$$c_{ij} = a_{ij} + b_{ij}$$

Un colegio funciona en un edificio de tres pisos, cada uno con cinco salones de varios tamaños. Cada año, el colegio debe asignar clases a los salones del edificio. Dada una lista que consiste en n clases identificadas por un número entero y su tamaño (# de estudiantes), dada además la capacidad de cada salón, escribir un programa que consiga e imprima una asignación de salones satisfactoria tal que todas las clases tengan un salón asignado. Para aquellas clases que no se encuentre un salón, el programa debe imprimir un mensaje “Salón no disponible”.

Matrices: Ejercicios

El programa deberá generar una lista de salones no asignados junto con su respectiva capacidad. Correr el programa en frío para los siguientes datos de entrada:

- Entradas *Número del salón*

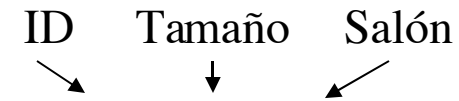
Piso

	0	1	2	3	4
0	30	30	15	30	40
1	25	30	25	10	110
2	62	30	40	40	30

El número del salón viene dado por la fórmula i*100+j, donde i es el subíndice correspondiente a la fila y j es el subíndice correspondiente a la columna.

Matrices: Ejercicios

- Entradas



	0	1	2
0	11	37	
1	12	55	
2	13	100	
3	110	26	
4	111	26	
5	125	39	
6	130	30	
7	131	45	

Matrices: Ejercicios

Escribir un procedimiento
void calendario (int v[], Cadena dia, int mes,
int c[][])

que reciba como parámetros:

- Un vector de enteros que indica el número de días que tiene cada mes del año.
- Una cadena que indica el día de la semana del primer día de un año.
- Un entero comprendido entre 1 y 12 indicando un mes del año.

y devuelva una matriz de 6 semanas por siete días que represente el calendario de dicho mes de dicho año. Por ejemplo, siendo

v ={31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}

Matrices: Ejercicios

la llamada

Calendario (v, “domingo” 6, t)

devuelve en t la tabla

lunes	martes	miércoles	jueves	viernes	sabado	domingo
0	0	0	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	0	0
0	0	0	0	0	0	0

Matrices: Ejercicios

Escribir un programa que genere una tabla de valores para la ecuación

$$y = 2e^{-0.1t}\text{sen } 0.5t$$

donde t varía entre 0 y 60. Permitir que el valor del incremento t sea introducido como un parámetro de entrada.

Matrices: Ejercicios

Considere la siguiente lista de estados y sus capitales:

Mérida	Mérida
Anzoategui	Barcelona
Monagas	Maturín
Nueva Esparta	La Asunción
Zulia	Maracaibo
Carabobo	Valencia

Escribir un programa interactivo que acepte el nombre de un estado como entrada y escriba su capital y viceversa. Diseñar el programa de modo que se ejecute repetidamente, hasta que se introduzca la palabra FIN.