Tema 7: Vectores y matrices

Departamento de Automática

Universidad de Alcalá







1/14

Introducción a las matrices Vectores numéricos Cadenas de caracteres Matrices

Índice

- Introducción a las matrices
- 2 Vectores numéricos
 - Definir un vector
 - Acceder a un vector
 - Acceder a un vector
- Cadenas de caracteres
 - Definición de una cadena de caracteres
 - Trabajar con cadenas de caracteres
- Matrices
 - Concepto de matriz
 - Definición de matrices

Introducción a las matrices

Concepto de arreglo (array)

- Un arreglo (o array) es un conjunto de datos del mismo tipo
 - Todo el arreglo tiene un nombre
 - Cada elemento se accede mediante un índice
 - Los datos se ordenan en memoria linealmente
- Dos tipos de arreglos:

Vector

Los elementos se ordenan unidimensionalmente

$$v = [a_0, a_1, a_2, ..., a_n]$$

Un único índice: v[i]

Matriz

Los elementos se ordenan en más de una dimensión

$$m = \begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m0} & a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Dos índices: v[i][j]

3 / 14

Vectores numéricos
Cadenas de caracteres
Matrices

Definir un vector Acceder a un vector Acceder a un vector

Vectores numéricos

Definir un vector

- La definición es análoga a una variable
 - Definir antes que usar
 - Necesita un tipo, que puede ser cualquiera
- Los corchetes indican matriz
 - Los paréntesis una función
 - Las llaves un bloque de código
- Se pueden inicializar los valores de un vector

Formato definición

tipo nombre[tamaño];

Ejemplos

int vector[10];
float vector2[5];
double vector3[500];

int
$$v[] = \{5, -2, 2\};$$

Vectores numéricos

Acceder a un vector

- Un elemento del vector es una variable
 - El acceso es análogo a las variables regulares
 - Mismos operadores y usos
- Hay que añadir el índice al vector
 - Empiezan en 0
 - Limitados por el tamaño
- Extremar cuidado con índices
 - Un índice negativo o mayor al tamaño cuelga el programa

```
Ejemplos
int v[100], i=0, temp;
// ...
v[i-2] = 33;
temp = v[51];
temp = v[i] + v[i+1];
v[i] = v[i+1];
v[i]++;
```

◆□▶ ◆□▶ ◆■▶ ■ りへ©

5 / 14

Vectores numéricos
Cadenas de caracteres
Matrices

Definir un vector Acceder a un vector Acceder a un vector

Vectores numéricos

Trabajo con vectores

- Los vectores se suelen manicupar dentro de bucles
- El tamaño máximo suele venir dado por constantes
- Al índice se le nombra con la letra i

```
#define TAM_MAX 100

int main() {
   int v[TAM_MAX], i;

   for (i=0; i<TAM_MAX; i++) {
      scanf("%d", &v[i]);
   }

   for (i=0; i<TAM_MAX; i++) {
      printf("%d", v[i]);
   }
}</pre>
```

Cadenas de caracteres

Definición de una cadena de caracteres (I)

- Cadena de caracteres es un vector de caracteres
- El tamaño de las cadenas puede variar...
 - ... pero los vectores no
 - El fin de cadena con '\0'
- Dos funciones útiles
 - gets(): Leer una cadena
 - puts(): Imprimir una cadena

```
Equivalencias
```

```
char cad1[]= { 'a', 'b', 'c', '\0'};
char cad2[]= {97, 98, 99, 0};
char cad3[]= "abc";

char cad4[4];
cad4[1] = 'a';
cad4[2] = 'b';
cad4[3] = 'c';
cad4[4] = '\0';

printf("%", cad1);
```



7 / 14

Vectores numéricos

Cadenas de caracteres

Matrices

Definición de una cadena de caracteres Trabajar con cadenas de caracteres

Cadenas de caracteres

Definición de una cadena de caracteres (II)

```
#define TAM_MAX 101

int main() {
    char cadena[TAM_MAX];

    while(((car = getchar()) != '\n') && i <(TAM_MAX-1)) {
        cadena[i] = car;
        i++;
    }

    cadena[i] = '\0';
    printf("%", cadena);

    return 0;
}</pre>
```

Cadenas de caracteres

Trabajar con cadenas de caracteres (I)

<pre>char* strcpy(char *cad1, char *cad2)</pre>	Copia cad2 en cad1
<pre>int strcmp(char *cad1, char *cad2)</pre>	Compara cad2 y cad1
int strlen(char *cad)	Longitud cad
char tolower(char car)	Convierte a minúscula
char toupper(char car)	Convierte a mayúscula



9/14

Introducción a las matrices
Vectores numéricos
Cadenas de caracteres
Matrices

Definición de una cadena de caracteres Trabajar con cadenas de caracteres

Cadenas de caracteres

Trabajar con cadenas de caracteres (II)

```
int verificar() {
  char secreto="abracadabra";
  char password[50];

  gets(password);
  if (strcmp(secreto, password) == 0) return 1;
  else return 0;
}
```

Cadenas de caracteres

Trabajar con cadenas de caracteres (III)

```
Ejemplo 2
int main() {
  char cadena[50], copia[50];

  gets(cadena);
  strcpy(copia, cadena);
  puts(copia);
  puts(cadena);

return 0;
}
```

La sentencia copia = cadena no sería válida ... ¿por qué?

◆ロ → ◆御 → ◆ 重 → ◆ 重 → りへの

11/14

Introducción a las matrices
Vectores numéricos
Cadenas de caracteres
Matrices

Concepto de matriz Definición de matrices

Matrices

Concepto de matriz

- La capacidad expresiva de los vectores es limitada
 - ¿Cómo representar las tablas de multiplicar?
 - ¿Cómo almacenar varias mediciones diarias durante un mes?
 - Respuesta: matrices (o arrays multidimensionales)
- Una matriz es un vector de vectores
 - Se puede visualizar como una simple tabla
 - Las matrices tienen dos índices (fila y columna)

Ejemplo matriz 3x3

m_{00}	$m_{00} m_{01} r$		
m_{10}	m_{11}	m_{12}	
m_{20}	m_{21}	m_{22}	

Matriz de 3x3 en memoria

m_{00}	m_{01}	m_{02}	m_{10}		m_{22}
----------	----------	----------	----------	--	----------

Matrices Definición de matrices

- Las matrices se definen análogamente a los vectores
- Un elemento de una matriz es una variable
 - Mismo uso que vectores
- Dos índices
 - Fila (normalmente i)
 - Columna (normalmente j)
 - Mínimo: 0
 - Máximo: tamaño matriz

Formato definición

```
tipo matriz[n_filas][n_columnas];
```

Ejemplos

```
int m[3][4];

m[2][3] = 5;
m[2][3] = m[3][3];
temp = m[2][3] + m[1][1];
temp = m[2][3] + m[1][1];
temp = m[2][3] * 6;
temp = m[0][0]++;
m[5][0]++; // ERROR
```

13 / 14

Introducción a las matrices Vectores numéricos Cadenas de caracteres Matrices

Concepto de matriz Definición de matrices

Matrices Ejemplo

```
#define N_FILAS 3
#define N_COLS 3

int main() {
   int m[N_FILAS][N_COLS];
   int i, j;

for (i=0; i<N_FILAS; i++)
     for (j=0; i<N_COLS; j++)
        scanf("%d", &m[i][j]);

for (i=0; i<N_FILAS; i++)
   for (j=0; i<N_COLS; j++)
     printf("%d", m[i][j]);

return 0;</pre>
```