

# Dades Estructurades (II)

## Què veurem en aquest laboratori?

Tipus de Dades Estructurades Homogènies:

- Taules de varies dimensions: Matrius.



# Matrius

## Recordem alguns aspectes sobre taules N-dimensionals

Són un conjunt de valors homogenis d'un tipus bàsic, agrupats en varies dimensions.

Declaració (en C):

**tipus nom[tam<sub>1</sub>][tam<sub>2</sub>][tam<sub>3</sub>]...[tam<sub>N</sub>];**

tam<sub>i</sub> és el n° de posicions que té cada dimensió.

Atenció: Amb aquesta declaració no s'inicialitza la matriu, només es reserva espai de memòria. Ex:

int matriu[10][5];  $\Rightarrow$  (10 x 5 x 4 bytes/enter)= 200 bytes

int matriu[10,5]; NO és correcte!

(Els elements de la matriu s'emmagatzemen per files.)

Inicialització: es pot fer de dues maneres,

- A la definició  $\Rightarrow$  int tabla[2][3]={ {10,12,14}, {16,18,20}};
- Per programa  $\Rightarrow$  for(i=0;i<tam1;i++)  
                                   for (j=0;j<tam2;j++)  
                                   tabla[i][j]=valor\_inicial;



# Matrius

## Recordem alguns aspectes sobre vectors N-dimensionals

Accés: a cada element de la matriu,

`a=matriu[i][j];`     $i \in [0..9]$  ,  $j \in [0..4]$

`matriu[i][j]=a;`     $i \in [0..9]$  ,  $j \in [0..4]$

Atenció:

- El compilador de C no realitza cap control d'errors per evitar l'accés a posicions de fora de la matriu, ho ha de controlar el propi programador. Exemples de programació incorrecta:

```
for(i=0;i<tam1;i++)
```

```
    for (j=0;j<tam2;j++)
```

```
        tabla[i+2][j+2]=valor_inicial; /* Ens sortim de la matriu */
```

```
for(i=0;i<tam1+4;i++)
```

```
    for (j=0;j<tam2+4;j++)
```

```
        tabla[i][j]=valor_inicial; /* Ens sortim de la matriu */
```

