Fundamentos de la programación

6 Tipos de datos estructurados II

Facultad de Informática
Universidad Complutense

Autora: Ana Gil Luezas (Adaptadas del original de Luis Hernández Yáñez)
Pequeños cambios realizados por: Yolanda García

Índice

1. Arrays bidimensionales

Arrays bidimensionales

Arrays de dos dimensiones

```
typedef tipo_base nombre[tamaño1][tamaño2];
```

Filas

Columnas

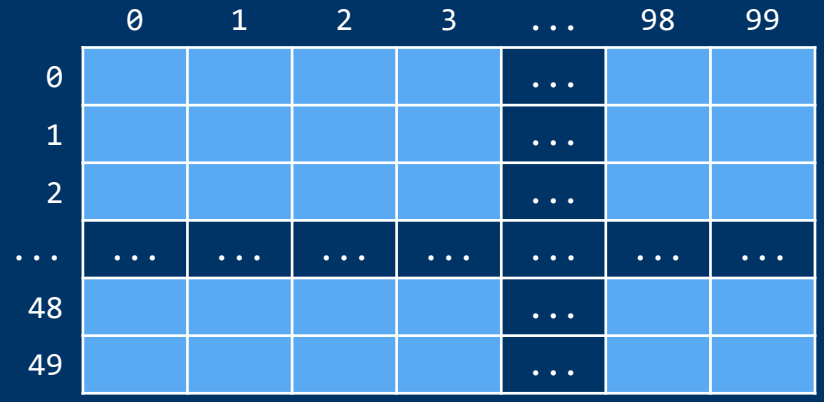
Arrays bidimensionales

Arrays de dos dimensiones

Ejemplo: Definimos un tipo matriz

```
typedef int tMatriz[50][100];
tMatriz matriz;
```

La variable `matriz` es una tabla bidimensional de 50 filas por 100 columnas:



Arrays bidimensionales

Arrays de dos dimensiones,

Acceso a los datos: acceso directo

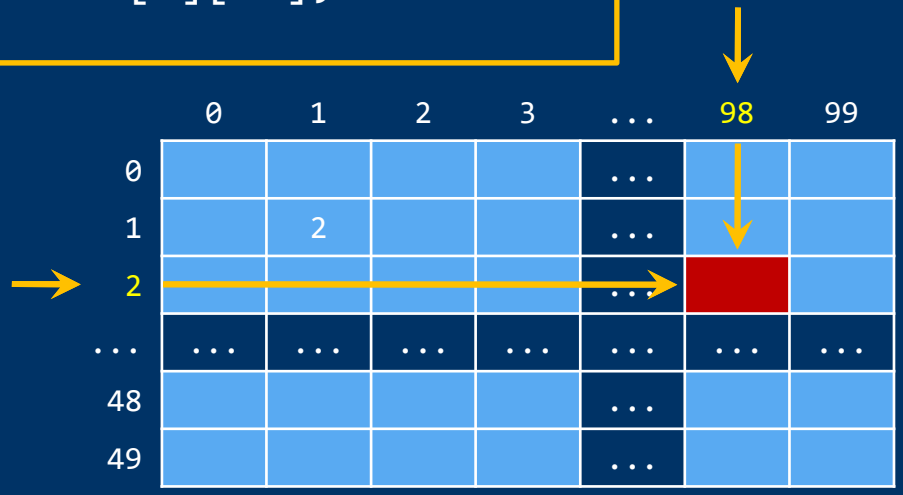
Ahora cada elemento del array se localiza con dos índices, uno por cada dimensión :

array[fila][columna]

Arrays bidimensionales

Arrays de dos dimensiones, acceso directo

```
typedef int tMatriz[50][100];
tMatriz matriz;

matriz[2][98] = 0;
cout << matriz[2][98];
```

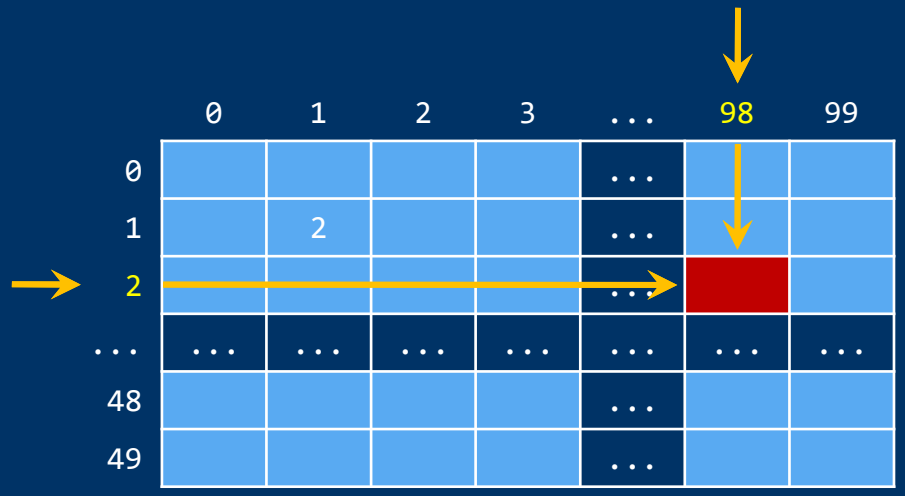


Arrays bidimensionales

Arrays de dos dimensiones, acceso directo

```
typedef int tMatriz[50][100];
tMatriz matriz;
matriz[matriz[1][1]][98] += 1;
```

2



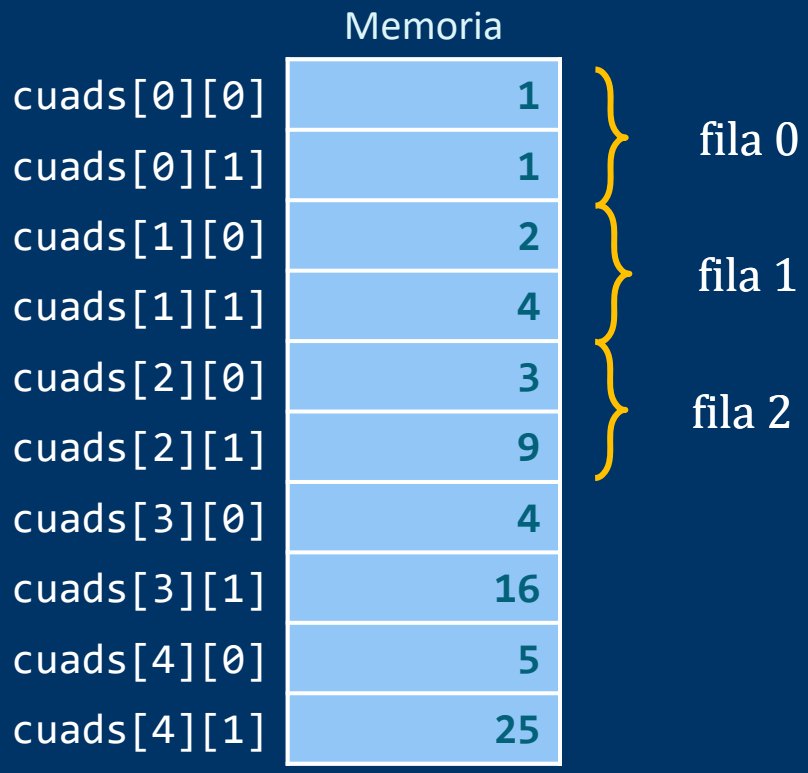
Arrays bidimensionales

La memoria es de una dimensión: secuencia de celdas

Los elementos de un array bidimensional se colocan en la memoria por filas: para cada valor del primer índice todos los valores del segundo.

```
int cuads[5][2];
```

5 filas de 2 columnas de enteros



Arrays bidimensionales

Inicialización

Para cada valor del primer índice todos los valores del segundo:

Para cada *fila* (de 0 a FILAS – 1):

Para cada *columna* (de 0 a COLUMNAS – 1):

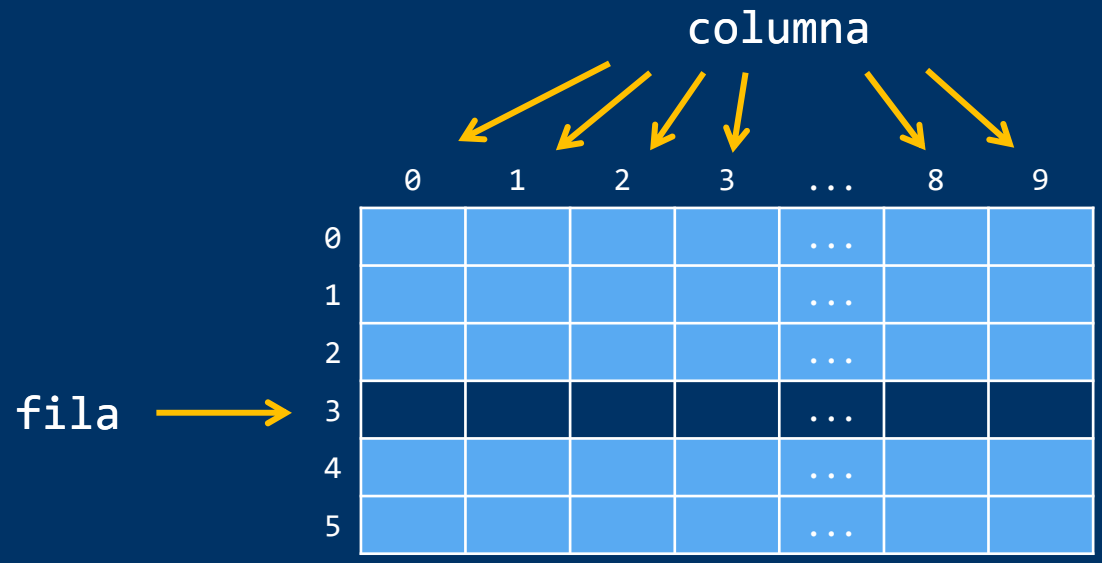
```
const int FILAS = 6;
const int COLUMNAS = 10;
typedef int tMatriz[FILAS][COLUMNAS];
tMatriz matriz;


for (int fila = 0; fila < FILAS; ++fila)
  for (int columna = 0; columna < COLUMNAS; ++columna)
     matriz[fila][columna] = 0;
```

Arrays bidimensionales

Función de inicialización de la matriz

`tElem` puede ser de cualquier tipo

```
void iniciar(tMatriz matriz, tElem vi){
  for (int fila = 0; fila < FILAS; ++fila)
     for (int columna = 0; columna < COLUMNAS; ++columna)
        matriz[fila][columna] = vi;
}
```



Arrays bidimensionales

Recorrido por filas

```
const int FILAS = …;
const int COLUMNAS = …;
typedef tElem tMatriz[FILAS][COLUMNAS];
tMatriz matriz;
```

```
… recorrer(tMatriz matriz, …) {
  for (int fila = 0; fila < FILAS; ++fila)
     for (int columna = 0; columna < COLUMNAS; ++columna)
        // Procesar matriz[fila][columna];
}
```

Arrays bidimensionales

Ejemplo: *Suma de matrices*

Arrays bidimensionales

Ejemplo: Suma de matrices

```
const int DimMat = 30;
typedef double tMat[DimMat][DimMat];
```

```
void suma(tMat const mat1, tMat const mat2,
          tMat matR) {
  for (int fila = 0; fila < DimMat; ++fila)
    for (int columna = 0; columna < DimMat; ++columna)
      matR[fila][columna] = mat1[fila][columna] +
                            mat2[fila][columna];
}
```

Arrays bidimensionales

Ejemplo: Mostrar matriz por filas

Arrays bidimensionales

Ejemplo: Mostrar matriz por filas

```
typedef int tMatriz[FILAS][COLUMNAS];
```

```
void mostrar(tMatriz const mat) {
  for (int fila = 0; fila < FILAS; ++fila) {
    for (int columna = 0; columna < COLUMNAS; ++columna)
      cout << mat[fila][columna] << ' ';
    cout << endl;
  }
}
```

Arrays bidimensionales

Ejemplo: Leer matriz por filas

Arrays bidimensionales

Ejemplo: Leer matriz por filas

```
const int DimMat = 30;
typedef double tMat[DimMat][DimMat];
```

```
void leer(tMat mat) {
  for (int fila = 0; fila < DimMat; ++fila)
    for (int columna = 0; columna < DimMat; ++columna)
      cin >> mat[fila][columna];
}
```

Arrays bidimensionales

Ejemplo: Producto de los elementos de la diagonal princi

Arrays bidimensionales

Ejemplo: Producto de los elementos de la diagonal princi

```
const int DimMat = 30;
typedef double tMat[DimMat][DimMat];
```

```
double productoDiagonal(tMat const mat) {
  double prod = 1.0;
  for (int i = 0; i < DimMat; ++i)
     prod *= mat[i][i];
  return prod;
}
// diagonal principal: fila == col
```

Arrays bidimensionales

Ejemplo: Intercambiar dos filas

Arrays bidimensionales

Ejemplo: Intercambiar dos filas

```
typedef double tMatriz[FILAS][COLUMNAS];
```

```
bool interFilas(tMatriz mat, int f1, int f2){
 if (f1 < 0 || f1 >= FILAS || f2 < 0 || f2 >= FILAS)
   return false;
 else {
   if (f1 != f2)
    for (int c = 0; c < COLUMNAS; ++c)
       intercambiar(mat[f1][c], mat[f2][c]);
   return true;
 }
}
```