



INSTITUTO TECNOLÓGICO BELTRÁN

Centro de Tecnología e Innovación



Algoritmos y estructuras de datos – Lic. Yaps David P.

VECTORES MULTIDIMENSIONALES

Podemos crear vectores con más de una dimensión, pasando de la idea vector de una sola fila a la idea de matriz de $m \times n$ elementos:

```
int mEnteros[10][5];  
float mDecimales[3][2][4];
```

En este ejemplo, *mEnteros* es una matriz de 10×5 valores enteros y *mDecimales* es un vector de tres dimensiones con $3 \times 2 \times 4$ números en coma flotante.

Para acceder a un elemento cualquiera de los vectores se deben utilizar tantos índices como dimensiones tiene el vector, por ejemplo:

```
mEnteros[4][2]  
mDecimales[1][0][3]
```

También es posible inicializar explícitamente un vector multidimensional:

```
int mBinarios[3][3] = { {1, 0, 0},  
                        {0, 1, 0},  
                        {0, 0, 1} };
```

VECTORES MULTIDIMENSIONALES

Vamos a almacenar en una matriz el número de alumnos con que cuenta una academia ordenados en función del nivel y del idioma que se estudia. Tendremos 3 filas que representarán Nivel básico, medio o de perfeccionamiento; y 4 columnas que representarán los idiomas (0 = Inglés, 1 = Francés, 2 = Alemán y 3 = Ruso). La declaración de la matriz sería:

```
Int mAlumnosXNivelIdiomas [3][4];
```

Podríamos asignar contenidos de la siguiente manera:

```
mAlumnosXNivelIdioma[0][0] = 7;  
mAlumnosXNivelIdioma[0][2] = 8;  
mAlumnosXNivelIdioma[1][0] = 6;  
mAlumnosXNivelIdioma[1][2] = 7;  
mAlumnosXNivelIdioma[2][0] = 3;  
mAlumnosXNivelIdioma[2][2] = 4;
```

```
mAlumnosXNivelIdioma[0][1] = 14;  
mAlumnosXNivelIdioma[0][3] = 3;  
mAlumnosXNivelIdioma[1][1] = 19;  
mAlumnosXNivelIdioma[1][3] = 2;  
mAlumnosXNivelIdioma[2][1] = 13;  
mAlumnosXNivelIdioma[2][3] = 1;
```

VECTORES MULTIDIMENSIONALES

La representación gráfica que podríamos asociar a esta asignación de datos sería esta matriz:

$$\begin{bmatrix} 7 & 14 & 8 & 3 \\ 6 & 19 & 7 & 2 \\ 3 & 13 & 4 & 1 \end{bmatrix}$$

O bien, podemos representar los valores como si fuera una tabla:

	Inglés	Francés	Alemán	Ruso
Nivel Básico	7	14	8	3
Nivel Medio	6	19	7	2
Perfeccionamiento	3	13	4	1

VECTORES MULTIDIMENSIONALES

La organización de la información en matrices nos generará importantes ventajas a la hora del tratamiento de datos en nuestros programas.

Para terminar en cuanto a multidimensionalidad, veamos casos de declaraciones con más de dos dimensiones. Para ello utilizaremos el ejemplo del conteo de coches (suponemos que queremos registrar el número de coches que han pasado por hora). La forma de declarar esos vectores sería la siguiente:

Duración del conteo	Tipo de array	Declaración con C (Nc es Número de coches)
Un día	Array de un localizador (hora)	<code>int Nc[24];</code>
Varios días	Array de dos localizadores (hora y día)	<code>int Nc[24][31];</code>
Varios meses	Array de tres localizadores (hora, día y mes)	<code>int Nc[24][31][12];</code>
Varios años	Array de cuatro localizadores (hora, día, mes y año)	<code>int Nc[24][31][12][2999];</code>

EJEMPLO DE INICIALIZACIÓN DE VECTORES MULTIDIMENSIONALES

```
int main() {  
    int mEnteros[10][5];  
    float mDecimales[3][2][4];  
    int i, j, k;  
  
    for (i=0; i<10; i++) {  
        for (j=0; j<5; j++) {  
            printf("Iniciando mEnteros[%d][%d]: \n", i, j);  
            mEnteros[i][j] = 0;  
        }  
    }  
  
    for (i=0; i<3; i++) {  
        for (j=0; j<2; j++) {  
            for (k=0; k<4; k++) {  
                printf("Iniciando mDecimales[%d][%d][%d]: \n", i, j, k);  
                mDecimales[i][j][k] = 0.0;  
            }  
        }  
    }  
}
```

La inicialización de los vectores multidimensionales necesita tantos bucles anidados como dimensiones tengan éstos.



INSTITUTO TECNOLÓGICO BELTRÁN

Centro de Tecnología e Innovación