

Linguagem de Programação

Arrays multidimensionais (matrizes)

ECT2303

helton.maia@ufrn.br

Arrays - Introdução

- Os dados dispostos em matrizes multidimensionais são armazenados em forma de tabela;
- Forma geral para se declarar matrizes N-dimensionais:

```
tipo array_nome[tamanho1][tamanho2]...[tamanhoN];
```

tipo: Tipo de dados para armazenamento na array

array_nome: Nome da array

tamanho 1,2,...,N: Tamanhos para o dimensionamento

Arrays - Exemplos

- Com duas dimensões:

```
int dois_d[10][20];
```

- Com três dimensões:

```
int tres_d[10][20][30];
```

Importante: O número total de elementos que podem ser armazenados em uma array multidimensional pode ser calculado multiplicando o tamanho de todas as dimensões.

Por exemplo: A matriz `int dois_d[10][20]` pode armazenar o total de $(10 * 20) = 200$ elementos do tipo `int`.

Arrays - Memória utilizada

`sizeof(<tipo_elementos> * <numero_linhas> * <numero_colunas>;`

- **Cosidere uma matriz de duas dimensões:**

```
int dois_d[2][3];  
int memoria_utilizada = sizeof(int)*2*3;  
//o array possui 24 Bytes
```

Arrays - Exemplo

Considere uma array bidimensional(uma matriz), sua forma básica de declaração é:

```
//tipo array_nome[x][y];
```

```
int m[10][20];
```

Obs: Normalmente referenciamos as matrizes utilizando a seguinte sintaxe: **x[i][j]**, onde “i” é o número de linhas e “j” de colunas.

Arrays - Visão de uma tabela

A matriz pode ser vista como uma tabela com "x" linhas e "y" colunas, onde o número da linha varia de 0 a (x-1) e o número da coluna varia de 0 a (y-1).

	Coluna 0	Coluna 1	Coluna 2
Linha 0	x[0][0]	x[0][1]	x[0][2]
Linha 1	x[1][0]	x[1][1]	x[1][2]
Linha 2	x[2][0]	x[2][1]	x[2][2]

Arrays são apenas uma abstração!

Arrays - Estrutura em memória

x[0][0]	x[0][1]	x[0][2]	x[0][3]
x[1][0]	x[1][1]	x[1][2]	x[1][3]
x[2][0]	x[2][1]	x[2][2]	x[2][3]

Posição em memória de um elemento alocado:

$\text{endereço} = \text{end_base_vetor} + \text{id_linha} * \text{num_col} + \text{id_col}$

Exemplo: x[1][2] está no endereço

$0010 + 0001 * 0100 + 0010 = 1000$ (binário)

$2 + 1 * 4 + 2 = 8$ (decimal)

0001	3643648
0010	x[0][0]
0011	x[0][1]
0100	x[0][2]
0101	x[0][3]
0110	x[1][0]
0111	x[1][1]
1000	x[1][2]
1001	x[1][3]
1010	x[2][0]
1011	x[2][1]
1100	x[2][2]
1101	x[2][3]
1110	4643686

Arrays - Inicialização

Existem **duas maneiras** em que um array(matriz) bidimensional pode ser inicializada:

- **Primeiro método:**

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

- **Segundo método:**

```
int x[2][3] = {{0, 1, 2}, {3, 4, 5}};
```

Os elementos são preenchidos da esquerda para direita.

Arrays - Acessando os elementos

Os elementos das arrays(matrizes) bidimensionais são acessados utilizando os índices da linha e da coluna.

Por exemplo:

```
int x[2][1];
```

Representa o elemento alocado na **terceira linha e segunda coluna** da matriz.

Arrays - Exemplo prático

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      //criando arrays com 3 linhas e 2 colunas
6      int x[3][2] = {{0,1},{2,3},{4,5}};
7
8      //imprimindo elementos da matriz
9      for(int i=0; i<3; i++){
10         for(int j=0; j<2; j++){
11             cout << "Elemento em x[" << i
12                 << "][" << j << "]:" << x[i][j] << endl;
13         }
14     }
15
16     return 0;
17 }
```

Output:

```
Elemento em x[0][0]: 0
Elemento em x[0][1]: 1
Elemento em x[1][0]: 2
Elemento em x[1][1]: 3
Elemento em x[2][0]: 4
Elemento em x[2][1]: 5
```

Arrays - Exercício

Escreva um programa no qual o usuário possa armazenar valores em uma *array* com duas dimensões. Depois, pesquise e imprima o maior e o menor valor armazenado.

?