LPG0002 – Linguagem de Programação

Matrizes

Profª Luciana Rita Guedes
Departamento de Ciência da Computação
UDESC / Joinville

Material elaborado por: Prof. Rui Jorge Tramontin Junior

Introdução

- Um vetor pode ter múltiplas dimensões, conforme a necessidade da aplicação;

Introdução

- Um vetor pode ter múltiplas dimensões, conforme a necessidade da aplicação;

- Conceitualmente, chamamos um vetor com 2 ou mais dimensões de **matriz**;

Introdução

- Um vetor pode ter múltiplas dimensões, conforme a necessidade da aplicação;

- Conceitualmente, chamamos um vetor com 2 ou mais dimensões de **matriz**;

- Em C, todavia, uma matriz é implementada como um **vetor de vetores**;

Declaração

- Uma matriz de 2 dimensões (4 linhas e 3 colunas):

```
int mat[4][3];
```

Declaração

- Uma matriz de 2 dimensões (4 linhas e 3 colunas):

```
int mat[4][3];
```

- Podemos dizer também que `mat` é um vetor de capacidade 4;
  - Em cada índice (linha), há um vetor de capacidade 3.

Modelo

- Conceitualmente, podemos visualizar **`mat`** como:

$$\begin{bmatrix} 13 & 8 & -6 \\ 8 & 7 & 2 \\ -1 & 4 & 0 \\ 16 & 12 & 10 \end{bmatrix}$$

Modelo

- Entretanto, é importante saber que a ideia de matriz é uma abstração da linguagem;

Modelo

- Entretanto, é importante saber que a ideia de matriz é uma abstração da linguagem;

- Na memória, uma matriz é sempre alocada em 1 dimensão;

Modelo

- Entretanto, é importante saber que a ideia de matriz é uma abstração da linguagem;
- Na memória, uma matriz é sempre alocada em 1 dimensão;
- Ou seja, as linha da matriz estão alocadas uma após a outra.

Modelo

- Portanto, em termo de alocação de memória, as duas declarações são equivalentes:

```
int mat[4][3];

int vet[12];
```

Manipulação de Matrizes

- O acesso é feito tal como em um vetor unidimensional;

Manipulação de Matrizes

- O acesso é feito tal como em um vetor unidimensional;

- Porém, é necessária uma variável para acessar os índices de cada dimensão;
  - `i` para linhas;
  - `j` para colunas.

Exemplo: entrada

```
int m[4][4], i, j;

for(i=0; i<4; i++){
   for(j=0; j<4; j++){
      printf("M[%d,%d]=", i, j);
      scanf("%d", &m[i][j]);
   }
}
```

Exemplo: saída

```
for(i=0; i<4; i++){
   for(j=0; j<4; j++){
      printf("%4d ", m[i][j]);
   }
   printf("\n");
}
```

Exemplos Práticos

- Diagonal Principal;

$$\begin{bmatrix} 13 & 8 & -6 & 3 \\ 8 & 7 & 2 & 9 \\ -1 & 4 & 0 & 13 \\ 16 & 12 & 10 & 17 \end{bmatrix}$$

Exemplos Práticos

- Diagonal Secundária;

$$\begin{bmatrix} 13 & 8 & -6 & 3 \\ 8 & 7 & 2 & 9 \\ -1 & 4 & 0 & 13 \\ 16 & 12 & 10 & 17 \end{bmatrix}$$

Exercícios

1. Dada uma matriz 4x5, calcule a média dos valores pares da matriz;

2. Dada uma matriz 5x6, gere a matriz transposta, ou seja, 6x5;

3. Implemente a soma entre duas matrizes.