



Pontifícia Universidade Católica de Minas Gerais

ICEI – Instituto de Ciências Exatas e Informática

DCC – Departamento de Ciência da Computação

Campus Belo Horizonte – Unidade Praça da Liberdade

Bacharelado em Ciência da Computação

MAIOR UNIVERSIDADE CATÓLICA DO MUNDO - Fonte: Vaticano

MELHOR UNIVERSIDADE PRIVADA DO BRASIL - Guia do Estudante, por 6x

ENTRE AS MELHORES UNIVERSIDADES DO MUNDO - Times (Ranking Times High Education)

COMPUTAÇÃO PUC MINAS: SEMPRE 2º..4º LUGAR DO PAÍS (RH) – Folha de São Paulo, RUF

CIÊNCIA DA COMPUTAÇÃO PUC MINAS: SEMPRE 4 OU 5 ESTRELAS - Guia do Estudante

Algoritmos e Estruturas de Dados I

Professor: Lúcio Mauro Pereira

Lista de Exercícios nº 21

8 a 10 de maio de 2024

Introdução aos Arranjos Bidimensionais (Matrizes)

Estudar:

Obra: Fundamentos da Programação de Computadores. Autora: Ana Ascêncio

Disponível na biblioteca da PUC Minas de forma física e *e-book*.

Capítulo 7: Matriz

Obra: C: como programar. 8ed. Autor: Deitel.

Disponível na biblioteca da PUC Minas de forma física e *e-book*.

Capítulo 6: Arrays

Introdução

Um arranjo pode representar uma coleção de dados organizada em diferentes dimensões. Um vetor pode ser entendido como uma matriz de uma única dimensão. No exemplo abaixo, é declarado uma matriz, de uma única dimensão, de tamanho igual a três:

```
float A[3];
```


Considere, por exemplo, uma matriz de dimensão (3x2), isto é, três linhas e duas colunas.

Para acrescentar essa segunda dimensão, tipicamente é utilizada uma vírgula, como em C#:

```
float A[ 3, 2];
```

ou um segundo par de colchetes, como em C:

```
float A[3][2];
```

Na instrução abaixo, a matriz é declarada e inicializada:

```
float A[3][2]= { {10,11}, {12,13}, {14,15} };
```

10	11
12	13
14	15

Para o exemplo abaixo e em toda esta lista, considere o número de linhas e o número de colunas declarados nas seguintes constantes globais: NUM_LIN e NUM_COL

Por exemplo, para os cenários apresentados acima:

```
const int NUM_LIN = 3;
```

```
const int NUM_COL = 2;
```

No exemplo abaixo, a função recebe uma matriz bidimensional de reais e a escreve na tela:

```
void escreveMatriz(float Matriz[][NUM_COL])
{
    for(int i=0; i< NUM_LIN; i++){
        for(int j=0; j< NUM_COL; j++){
            printf("\nM[%i][%i]= %f", i, j, Matriz[i][j]);
        }
        printf("\n");
    }
}
```

Questões:

Considere a função principal abaixo:

```
const int NUM_LIN = 3;
const int NUM_COL = 2;

int main() {

    float M1[NUM_LIN][ NUM_COL];
    leMatriz(M1);

    float M2[NUM_LIN][ NUM_COL];
    leMatriz(M2);

    if( iguais(M1, M2) ) printf("\nMatrizes iguais!");
    else printf("\nMatrizes diferentes!");

    return 0;
}
```

1. Implemente a função para realizar a leitura dos valores para uma matriz bidimensional.
Argumento: o endereço para uma matriz de reais
Valor gerado: nenhum
2. Implemente uma função que receba duas matrizes de reais. A função deverá copiar a primeira matriz na segunda.
Argumentos: os endereços para duas matrizes de reais
Valor gerado: nenhum
3. Implemente a função que verifica se duas matrizes bidimensionais são iguais ou não.
Argumentos: os endereços das duas matrizes a serem comparadas
Valor gerado: *true*, caso sejam as matrizes iguais, ou *false*, caso contrário
4. Implemente uma função que receba duas matrizes de reais. A função deverá calcular a matriz transposta da primeira, armazenando-a na segunda matriz.
* Planeje, com cuidado, as dimensões de ambas as matrizes parametrizadas