

Matrizes

Algoritmos e Estrutura de Dados I

Instituto de Engenharia – UFMT

Roteiro

- 1 Objetivos
- 2 Introdução
- 3 Manipulando matrizes
- 4 Exemplos

Objetivos

Esta aula tem como objetivos:

- 1 Apresentar os conceitos básicos sobre Matrizes;
- 2 Explicitar os exemplos de funções que manipulam Matrizes;
- 3 Exemplificar a execução desses algoritmos.

Introdução

- Imaginemos que queremos ler as notas de 4 provas para cada aluno e calcular a média do aluno e a média da classe.
- O tamanho máximo da turma é de 50 alunos.
- Uma solução seria criar 4 vetores cada um com 50 posições e então ler as respectivas informações.

```
float nota1[50], nota2[50], nota3[50], nota4[50];
```

Introdução

	Aluno		Nota1		Nota2		Nota3		Nota4		Média
0	Pedro	0	5.6	0	6.0	0	7.3	0	5.6	0	6.1
1	Ana	1	10	1	4.0	1	5.0	1	7.3	1	6.6
2	Luiz	2	4.5	2	2.0	2	5.5	2	1.0	2	3.3
...
48	Matheus	48	7.2	48	6.6	48	8.1	48	8.8	48	7.7
49	Andre	49	6.0	49	9.0	49	7.3	49	4.5	49	6.6

Introdução

- Agora suponha que estamos trabalhando com no máximo 100 provas e 50 alunos.
- Seria cansativo criar 100 vetores e atribuir 100 nomes diferentes.

	Aluno		Nota1		Nota2		Nota3		Nota99		Nota100		Média
0	Pedro	0	5.6	0	6.0	0	7.3	0	7.3	0	5.6	0	6.1
1	Ana	1	10	1	4.0	1	5.0	1	5.0	1	7.3	1	6.6
2	Luiz	2	4.5	2	2.0	2	5.5	2	5.5	2	1.0	2	3.3
...
48	Matheus	48	7.2	48	6.6	48	8.1	48	8.1	48	8.8	48	7.7
49	Andre	49	6.0	49	9.0	49	7.3	49	7.3	49	4.5	49	6.6

- Para resolvermos esse problema, utilizamos matrizes.

Introdução

- Uma matriz é um vetor (conjunto de variáveis de mesmo tipo) que possui duas ou mais dimensões.
- Matrizes podem ser considerados como “vetores de vetores”.
- Por exemplo, uma matriz bidimensional pode ser vista como uma tabela de m linhas e n colunas.

A diagram illustrating a 3x4 matrix. The matrix is represented as a table with 3 rows and 4 columns. The rows are indexed 0, 1, and 2 on the left, with a bracket labeled m indicating the total number of rows. The columns are indexed 0, 1, 2, and 3 on top, with a bracket labeled n indicating the total number of columns.

	0	1	2	3
0				
1				
2				

Declarando uma matriz

- Utiliza-se a forma geral para declaração de matrizes:

```
<tipo> nome_da_matriz [<linhas>][<colunas>;
```

- Uma matriz possui linhas x colunas do tipo <tipo>.
- As linhas são numeradas de 0 a linhas -1.
- As colunas são numeradas de 0 a colunas -1.
- Exemplo:

```
// Matriz com 100 linhas e 50 colunas  
float notas[100][50];
```

Acessando elementos da matriz

- Para acessar um elemento da matriz deve-se especificar a linha e coluna:

```
nome_da_matriz [<linha>][<coluna>];
```

- Exemplo: imprimir o elemento da linha 1 e coluna 10.

```
printf("%d", matriz[1][10]);
```

- O compilador não verifica se foram utilizados valores válidos para a linha e para a coluna.

Atribuindo valores para a matriz

- Para atribuir um valor para um elemento da matriz deve-se especificar a linha e coluna:

```
nome_da_matriz [<linha>][<coluna>] = 0;
```

- Para atribuir zero a todos os elementos da matriz é necessário um laço de repetição para cada dimensão da matriz.
- Por exemplo, para uma matriz bidimensional, são necessários dois laços de repetição:

```
int matriz[100][50], i,j;  
for(i=0;i<100;i++) {  
    for(j=0;j<50;j++) {  
        matriz[i][j] = 0;  
    }  
}
```

Exemplo

Criar uma matriz com 3 linhas e 4 colunas, inicializar a matriz com valores fornecidos pelo usuário e imprimir o valores.

Exemplo

```
#include<stdio.h>
int main() {
    int matriz[3][4], i, j;
    // Inicializa a matriz com valores fornecidos pelo usuario
    for (i=0;i<3;i++) {
        for (j=0;j<4;j++) {
            scanf("%d",&matriz[i][j]);
        }
    }
    // Imprime a matriz
    for (i=0;i<3;i++) {
        for (j=0;j<4;j++) {
            printf("%d",matriz[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Dúvidas

Fim