

Conjuntos Bidimensionais Homogêneos: Matrizes

Matriz é um conjunto de dados com o mesmo tipo, referenciado por um nome e que necessita de mais de um índice para armazenar seus elementos.

Exemplo: para armazenar 3 notas de 5 alunos, pode-se criar uma matriz de 2 dimensões, sendo uma dimensão com 5 linhas representando os alunos e a outra com 3 colunas representando as notas.

Em C, a declaração de uma matriz é como segue:

```
float nota[5][3];
```

		índice da coluna (nota)		
		0	1	2
variável nota	0	7.0	5.0	3.0
	1	8.0	4.5	6.7
	2	4.5	5.0	10.0
	3	9.0	8.0	6.0
	4	6.0	5.0	5.0
		↓		
		índice da linha (aluno)		

Acessando os elementos de uma matriz

Para particularizar uma célula de uma estrutura “matriz”, deve-se utilizar o identificador e um valor de índice para cada dimensão.

Sintaxe: identificador[índice da linha][índice da coluna]

Para acessar, na variável nota, a nota 2 do aluno 3, deve-se utilizar `nota[3][2] = 6.0`.

Observação: o índice em linguagem C começa sempre em zero.

Armazenando e Acessando dados em uma matriz

Para o armazenamento ou acesso em linha, deve-se fixar um valor de índice de linha e variar sequencialmente os índices de coluna. Após o término do mapeamento de coluna para a linha fixada, se fixa à próxima linha e percorre-se o mapeamento da coluna para esta nova linha. Para o armazenamento em coluna o procedimento é o mesmo, porém, fixando-se a coluna e variando a linha.

Exemplo: armazenando dados na matriz “nota”

```
#include<stdio.h>
void main(){
    float nota[5][3];
    int i,j;
    for (i=0; i<5; i++) {
        for (j=0; j<3; j++) {
            printf("Entre com a nota %d do aluno %d", j, i);
            scanf("%f", &nota[i][j]);}}}
```

Inicializando uma matriz

A inicialização de uma matriz é feita da mesma forma que a de um vetor, onde os elementos de cada linha são colocados entre chaves.

Exemplo: `float nota[2][3] = { {8.0, 3.0, 5.0}, {8.5, 3.5, 5.5} };`

Declarando e armazenando dados em uma matriz de strings

```
#include <stdio.h>
void main(){
    char aluno[5][30];
    int i;
    for (i=0; i<5; i++){
        printf("Entre com o nome do aluno %d", i);
        scanf("%s", aluno[i]);
    }
```

OBS: É importante notar que para a leitura de uma *string*, usando *scanf*, não é necessário usar o símbolo & antes do nome da variável.

Exercícios de fixação (utilizar funções de entrada, processamento e saída):

- 1) Faça um programa que leia 20 números inteiros e armazená-los por linha em uma matriz de 2 linhas e 5 colunas.
- 2) Faça um programa onde os seguintes números devem ser lidos para uma matriz de 4 linhas e 3 colunas: 21, -3, 55, 788, 5, 93, 4, 45, 595, 666, 5 e 88. A saída do programa deve ter o seguinte formato.

21	5	595
-3	93	666
55	4	5
788	45	88

- 3) Faça um programa que leia os dados de uma matriz de 4 linhas e 4 colunas, composta de elementos reais, e calcule a soma dos elementos da diagonal principal da matriz.
- 4) Faça um programa que leia os valores de uma matriz 10x10 de elementos reais, e crie a matriz transposta da matriz fornecida.
- 5) Faça um programa que leia uma matriz 20x20 de elementos reais e gere um vetor de 20 elementos reais, onde cada elemento do vetor seja a soma de cada linha da matriz.
- 6) Faça um programa que receba uma matriz 10x10 de elementos inteiros, e localize a posição (linha e coluna) do maior elemento da matriz.
- 7) Faça um programa que calcule a média das notas por aluno e a média geral de 5 alunos, sabendo que cada um fez 3 provas.