

Linguagem C: Variáveis Indexadas - Matrizes

Prof. Paulo R. S. L. Coelho

paulo@facom.ufu.br

Faculdade de Computação
Universidade Federal de Uberlândia



Organização

1

Matrizes

- Introdução
- Exemplo
- Exercícios



Organização

1 Matrizes

- Introdução
- Exemplo
- Exercícios



Introdução

- Uma **matriz** é uma variável indexada n-dimensional, formada por uma sequência de variáveis.
- Todas essas variáveis são do mesmo tipo, ocupam regiões adjacentes de memória e são acessadas pelo mesmo identificador.
- Desta forma, o que as distingue são seus índices, que referenciam sua localização dentro dessa estrutura.



Matrizes em C

- Como em vetores, os índices utilizados na linguagem C para identificar posições de uma matriz começam sempre em 0 (zero) e vão até o tamanho da matriz menos uma unidade, para cada dimensão;

- Em C, uma matriz é declarada da seguinte maneira:

```
tipo variavel[tamanho1][tamanho2][tamanhoN];
```

Ex:

```
int vetor[3][4]; //matriz 3 x 4
```

Esta matriz é acessada através dos índices de 0 a 2 na linha, e de 0 a 3 na coluna:

```
vetor[2][1] = 88;
```

- Uma matriz pode ser inicializada na sua declaração:

```
int vetor[3][4] = {{0, 1, 2, 4}, {0, 3, 7}};
```

```
int vetor[3][4] = {0};
```



Exemplo

O programa a seguir contabiliza o número de acidentes por esquina em uma cidade:

```
#include <stdio.h>
int main() {
    int i, j, n, h, v, cruza[10][10];
    for (i = 0; i <= 9; i++)
        for (j = 0; j <= 9; j++)
            cruza[i][j] = 0;
    printf("Mapa de acidentes:\n\n\tNumero de Acidentes: ");
    scanf("%d", &n); printf("\n\n");
    for (i = 1; i <= n; i++) {
        printf("\tCruzamento: ");
        scanf("%d%d", &h, &v);
        if (h >= 0 && h <= 9 && v >= 0 && v <= 9)
            cruza[h][v]++;
    }
    printf("\n\n");
    for (i = 0; i <= 9; i++) {
        for (j = 0; j <= 9; j++)
            printf("%3d", cruza[i][j]);
        printf("\n");
    }
    return 0;
}
```

Exercícios

- 1 Escrever um programa para ler uma matriz 5 x 5, imprimir essa matriz, bem como calcular e exibir sua transposta.
- 2 Escrever um programa que lê um número positivo, guardando na variável n , constrói e imprime um *triângulo de Pascal* de dimensão n , usando matrizes. Nesse triângulo:
 - elementos da primeira coluna e da diagonal são iguais a 1;
 - os outros elementos (abaixo da diagonal principal) são dados por:
$$A[i][j] = A[i-1][j-1] + A[i-1][j];$$
 - os elementos acima da diagonal principal não devem ser impressos.



Respostas I

```
1 #include <stdio.h>
int main() {
    int mat[5][5], transp[5][5], i, j, n, aux;

    printf("entre elementos da matriz 5 x 5\n");
    for (i = 0; i < 5; i++)
        for (j = 0; j < 5; j++)
            scanf("%d", &mat[i][j]);

    //calculo da transposta
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            transp[i][j] = mat[j][i];
        }
    }

    printf("\n\nMatriz original:\n");
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++)
            printf("%5d", mat[i][j]);
        printf("\n");
    }
}
```


Respostas II

```
printf("\n\nMatriz transposta:\n");
for (i = 0; i < 5; i++) {
    for (j = 0; j < 5; j++)
        printf("%5d", transp[i][j]);
    printf("\n");
}
return 0;
}
```

2

```
#include <stdio.h>
int main() {
    int mat[20][20] = {0}, i, j, n;

    printf("Entre dimensao do triangulo de Pascal: ");
    scanf("%d", &n);

    //1a. coluna e diagonal = 1
    for (i = 0; i < n; i++) {
        mat[i][i] = 1;
        mat[i][0] = 1;
    }
}
```

Respostas III

```
//demais elementos
for (i = 2; i < n; i++) {
    for (j = 1; j < i; j++) {
        mat[i][j] = mat[i-1][j-1] + mat[i-1][j];
    }
}

//impressao do resultado
for (i = 0; i < n; i++) {
    //ate diagonal apenas (j == i)
    for (j = 0; j <= i; j++) {
        printf("%5d", mat[i][j]);
    }
    printf("\n");
}

return 0;
}
```