

Rodrigo Henrich

rodrigohenrich@faccat.br

Matrizes

- Matrizes, são como os vetores, porém com duas dimensões
- Podemos associar matrizes como se fossem tabelas na memória do computador
- Cada posição então tem uma posição de linha e coluna

| | | | |
|-------|-------|-------|-------|
| L0 C0 | L0 C1 | L0 C2 | L0 C3 |
| L1 C0 | L1 C1 | L1 C2 | L1 C3 |
| L2 C0 | L2 C1 | L2 C2 | L2 C3 |

Matrizes em C

- Pensando em uma matriz temos o seguinte endereçamento

| | | | |
|--------|--------|--------|--------|
| [0][0] | [0][1] | [0][2] | [0][3] |
| [1][0] | [1][1] | [1][2] | [1][3] |
| [2][0] | [2][1] | [2][2] | [2][3] |

Matrizes em C

- Populando uma matriz

| [0][0] | [0][1] | [0][2] | [0][3] |
|--------|--------|--------|--------|
| 10 | 25 | 9 | 8 |
| [1][0] | [1][1] | [1][2] | [1][3] |
| 6 | 11 | 58 | 99 |
| [2][0] | [2][1] | [2][2] | [2][3] |
| 5 | 1 | 0 | 100 |

Matrizes em C

- Declarar uma matriz em C
- `tipoMatriz nome[numeroLinhas][numeroColunas];`
- Exemplo:

```
int matrizInteiros[3][4];
```

Matrizes em C

- inicializando uma matriz

```
int matrizInteiros[3][4] = {  
    L0      L1      L2  
    {10,25,9,8}, {6,11,58,99}, {5,1,0,100}};  
      C0  C1 C2 C3      C0  C1 C2 C3      C0  C1 C2 C3
```

Matrizes em C

- Acessando uma posição da matriz

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      setlocale(LC_ALL, "");
5      int numeros[3][4];
6      numeros[2][3] = 5;
7      printf("%d", numeros[2][3]);
8  }
```

Atribuindo valor a uma posição

Lendo o valor de uma posição

Matrizes em C

- Percorrendo uma matriz em c
- Para isso precisamos de duas estruturas de repetição
- Uma para percorrer as linhas
- E outra para percorrer as colunas
- Essas duas estruturas ficam aninhadas

Matrizes em C

```
int linha;  
int coluna;  
for(linha=0;linha<3;linha++){  
    for(coluna=0;coluna<4;coluna++){  
        <comandos>  
    }  
}
```

Matrizes em C

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "");
6      int matrizInteiros[3][4] = {{10,25,9,8},{6,11,58,99},{5,1,0,100}};
7      int linha;
8      int coluna;
9      for(linha=0;linha<3;linha++){
10         for(coluna=0;coluna<4;coluna++){
11             printf("%d ",matrizInteiros[linha][coluna]);
12         }
13         printf("\n");
14     }
15 }
```

Matrizes em C

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "");
6      int matrizInteiros[3][4] = {{10,25,9,8},{6,11,58,99},{5,1,0,100}};
7      int linha;
8      int coluna;
9      for(linha=0;linha<3;linha++){
10         for(coluna=0;coluna<4;coluna++){
11             printf("%d ",matrizInteiros[linha][coluna]);
12         }
13         printf("\n");
14     }
15 }
```

Matrizes em C

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "");
6      int matrizInteiros[3][5];
7      int linha;
8      int coluna;
9      for(linha=0;linha<3;linha++){
10         for(coluna=0;coluna<5;coluna++){
11             printf("\nNúmero posição [%d][%d] da matriz ",linha,coluna);
12             scanf("%d",&matrizInteiros[linha][coluna]);
13         }
14     }
15 }
```

Matrizes em C

Complete o exemplo anterior para mostrar a matriz lida.

Modelo matriz

| | | | | | |
|--------|----|----|----|----|----|
| linha | | | | | |
| coluna | | | | | |
| | 00 | 01 | 02 | 03 | 04 |
| | 10 | 11 | 12 | 13 | 14 |
| | 20 | 21 | 22 | 23 | 24 |
| | 30 | 31 | 32 | 33 | 34 |
| | 40 | 41 | 42 | 43 | 44 |

The diagram illustrates a 5x5 matrix model. A legend box on the left identifies 'linha' (row) and 'coluna' (column). Red arrows point from these labels to the first row and first column of the matrix. The matrix cells are labeled with two-digit numbers (e.g., 00, 01, ..., 44). The cells are color-coded: the main diagonal (00, 11, 22, 33, 44) is light blue; the anti-diagonal (04, 13, 22, 31, 40) is light orange; and the cell at (2,3) is green. A green line with arrows at both ends represents the 'Diagonal secundária' (anti-diagonal), running from the top-left to the bottom-right. A black line with arrows at both ends represents the 'Diagonal principal' (main diagonal), running from the top-right to the bottom-left.

Diagonal secundária

Diagonal principal