Algoritmos II

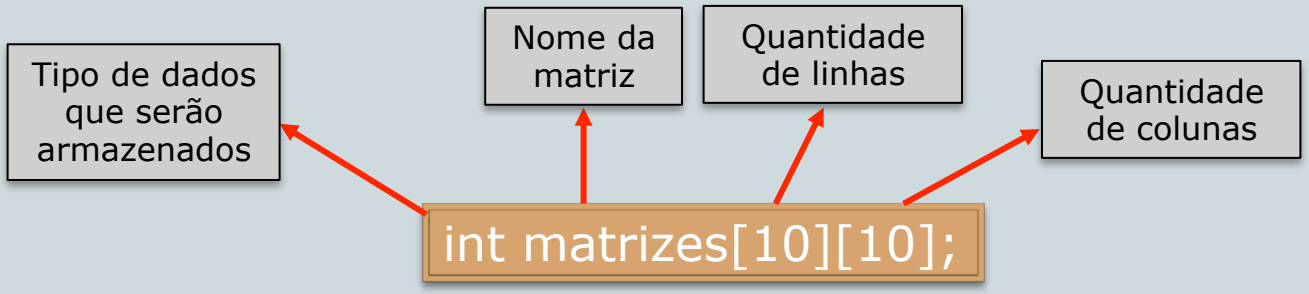
MATRIZES

Matrizes

- Uma matriz é uma variável multidimensional homogênea (possui elementos somente do tipo declarado), armazenados sequencialmente e que utilizam o mesmo nome de variável para acessar os valores armazenados.
- Como o vetor, uma matriz é um array. A diferença é que possui mais que uma dimensão.
- O vetor possui um índice de referência. Uma matriz possui dois índices para referenciar a posição de um determinado elemento: linha e coluna.

Matrizes

- Mas como se declara uma matriz?
  - Tipo dos dados que serão armazenados na matriz. Como o vetor a matriz, no C/C++, armazena somente um tipo de valor.
  - Nome da matriz, que também segue as regras de nomeação de variáveis.
  - Quantidade de linhas da matriz.
  - Quantidade de colunas da matriz.



Matrizes

- Matrizes podem ser multidimensionais, ou seja, possuírem mais que 2 dimensões (linhas e colunas).
- Com os vetores foi criado um laço de repetição para controlar o índice. Em uma matriz, é necessário utilizar dois laços de repetição: um de controle das linhas, e outro de controle das colunas.
- Exemplo:

Matrizes

- Faça um programa que preencha uma matriz 5x5 e exiba a mesma ao final.

```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <locale.h>

#define LINHAS 5
#define COLUNAS 5

int main(){
    srand(time(NULL));
    setlocale(LC_ALL,"Portuguese");

    int mat[LINHAS][COLUNAS], i, j;

    for(i=0;i<LINHAS;i++){
        for(j=0;j<COLUNAS;j++){
            mat[i][j] = rand() % (LINHAS * COLUNAS) + 1;
        }
    }

    for(i=0;i<LINHAS;i++){
        for(j=0;j<COLUNAS;j++){
            printf("%d\t",mat[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Matrizes

- Ao ser preenchida desta forma (linha representada pelo índice mais externo, coluna representada pelo índice mais interno) é feito preenchimento por linha, ou seja:

	0	1	2	3	4
0	1^0 elemento	2^0 elemento	3^0 elemento	4^0 elemento	5^0 elemento
1	6^0 elemento	7^0 elemento	8^0 elemento	9^0 elemento	10^0 elemento
2	11^0 elemento	12^0 elemento	13^0 elemento	14^0 elemento	15^0 elemento
3	16^0 elemento	17^0 elemento	18^0 elemento	19^0 elemento	20^0 elemento
4	21^0 elemento	22^0 elemento	23^0 elemento	24^0 elemento	25^0 elemento

Matrizes

- E caso invertessemos e o preenchimento fosse efetuado da forma ao lado, sendo o laço externo a coluna e o interno a linha?

```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <locale.h>

#define LINHAS 5
#define COLUNAS 5

int main() {
    srand(time(NULL));
    setlocale(LC_ALL,"Portuguese");

    int mat[LINHAS][COLUNAS], i, j;

    for (j=0; j<COLUNAS; j++) {
        for (i=0; i<LINHAS; i++) {
            mat[i][j] = rand() % (LINHAS * COLUNAS) + 1;
        }
    }

    for (i=0; i<LINHAS; i++) {
        for (j=0; j<COLUNAS; j++) {
            printf("%d\t",mat[i][j] );
        }
        printf("\n");
    }

    return 0;
}
```

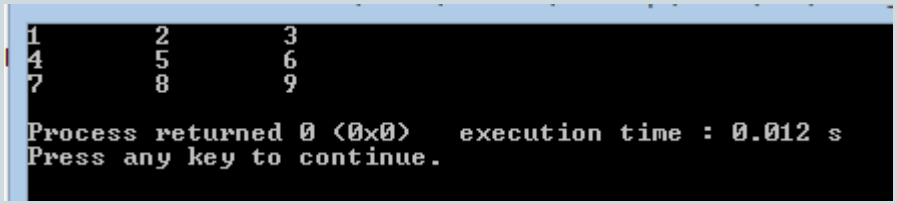
Matrizes

- Ao ser preenchida desta forma (linha representada pelo índice mais interno, coluna representada pelo índice mais externo) é feito preenchimento por coluna, ou seja:

	0	1	2	3	4
0	1^0 elemento	6^0 elemento	11^0 elemento	16^0 elemento	21^0 elemento
1	2^0 elemento	7^0 elemento	12^0 elemento	17^0 elemento	22^0 elemento
2	3^0 elemento	8^0 elemento	13^0 elemento	18^0 elemento	23^0 elemento
3	4^0 elemento	9^0 elemento	14^0 elemento	19^0 elemento	24^0 elemento
4	5^0 elemento	10^0 elemento	15^0 elemento	20^0 elemento	25^0 elemento

Matrizes

- Como funcionaria o preenchimento de uma matriz?
  - Executa uma vez o laço externo (i). Executa o laço interno completo (j).
  - Retorna ao laço externo e executa mais uma vez. Executa o laço interno completo novamente. E segue desta forma até executar o laço externo completamente.
- E se fosse necessário atribuir valores a matriz no momento da declaração de variáveis? Existem duas formas:
  - Atribuição de zero para toda a matriz. int matriz[3][3] = {0};
  - Atribuição de valores para a matriz:
    - int matriz[3][3] = {1,2,3,4,5,6,7,8,9}; . A impressão da matriz neste caso, ficaria:



Matrizes

- **Detalhe importante**: não é o nome da variável que está servindo de índice que representa a linha e coluna, e sim em que posição esta variável está sendo utilizada. Neste caso, se tivessemos:
  - matriz[coluna][linha] = 3;
- O índice de coluna está representando a linha da matriz, e o índice de linha está representando a coluna da matriz. Porque? A variável coluna está dentro do primeiro conjunto de colchetes, que representam o índice de linha, o mesmo ocorrendo com a variável linha. Então pode-se utilizar qualquer nome de variável para representar o índice, pois o que é observado é em que posição dos colchetes o índice se encontra.

Matrizes

- Faça um programa que preencha uma matriz 5x5 e exiba a diagonal principal.

```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <locale.h>

#define TAM 5

int main() {
    srand(time(NULL));
    setlocale(LC_ALL,"Portuguese");

    int mat[TAM][TAM],linhas, colunas;

    for (linhas=0; linhas<TAM; linhas++) {
       for (colunas=0; colunas<TAM; colunas++) {
            mat[linhas][colunas] = rand() % (TAM*TAM) + 1;
       }
    }

    for (linhas=0; linhas<TAM; linhas++) {
            printf("%d\t",mat[linhas][linhas] );
    }

    return 0;
}
```

Matrizes

- A maior parte do trabalho com uma matriz é manipulação dos índices.
- Neste exemplo foi solicitada a impressão da diagonal principal. Ao se analisar o que é uma diagonal principal pode-se perceber que nada mais é do que o elemento em que o índice de linha é igual ao da coluna.
- Neste caso os dois índices podem ser iguais.