

# 1º LABORATÓRIO DE CES-11 / 2013

## CTA - ITA - IEC

**Objetivo:** Trabalho com variáveis indexadas, ponteiros, passagem de parâmetros, alocação dinâmica de memória e recursividade.

**Tarefa:** As operações abaixo, efetuadas sobre **matrizes quadradas de inteiros** (onde o parâmetro **n** é a ordem delas), devem ser implementadas através de **funções recursivas**:

1) Maior elemento

```
int retornarMaiorElemento(matriz M, int n)
```

2) Transposição

```
matriz criarTransposta(matriz M, int n)
```

Obs: Retorna outra matriz (a matriz **M** de entrada não pode ser alterada).

3) Produto entre duas matrizes

```
matriz calcularProduto (matriz M1, matriz M2, int n)
```

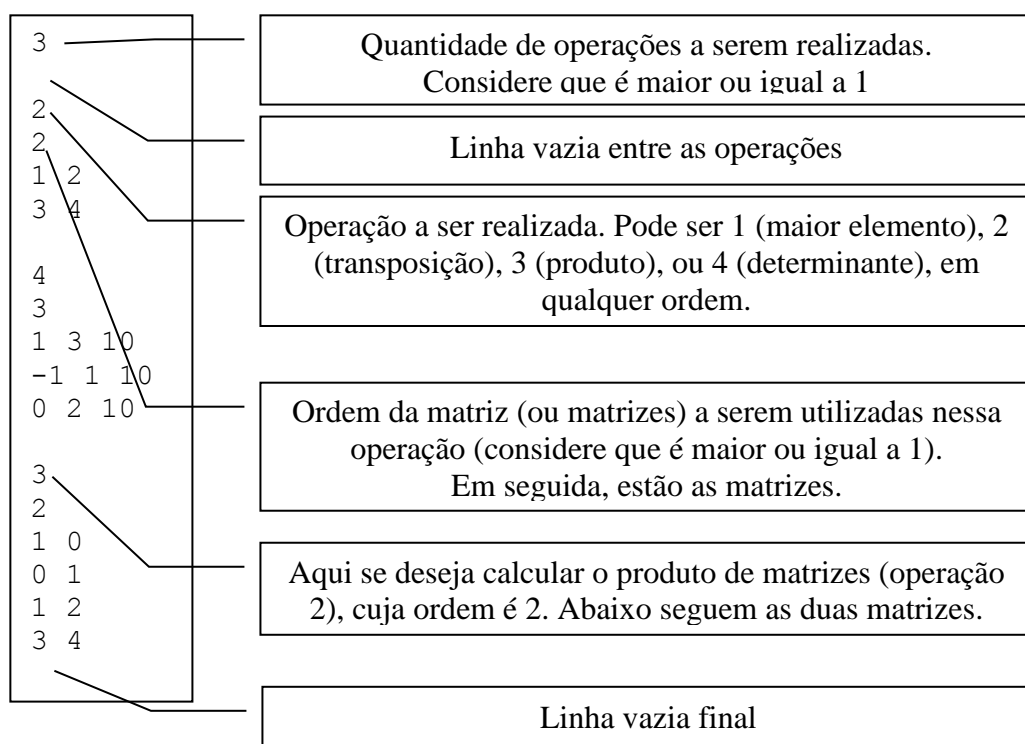
4) Determinante

```
int calcularDeterminante(matriz M, int n)
```

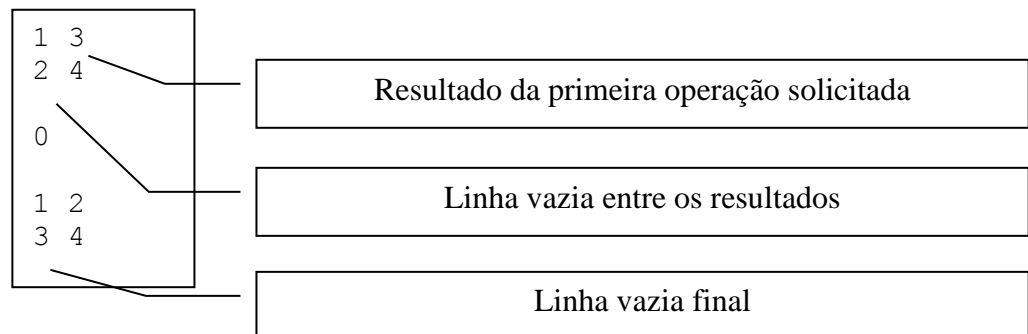
### Requisitos gerais:

- É proibido usar variáveis globais.
- Utilize as seguintes declarações de tipos:  

```
typedef int *vetor;  
typedef vetor *matriz;
```
- Utilize boas práticas de programação, o que inclui comentários no código.
- O programa deverá processar as operações indicadas no arquivo de entrada:
  - O nome é “**entrada.txt**” e deverá estar no mesmo diretório do executável.
  - O arquivo de entrada obedece ao formato abaixo:



- Para cada operação, seu programa deverá:
  - **Alocar dinamicamente** a(s) matriz(es) de entrada e então realizar as leituras.
  - **Alocar dinamicamente** a matriz resultante e calculá-la, exceto nas operações 1 e 4, cujas respostas não são matrizes.
  - Escrever o resultado da operação no arquivo de saída “saida.txt”
  - **Desalocar** as matrizes alocadas dinamicamente, quando não forem mais necessárias.
- Arquivo de “saida.txt” para o exemplo acima:



- Funções auxiliares que podem ser úteis:
  - `matriz alocarMatriz(int n)`
  - `void lerMatriz(matriz M, int n, FILE * entrada)`
  - `void escreverMatriz(matriz M, int n, FILE * saida)`
  - `void desalocarMatriz(matriz M, int n)`

### Requisitos das operações 1, 2 e 3:

- Deverão ser implementadas através de **funções recursivas**. Podem-se usar funções auxiliares.
- Em **nenhuma parte** do código dessas operações poderá haver comandos de repetição (**for**, **while**, **do-while**, etc.).

### Requisitos da operação 4:

- O determinante de uma matriz quadrada  $A$  de dimensão  $n$  deverá ser calculado através da seguinte **formulação recursiva** (*Regra de Laplace*):

$$Det(A, n) = \begin{cases} A[0][0], n = 1 \\ \sum_{i=0}^{n-1} (-1)^i \cdot A[0][i] \cdot Det(X, n-1), n > 1 \end{cases}$$

**X = MenorComplementar (A,n,0,i)**: retorna a matriz obtida pela eliminação da linha **0** (zero) e da coluna **i** da matriz **A** de ordem **n**.

Assim a dimensão da matriz menor complementar é de uma unidade a menos em relação à matriz **A**. Por isso, ao chamar de novo o cálculo do determinante, usa-se **n-1**.

**Atenção:** Lembre-se de desalocar a matriz **X** quando não for mais necessária.

Exemplo: Considere a matriz abaixo

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Aplicando a Regra de Laplace à linha 0, temos:

$$\begin{aligned} |B| &= 1 \cdot \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} - 2 \cdot \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + 3 \cdot \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix} \\ &= 1 \cdot (-3) - 2 \cdot (-6) + 3 \cdot (-3) = 0 \end{aligned}$$

**Entregar (através do TIDIA):**

- **Códigos fonte e executável:** esses arquivos deverão ter o nome do aluno e o número do laboratório.
- **Prazo:** 14 de agosto, 5<sup>a</sup>f, às 23h55. Desconto de 1 ponto na nota por dia de atraso.