

A prova é **individual** e todo o código submetido pelo aluno **deve ser de sua inteira autoria**. Caso seja identificado qualquer forma de plágio, a respectiva questão será zerada. Para responder às questões da prova é necessário utilizar o código em linguagem C a seguir.

Produto de matrizes

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 float **lerMatrizArquivo(char *nomeArquivo, int *m, int *n){
5 //Q1: Lê o arquivo cujo nome é passado no primeiro parametro (
6     nomeArquivo), retorna a matriz que está armazenada no arquivo lido e
7     atualiza os parametros m e n com as dimensões da matriz lida.
8 }
9 float **produtoMatrizes(float **A, int mA, int nA, float **B, int mB,
10     int nB){
11 //Q2: Calcula o produto de matrizes A * B. Retorna o resultado do
12     produto.
13 }
14 void imprimeMatriz(float **A, int m, int n){
15 //Q3: Imprime na tela a matriz passada como parâmetro.
16 }
17 }
18
19 int main(){
20     float **A, **B, **R;
21     int mA, nA, mB, nB;
22
23     A = lerMatrizArquivo( "matrizA.dat", &mA, &nA);
24     B = lerMatrizArquivo( "matrizB.dat", &mB, &nB);
25
26     if(nA == mB){
27         R = produtoMatrizes(A, mA, nA, B, mB, nB);
28
29         printf("A=A\n");
30         imprimeMatriz(A, mA, nA);
31         printf("B=B\n");
32         imprimeMatriz(B, mB, nB);
33         printf("A*B=A\n");
34         imprimeMatriz(R, mA, nB);
35
36         return 0;
37     } else {
38         printf("As dimensões das matrizes não permitem o cálculo do
39             produto entre elas na ordem pedida\n");
40         return 1;
41     }
42 }
```

O código encontra-se **incompleto** e cada questão solicita que uma parte do código seja finalizada. Caso todos os códigos sejam finalizados corretamente, o código completo irá calcular o produto de duas matrizes de números reais de dimensões potencialmente diferentes. Apenas o **código das funções `lerMatrizArquivo`, `produtoMatrizes` e `imprimeMatriz` precisam ser anexados à prova** e os demais códigos não devem ser alterados.

O produto de duas matrizes $A_{m \times k} \cdot B_{k \times n} = R_{m \times n}$ pode ser visto como:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kn} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}$$

Onde, para todo r_{ij} com $1 \leq i \leq m$ e $1 \leq j \leq n$, temos que

$$r_{ij} = \sum_{q=1}^k a_{iq} \cdot b_{qj}$$

As dimensões e o conteúdo de cada matriz deve ser lido de um arquivo. Cada arquivo de matriz terá várias linhas, sendo que, na primeira linha do arquivo aparecerão dois números inteiros que representam, respectivamente, o número de linhas e o número de colunas da matriz ali representada. As linhas seguintes consistirão de vários números reais representando o conteúdo da matriz, cada linha do arquivo representará uma linha da matriz e cada linha conterá a quantidade de números reais igual à quantidade de colunas da matriz.

Veja um **exemplo** de matriz 2x3 representada a seguir.

Matriz 2x3 representada num arquivo

```
1 2 3
2 1.4 2.5 6.2
3 9.9 -2.3 -0.2
```

Veja também um **exemplo** de matriz 3x4 representada a seguir.

Matriz 3x4 representada num arquivo

```
1 3 4
2 -1.3 4.2 -0.4 4.4
3 4.2 0.0 -0.1 0.6
4 -2.2 3.1 -4.0 3.0
```

Veja também um **exemplo** de matriz 3x3 representada a seguir.

Matriz 3x3 representada num arquivo

```
1 3 3
2 2.1 6.5 -9.4
3 4.2 4.4 6.0
4 0.3 -4.7 -7.7
```

Questão 1:

[40 pontos]

Considerando o código fornecido, termine a função *lerMatrizArquivo*. A função em questão deve receber o nome de um arquivo que contenha uma matriz, por exemplo "matrizA.dat", e dois parâmetros que sejam referências para inteiros que irão armazenar as dimensões (quantidade de linhas e quantidade de colunas) da matriz lida do arquivo. Após ler as dimensões da matriz armazenada no arquivo, a função deve alocar dinamicamente uma matriz do tamanho exato registrado no arquivo e armazenar os dados do arquivo na matriz alocada, além de atualizar as referências m e n para conterem as dimensões da matriz lida. A matriz lida deve ser retornada ao final da função.

Questão 2:

[35 pontos]

Considerando o código fornecido, termine a função *produtoMatrizes*. A função em questão deve receber a primeira matriz de reais, a quantidade de linhas da primeira matriz, a quantidade de colunas da primeira matriz, a segunda matriz de reais, a quantidade de linhas da segunda matriz e a quantidade de colunas da segunda matriz. A função deve então alocar a matriz que receberá o resultado do produto das duas matrizes. A matriz de resultados deve ter a mesma quantidade de linhas da primeira matriz e a mesma quantidade de colunas da segunda matriz. Após alocar a matriz resultado, a função deve calcular e armazenar nela o resultado do produto da primeira matriz pela segunda. A matriz que contém o resultado do produto das matrizes deve ser retornada ao final da função.

Questão 3:

[25 pontos]

Considerando o código fornecido, termine a função *imprimeMatriz*. A função em questão deve receber uma matriz de reais e dois parâmetros inteiros que representem as dimensões (quantidade de linhas e quantidade de colunas) da matriz. A função deve imprimir na tela formatadamente, cada linha da matriz numa linha na tela e cada coluna de uma linha alinhada com a coluna respectiva das linhas anteriores e posteriores, a matriz recebida como parâmetro. A função não deve retornar nada.

| | | | | |
|--------------------|----|----|----|-------|
| Questões | 1 | 2 | 3 | Total |
| Total de pontos | 40 | 35 | 25 | 100 |
| Pontos conseguidos | | | | |