

Discente: _____

1) (3 pts) Crie uma função em C que implemente a rotação à esquerda de um vetor em $n_positions$ posições. A função deve ter a seguinte assinatura:

```
void rotateLeft(int *arr, int
size, int n_positions);
```

- ***arr**: É o ponteiro para o início do vetor de inteiros a ser rotacionado.
- **size**: É o número total de elementos no vetor.
- **n_positions**: É o número de posições para as quais o vetor deve ser rotacionado à esquerda.

Ex. de entrada: Array inicial: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Número de posições para rotacionar: 3

Saída esperada: Array após a rotação: [40, 50, 60, 70, 80, 90, 100, 10, 20, 30]

2) (4 pts) Crie uma função em C para realizar a multiplicação de duas matrizes. Ambas as matrizes de entrada e a matriz resultante devem ser alocadas dinamicamente usando ponteiros para ponteiros ("int**"). A função deve verificar se a multiplicação é possível e, em caso negativo, retornar "NULL".

```
int** multiplyMatrices(int** mat1
, int rows1, int cols1,
int** mat2
, int rows2, int cols2);
```

- **mat1**: Ponteiro para a primeira matriz (alocada dinamicamente).
- **rows1**: Número de linhas da primeira matriz.
- **cols1**: Número de colunas da primeira matriz.
- **mat2**: Ponteiro para a segunda matriz (alocada dinamicamente).

- **rows2**: Número de linhas da segunda matriz.
- **cols2**: Número de colunas da segunda matriz.

Valor de Retorno:

- Um ponteiro para a nova matriz resultante (alocada dinamicamente), se a multiplicação for bem-sucedida.
- NULL, se a multiplicação não for possível (o número de colunas da primeira matriz não for igual ao número de linhas da segunda matriz).

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 10 & 11 \\ 20 & 21 \\ 30 & 31 \end{bmatrix} = \begin{bmatrix} 1 \times 10 + 2 \times 20 + 3 \times 30 & 1 \times 11 + 2 \times 21 + 3 \times 31 \\ 4 \times 10 + 5 \times 20 + 6 \times 30 & 4 \times 11 + 5 \times 21 + 6 \times 31 \end{bmatrix}$$

$$= \begin{bmatrix} 10+40+90 & 11+42+93 \\ 40+100+180 & 44+105+186 \end{bmatrix} = \begin{bmatrix} 140 & 146 \\ 320 & 335 \end{bmatrix}$$

Figura 1: Exemplo de multiplicação

3) (3 pts) Você precisa desenvolver um sistema em C para gerenciar dados de alunos, que incluem nome, matrícula e duas notas. Estes dados devem ser persistidos em um arquivo .txt para poderem ser salvos.

Defina uma struct chamada Aluno para armazenar as seguintes informações: nome, matrícula, nota1 e nota2.

Em seguida, implemente a seguinte função para Salvar Alunos em Arquivo de Texto:

```
void salvarAlunos(Aluno alunos[], int
numAlunos, const char*
nomeArquivo);
```

- **alunos[]**: É o array de Alunos a ser salvo.
- **numAlunos**: É o número de alunos no array.
- **nomeArquivo**: É o nome do arquivo de texto onde os dados serão salvos (ex: "dados_alunos.txt").