PROVA 3

MAC 2166 – INTRODUÇÃO À COMPUTAÇÃO (ELÉTRICA E COMPUTAÇÃO)

PRIMEIRO SEMESTRE DE 2026

Nome completo: _____

NUSP: _____ **Turma:** _____

Assinatura: _____

Instruções:

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis. Cuidado com a legibilidade.
3. A prova consta de **10 páginas** (incluindo a atual), totalizando **3 questões**. Verifique, antes de começar a prova, se o seu caderno de questões está completo.
4. Preencha o cabeçalho acima e os campos NUSP e Turma das folhas seguintes.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido o uso de artefatos eletrônicos de qualquer natureza.
7. Não é permitida a consulta a livros, apontamentos ou colegas.
8. Não é necessário apagar rascunhos no caderno de questões.

DURAÇÃO DA PROVA: 2 horas

Questão	Nota
1	
2	
3	
Total	

Q1. [**3 pontos**] Considere o seguinte código.

```
#include <stdio.h>
#include <stdlib.h>
void imprima_vetor(int N, int v[N]) {
    for (int i = 0; i < N; i++)
        printf("%d ", v[i]);
    printf("\n");
}
void imprima_matriz(int M, int N, int a[M][N]) {
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
}
int main(int argc, char *argv[]) {
    int x, num_dig = 0;
    int *NUSP2166 = malloc(24 * sizeof(int));
    while (scanf("%d", &x) != EOF) {
        NUSP2166[num_dig] = x;
        num_dig++;
    }
    imprima_vetor(num_dig, NUSP2166);
    printf("\n");

    int (*a)[2] = (int (*)[2])NUSP2166;
    imprima_matriz(4, 2, a);
    printf("\n");

    int (*b)[3] = (int (*)[3])NUSP2166;
    imprima_matriz(3, 3, b);
    printf("\n");

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            b[i][j] = b[i][j] + j;
    imprima_matriz(3, 3, b);
    printf("\n");

    imprima_vetor(num_dig, NUSP2166);
    free(NUSP2166);
    return 0;
}
```

Escreva aqui os dígitos de seu número USP, separados por espaços, seguidos dos dígitos `2 1 6 6`.

Dígitos NUSP seguidos de `2 1 6 6`: _____

Por exemplo, se seu NUSP fosse 12345678, você deveria escrever acima

```
1 2 3 4 5 6 7 8 2 1 6 6
```

Note que seu NUSP não é 12345678!

Determine a saída impressa pelo programa acima, supondo que a entrada é formada pelos seus dígitos NUSP seguidos de `2 1 6 6`, como você escreveu acima.

Saída:

NUSP: _____ **Turma:** _____

Q2. [3 pontos] Nesta questão, dizemos que dois números x e y *têm sinais opostos* se $x < 0$ e $y > 0$ ou $x > 0$ e $y < 0$. Note que isto acontece se e só se $xy < 0$.

Seja A uma matriz com entradas a_{ij}. Dizemos que a_{ij} e $a_{i'j'}$ são *vizinhos horizontais* se $i = i'$ e j e j' diferem de 1 (isto é $|j - j'| = 1$); dizemos que a_{ij} e $a_{i'j'}$ são *vizinhos verticais* se $j = j'$ e i e i' diferem de 1 (isto é $|i - i'| = 1$). Finalmente, a_{ij} e $a_{i'j'}$ são *vizinhos* se são vizinhos horizontais ou vizinhos verticais.

Queremos saber, para cada entrada a_{ij} de A, para quantos vizinhos $a_{i'j'}$ de a_{ij} vale que a_{ij} e $a_{i'j'}$ têm sinais opostos. Seja b_{ij} este número de "vizinhos opostos" de a_{ij}. Nesta questão, você deve escrever um programa que imprime a matriz B formada por esses valores b_{ij}.

Para verificar seu entendimento, estude o exemplo abaixo:

$$A = \begin{bmatrix} 0 & 2 & -2 & 1 & 0 \\ 0 & -1 & -1 & 2 & 0 \\ 0 & -1 & -2 & 0 & 1 \\ -1 & 0 & 0 & -2 & -2 \\ -1 & 0 & 1 & 0 & -1 \end{bmatrix} \quad \longrightarrow \quad B = \begin{bmatrix} 0 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

(*a*) Escreva uma função

```
int vizinhos_opostos(int N, int A[N][N], int i, int j)
```

que recebe uma matriz `A` de inteiros `N` × `N` e índices `i` e `j` dessa matriz, e que devolve o número de vizinhos de `A[i][j]` que têm sinais opostos (este é o valor de b_{ij} acima). [*Sugestão.* Considere `A[i - 1][j]` se e só se `i > 0`, considere `A[i + 1][j]` se e só se `i < N - 1`, etc.]

(*b*) Escreva aqui a função `main` de seu programa. Ela deve ler uma matriz A e imprimir a matriz B correspondente, como descrito no começo desta questão. Suponha que A seja uma matriz de inteiros $N \times N$. Seu programa deve ler da entrada padrão (1) o valor de N e (2) as N^2 entradas de A. Seu programa deve imprimir a matriz B dos b_{ij}. Use, obrigatoriamente, a função do item (*a*), mesmo que você não tenha resolvido aquele item.

NUSP: _____ **Turma:** _____

Q3. **[4 pontos]** Dizemos que uma matriz A é um *quadrado latino* de ordem N se todos os inteiros $0, 1, \ldots, N-1$ aparecem exatamente uma vez em **cada linha** e em **cada coluna** de A. Por exemplo, a matriz

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

é um quadrado latino de ordem 4. Você deve escrever um programa que lê N e uma matriz $N \times N$ e que determina se a matriz lida é ou não um quadrado latino.

(*a*) Implemente uma função `void transpoe(int N, int A[N][N])` que recebe um inteiro `N` e uma matriz `N` × `N` de inteiros `A` e que transpõe `A`; isto é, depois da chamada desta função, `A` deve conter a transposta da matriz original.

(*b*) Escreva uma função

```
void conta(int N, int A[N][N], int x, int c[N])
```

que recebe um inteiro `N`, uma matriz `N` × `N` inteira `A`, um inteiro `x` e um vetor inteiro `c` com `N` entradas, e que devolve em `c` o número de ocorrências de `x` em cada linha de `A`. Mais precisamente, para todo `0 <= i < N`, a função deve colocar em `c[i]` o número de vezes que `x` ocorre na linha de índice `i` de `A`. **Cuidado**: O vetor `c` não necessariamente começa zerado.

Espaço extra para a solução do item (b)**:**

(c) Escreva aqui a função `main` de seu programa. Sua `main` deve ler da entrada padrão (1) o valor de N e (2) as N^2 entradas de uma matriz $N \times N$ inteira A, e deve imprimir `SIM` caso A seja um quadrado latino e deve imprimir `NAO` caso contrário. Use, obrigatoriamente, as funções dos itens (a) e (b) acima, mesmo que você não tenha resolvido aqueles itens.

NUSP: _____ **Turma:** _____

Espaço extra para a solução do item (c):

(Rascunho)

NUSP: _____ **Turma:** _____

(Rascunho)