

PROVA 1
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 **8 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>

int main(int argc, char *argv[]) {
    int n;
    scanf("%d", &n);
    int quantidade = 0;
    int ultimo = n % 10;
    while (n > 0) {
        if (n % 10 != ultimo) {
            printf("%d %d\n", quantidade, ultimo);
            quantidade = 0;
            ultimo = n % 10;
        }
        quantidade++;
        n = n / 10;
    }
    printf("%d %d\n", quantidade, ultimo);
    return 0;
}
```

Determine a saída impressa por tal código para cada uma das seguintes entradas:

(a) 1234

Saída:

(b) 1112222

Saída:

(c) 89888988

Saída:

NUSP: _____ **Turma:** _____

Q2. [3 pontos] Dados dois números naturais n e d , dizemos que d é *divisor próprio* de n se $d < n$ e $\frac{n}{d}$ é inteiro. Um número $n \geq 2$ é *perfeito* se a soma dos seus divisores próprios é igual a n . Por exemplo, 6 é perfeito, pois seus divisores próprios são 1, 2 e 3, e $1 + 2 + 3 = 6$. Por outro lado, 12 não é perfeito, pois $1 + 2 + 3 + 4 + 6 = 16$.

(a) Escreva uma função `int somaDivProp(int n)` que devolve a soma dos divisores próprios de um número $n \geq 2$. Sua função não deve ler da entrada padrão nem imprimir na saída padrão.

(b) Escreva um programa (basta escrever o `main`) que recebe dois números naturais a e b com $2 \leq a < b$ na entrada padrão e imprime todos os números perfeitos no intervalo $[a, b)$, um por linha. Seu programa deve usar a função `somaDivProp` descrita no item anterior (você não precisa ter resolvido o item anterior para usá-la).

Q3. [4 pontos] Uma sequência de inteiros é *crescente* se todo elemento da sequência é menor que o próximo, e *decrecente* se todo elemento é maior que o próximo. Por exemplo, 1 3 4 7 é uma sequência crescente e 16 8 4 2 é uma sequência decrescente. Quando uma sequência não é nem crescente nem decrescente, alguns dos seus elementos podem ser vales ou picos:

- um elemento é um *vale* se é menor que os dois elementos adjacentes a ele;
- um elemento é um *pico* se é maior que os dois elementos adjacentes a ele.

O primeiro e o último elementos da sequência nunca são vales nem picos, por não terem dois elementos adjacentes. Por exemplo, a sequência 1 5 10 3 4 não é nem crescente nem decrescente. O elemento 10 é um pico (pois 10 é maior que tanto 5 quanto 3), e o elemento 3 é um vale (pois 3 é menor que tanto 10 quanto 4). Os elementos 1, 5 e 4 não são nem vales nem picos.

(a) Considere uma sequência a, b, c formada por exatamente três inteiros distintos. Escreva uma função `int monotona(int a, int b, int c)` que devolve 1 se a sequência é crescente, -1 se é decrescente, ou 0 caso contrário.

(b) Considere uma sequência a, b, c formada por exatamente três inteiros distintos. Escreva uma função `int valePico(int a, int b, int c)` que devolve 1 se b é um pico da sequência, -1 se b é um vale, ou 0 caso contrário.

NUSP: _____ Turma: _____

- (c) Escreva um programa que lê uma sequência de **pelo menos** três elementos da entrada padrão, **todos distintos**. Caso a sequência seja crescente ou decrescente, seu programa deve imprimir **Crescente** ou imprimir **Decrescente**, respectivamente. Caso contrário, seu programa deve imprimir uma linha para cada vale ou pico, na ordem em que aparecem na sequência. Toda sequência que não é crescente nem decrescente tem pelo menos um vale ou um pico. Por exemplo, para a sequência 1 5 10 3 4 2, seu programa deve imprimir

```
Elemento 10 é um pico
Elemento 3 é um vale
Elemento 4 é um pico
```

Seu programa deve ler os elementos até o fim do arquivo (EOF). Seu programa deve usar as funções dos itens (a) e (b), mesmo que você não tenha feito um (ou ambos) desses itens. Você deve escrever apenas o **main**.

Aviso: Arrays/vetores não são assuntos desta prova e não podem ser usados.

Espaço extra para a solução da Questão 3:

NUSP: _____ **Turma:** _____

(Rascunho)