

Lista de Exercícios 3 (2025/2)

Este trabalho consiste em resolver a lista de exercícios das páginas a seguir, em C.

Para a entrega devem ser seguidas as seguintes regras:

- criar um arquivo compactado no formato ZIP com o nome do aluno no formato *camelHump* (por exemplo, para João Pedro da Silva, usar `JoaoPedroDaSilva.zip`), **SEM SUBDIRETÓRIOS** e **APENAS COM OS ARQUIVOS DE CÓDIGO-FONTE** (NÃO envie quaisquer outros arquivos, como, por exemplo, arquivos `.class`, `.txt`, `README.txt`, `.o` ou executáveis);
- o código-fonte deve ser **ADEQUADAMENTE INDENTADO**;
- o arquivo compactado deve conter programas em C para resolver cada um dos exercícios, salvando o código-fonte em um arquivo com o nome `Exercicio` seguido do número do exercício com **TRÊS dígitos** (por exemplo, `Exercicio001.c`, `Exercicio002.c`, ..., `Exercicio050.c`, ..., `Exercicio101.c`, ...);
- **ATENÇÃO:** os exercícios NÃO seguem necessariamente uma sequência contínua, então tome cuidado de **USAR O NÚMERO CORRETO DO EXERCÍCIO NO RESPECTIVO ARQUIVO DE CÓDIGO-FONTE**;
- **NÃO USAR ACENTOS NO NOME DE ARQUIVOS E DE FUNÇÕES** ;
- no início de cada arquivo em C, incluir um comentário informando o nome do arquivo, o nome do autor, a finalidade do programa e a versão (ou data) de criação (ou atualização);
- quando houver dados a serem lidos, **LER OS DADOS SEMPRE NA MESMA ORDEM EM QUE ELES SÃO CITADOS NO ENUNCIADO**, escolhendo os tipos numéricos adequadamente (se houver dúvida entre usar um tipo inteiro ou um tipo real, use os exemplos de entradas e saídas que aparecem após cada exercício);
- **ESCREVER OS RESULTADOS SEMPRE NA MESMA ORDEM EM QUE ELES SÃO CITADOS NO ENUNCIADO**, escolhendo os tipos numéricos adequadamente (**NÚMEROS REAIS DEVEM SER IMPRESSOS SEMPRE COM 4 CASAS DECIMAIS**, a não ser que seja explicitamente indicado de outra forma);
- na versão final, tomar o cuidado de **NÃO IMPRIMIR NADA DIFERENTE DA SAÍDA ESPERADA** (não devem aparecer, por exemplo, mensagens pedindo que o usuário forneça ou digite determinado valor no terminal);
- a entrega deverá ser feita no dia e horário informado pelo professor em sala de aula e/ou definida na opção de entrega da plataforma moodle da PUCRS.

3. Considere uma linha no formato CSV (*Comma-Separated Values*) que use como separador o caractere ‘;’ (ponto-e-vírgula) e que contenha as seguintes informações sobre uma pessoa (exatamente na seguinte ordem): nome (até 40 caracteres úteis), idade (expressa como um valor inteiro), peso (em kg, expresso como um número real, usando ponto como separador de casas decimais) e altura (em metros, expressa como um número real, usando ponto como separador de casas decimais). Escreva um programa em C que leia uma linha nesse formato CSV, calculando o IMC (Índice de Massa Corporal) dessa Pessoa, usando a seguinte fórmula:

$$IMC = \frac{Peso}{Altura^2}$$

Mostre o resultado no seguinte formato: nome da pessoa, texto “: ” e IMC dessa pessoa (com 4 casas decimais).

Exemplo(s):

Teste	Entrada	Saída
1	José da Silva;22;68.0;1.75	José da Silva: 22.2041
2	Claudia Fonseca;35;82.5;1.80	Claudia Fonseca: 25.4630
3	Paulo Gonçalves;47;59.0;1.60	Paulo Gonçalves: 23.0469
4	Lúcia Machado;29;95.0;1.85	Lúcia Machado: 27.7575
5	Saulo Castro;18;50.0;1.55	Saulo Castro: 20.8117
6	Pablo Souza;63;72.0;1.70	Pablo Souza: 24.9135
7	Fernanda Fagundes;40;110.0;1.90	Fernanda Fagundes: 30.4709
8	Francisco Aguiar;31;77.0;1.68	Francisco Aguiar: 27.2817
9	Silvia Moreira;55;65.0;1.60	Silvia Moreira: 25.3906
10	Celso Melo;26;58.0;1.72	Celso Melo: 19.6052

4. Considere uma estrutura (struct) para armazenar dia, mês e ano para determinado dia (conforme declaração do exemplo abaixo) e implemente duas funções em C: uma para incrementar o dia (recebendo uma variável da estrutura por referência) e outra para mostrar o dia na saída padrão no formato DD/MM/AAAA (recebendo a variável da estrutura por valor).

Essas funções devem ser inseridas, e devem funcionar corretamente, no seguinte programa:

```
#include <stdio.h>

typedef struct {
    int dia, mes, ano;
} dia_t;

void incrementa_dia(dia_t *d);
void mostra_dia(dia_t d);

int main() {
    dia_t d;
    scanf("%d_%d_%d", &d.dia, &d.mes, &d.ano);
    incrementa_dia(&d);
    mostra_dia(d);
    return 0;
}
```

Observe que o código acima apresenta apenas os protótipos das funções. Este código lê uma data e chama as duas funções na sequência. Todas as datas na entrada são datas válidas. E todos os dias na saída devem ser datas válidas também.

Exemplo(s):

Teste	Entrada	Saída
1	31 1 2023	01/02/2023
2	28 2 2024	29/02/2024
3	29 2 2024	01/03/2024
4	30 4 2023	01/05/2023
5	21 8 2025	22/08/2025
6	31 12 2022	01/01/2023
7	30 6 2023	01/07/2023
8	30 9 2023	01/10/2023
9	30 11 2023	01/12/2023
10	27 2 2023	28/02/2023
11	28 2 2023	01/03/2023
12	31 12 2024	01/01/2025
13	31 7 2023	01/08/2023
14	31 10 2023	01/11/2023
15	31 3 2023	01/04/2023
16	31 8 2023	01/09/2023