

**CADERNO DE RESPOSTAS DA ATIVIDADE PRÁTICA DE:**

**Linguagem de Programação**

**ALUNO: [Nome] – [RU]**

**ANO**

**2025**

Prof. Winston Sen Lun Fung, Me.

|  |
| --- |
| **PRÁTICA 01** |
|  |
| Cálculo de Momento de Inércia de uma Seção Retangular (Engenharia) e Análise Básica. Neste exercício, você irá:   1. Solicitar ao usuário as dimensões (largura e altura) de uma viga com seção transversal retangular. 2. Calcular dois valores importantes na área de Engenharia Estrutural:    1. Área da seção transversal (b × h)    2. Momento de inércia dessa seção em relação ao eixo neutro, usando a fórmula: 3. Exibir os resultados e verificar se a área resultante atinge ou excede determinado valor de referência (por exemplo, 100 cm²), interpretando de forma simples se a viga poderia suportar uma “carga básica”.   Dessa forma, você exercitará tanto o cálculo de propriedades de uma seção retangular (conceito importante na Engenharia), quanto a implementação de entrada, processamento e saída de dados em Linguagem C.   1. **Objetivo**   - Ler a largura (b) e a altura (h) de uma viga retangular (em cm).  - Calcular:  • A área da seção transversal: A = b × h  • O momento de inércia em relação ao eixo neutro (seção retangular): I = (b × h³) / 12  - Verificar se a área ≥ 100 cm² (por exemplo), informando se está apta ou não para suportar uma “carga básica”.   1. **Passos Sugeridos**   a) Ler b e h do teclado (como float/double).  b) Calcular a área e o momento de inércia.  c) Exibir os valores calculados.  d) Exibir a mensagem se A >= 100: “A viga está apta…”; e a mensagem se A < 100: “A viga não está apta ...”.   1. **Exemplo de Saída**  |  | | --- | | **Digite a largura (cm): 10**  **Digite a altura (cm): 15**  **Area = 150.00 cm^2**  **Momento de inercia = 2812.50 cm^4**  **A viga esta apta para carga basica (A >= 100 cm^2).** |  1. **Demonstrando o funcionamento do programa**   - Informe a soma dois primeiros dígitos do seu RU para a largura.  - Informe a soma dois últimos dígitos do seu RU para a altura.  \* Caso a soma resulte em zero informe o valor 15. |
| 1. **Apresentação do código-fonte:** |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 02** |
|  |
| Análise de Leituras de Sensores c/ Alocação Dinâmica e Ponteiros  **1. Objetivo**  - Ler N leituras (ex.: temperaturas), usando malloc/calloc.  - Calcular a média, valor mínimo e valor máximo das leituras.  **2. Passos Sugeridos**  a) Pedir ao usuário “Quantas leituras?” (N).  b) Alocar dinamicamente (float \*leituras = malloc(N\*sizeof(float));).  c) Ler os N valores.  d) Calcular:  - Média (soma / N).  - Min e Max percorrendo o vetor.  e) Exibir todos os resultados.  **3. Exemplo de Saída**   |  | | --- | | Quantas leituras? 5  Digite as leituras:  22.5  24.0  21.8  23.2  25.1  Resultado:  - Média: 23.32  - Mínimo: 21.80  - Máximo: 25.10 |   **4. Demonstrando o funcionamento do programa**  - Informe a para “quantas leituras” a quantidade de dígitos do seu RU.  - As medidas deverão ser informadas pela composição: [digito do RU].[posição], informando, na ordem, do primeiro ao último dígito do seu RU.  Exemplo: RU 9876543  1° medida: 9.1  2° medida: 8.2  3° medida: 7.3  ... |
| 1. **Apresentação do código-fonte:** |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 03** |
|  |
| Criptografia Simples usando Ponteiros e Manipulação de Strings  **1. Objetivo**  - Ler uma frase (máx. 100 caracteres).  - Substituir vogais (A, E, I, O, U) por números (2,3,4,5,6).  - Substituir dígitos (0-9) por vogais, conforme um mapeamento que **você** definir.  - Percorrer a string via ponteiro de char.  **2. Exemplo de Saída**   |  | | --- | | Frase original: "Eu tenho 2 exames e 3 provas"  Frase criptografada: "36 t3nh4 i 3x2m3s 3 4 pr5v2s" |   (mapeamento ilustrativo; defina o seu)  **3. Demonstrando o funcionamento do programa**  - Para a frase original informe o seu nome completo e o seu RU. |
| 1. **Apresentação do código-fonte:** |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 04** |
|  |
| Imagine o projeto de um circuito eletrônico e para controle na sua fabricação necessita-se de uma lista de componentes de uma placa. Desenvolva uma Lista Encadeada de Componentes Eletrônicos.  1. Objetivo  - Implementar lista encadeada que guarda:  • nome[50]  • valor (por exemplo, ohms, microF etc.)  • tipo (resistor, capacitor etc.)  - Inserir quantos itens o usuário quiser (utilizar alocação dinâmica de memória).  - Ao final, salvar num arquivo CSV.  2. Passos  a) Definir struct nó (contendo nome, valor, tipo e ponteiro -> próximo).  b) Funções para inserir (no fim ou início) e exibir.  c) Um menu no main:  - Inserir componente  - Listar componentes  - Gerar arquivo CSV com a listagem de componentes (utilizador como separador o ponto e vírgula)  - Sair  3. Demonstrando o funcionamento do programa  - Cadastre 10 componentes, solicite a listagem em tela dos componentes, grave o arquivo CSV e mostre o seu conteúdo. |
| 1. **Apresentação do código-fonte:** |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 05** |
|  |
| Crie um Agendador Simples de Tarefas com Prioridade.  **1. Objetivo**  - Criar um “mini-sistema” que gerencia tarefas:  • idTarefa (int)  • descricao[100]  • prioridade (1=alta, 2=media, 3=baixa)  - Manter numa lista encadeada ou fila de prioridade.  - Inserir, remover tarefa de maior prioridade, listar e salvar em CSV, utilizador como separador o ponto e vírgula.  **2. Passos**  a) struct Tarefa {int id; char desc[100]; int prio; ...}  b) Incluir funções para inserir (no fim ou com base na prioridade) e remover (sempre a de maior prioridade).  c) Menu:  - 1) Inserir Tarefa  - 2) Remover Tarefa Prioridade Máxima  - 3) Editar uma tarefa  - 4) Listar  - 5) Sair (salvar CSV – utilizador como separador o ponto e vírgula).  d) Todos os dados devem ser salvos em um arquivo CSV quando sair do programa, ao iniciar o programa todos os dados salvos devem ser recarregados, o nome do arquivo CSV deve ser o seu RU.  **3. Exemplo de Saída**   |  | | --- | | Menu:  **1) Inserir Tarefa**  **2) Remover Tarefa Prioridade Máxima**  **3) Editar uma tarefa**  **4) Listar**  **5) Sair (salvar CSV).**  **Inserir Tarefa**  **ID\_Tarefa:  Descricao da Tarefa:**  **Prioridade [1-Alta, 2-Media, 3-baixa]:** |     **4. Demonstrando o funcionamento do programa**  - Cadastre 8 atividades diferentes, mostre as telas de cadastro, tela com a listagem de o conteúdo salvo no arquivo CSV. |
| 1. **Apresentação do código-fonte:** |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |
| 1. **Apresente a captura de tela evidenciando o arquivo CSV:** |