INF 1037 - Programação em C - Turmas 33A - 13/10/2025
Lista de Exercícios 08: Estruturas (*Structs*)

Observação:

- **As questões de 07 a 10 foram retiradas do capítulo 09 do livro "Introdução a Estruturas de Dados - Com Técnicas de Programação em C (Portuguese Edition)" - Waldemar Celes.**

Questão 01) Crie uma estrutura com nome (tag) Pessoa que possui os seguintes campos: nome, idade e altura. Você pode assumir o nome como sendo uma cadeia de caracteres com no máximo 50 caracteres, a idade como sendo um valor inteiro e a altura como sendo um valor ponto-flutuante. Escreva um programa que apenas lê os dados de 3 pessoas e imprime-os em formato de tabela.

Questão 02) Crie uma estrutura chamada Produto que possui os campos nome, preço e quantidade. Escreva um programa que fica em um laço criando e cadastrando novos produtos até que o usuário digita o valor -1 como quantidade. No final, o seu programa deve exibir o valor total do estoque (somatório de todos os produtos cadastrados)..

Questão 03) Crie uma estrutura chamada Data com dia, mês e ano. Escreva uma função que receba duas datas e indique qual vem primeiro. Faça uma função principal para receber estas datas via teclado, fornecidas pelo usuário, e também para exibir a saída do seu programa.

Questão 04) Crie uma estrutura chamada Aluno com os campos nome, matrícula e nota. Escreva um programa que lê os dados de 5 alunos. Adicionalmente,escreva funções para imprimir todos os alunos cadastrados, calcular a média de cada aluno e exibir o aluno com maior nota.

Questão 05) Crie uma estrutura chamada Livro com os campos título, autor e ano. Escreva um programa que lê os dados de 5 livros e exiba apenas os lançados após uma determinada data fornecida pelo usuário até que o usuário informe o valor -1 como ano.

Questão 06) Crie uma estrutura chamada Contato que possui os campos nome, telefone e email. Crie um programa que exibe, no início da execução, um menu com opções cadastrar, listar e buscar contato por nome para os usuários.

Questão 07) Considere estruturas para representar pontos no plano e círculos, dados pelos centros e raios.

A. Implemente uma função que determina se dois círculos se interceptam. A função deve receber os ponteiros para os dois círculos e retornar 1, se houver interseção, ou 0, caso contrário:

```
B. int intersecao (Circulo* a, Circulo* b);
```

C. Considere um vetor de pontos que representa uma linha poligonal. Implemente uma função que calcule o comprimento da poligonal:

```
D. float comprimento (int n, Ponto* v);
```

E. Implemente uma função que determina se o primeiro círculo contém o segundo. A função deve receber dois ponteiros para dois círculos e retornar 1, se verdadeiro, ou 0, caso contrário.

F. Implemente uma função que retorna um círculo cuja circunferência passa por dois pontos dados. A função deve receber ponteiros para dois pontos e retornar um círculo por valor (não o seu ponteiro; não tem alocação dinâmica).

Questão 08) Considerando uma estrutura para representar um ponto no espaço 2D, implemente uma função que indique se um ponto p está localizado dentro ou fora de um retângulo. O retângulo é definido por seus vértices inferior esquerdo v1 e superior direito v2. A função deve retornar 1, caso o ponto esteja localizado dentro do retângulo, e 0, caso contrário. Essa função deve obedecer ao protótipo:

```
int dentroRet (Ponto* v1, Ponto* v2, Ponto *p);
```

Questão 09 Considerando uma estrutura para representar um vetor no espaço 3D, implemente uma função que calcule o produto escalar de dois vetores. Essa função deve obedecer ao protótipo:

```
float escalar (Vetor* v1, Vetor* v2);
```

Questão 10) Defina estruturas para representar retângulos (dados a base e a altura) e círculos (dado o raio), e implemente as funções a seguir:

A. Dado um círculo, crie e retorne o maior retângulo inscrito de altura h; considere que h é menor do que o diâmetro do círculo:

```
Ret* ret_inscrito (Circ* c, float h);
```

B. Dado um retângulo, crie e retorne o maior círculo interno ao retângulo:

```
Circ* circ_interno (Ret *r);
```

Questão 11) Considerando as declarações a seguir para representar o cadastro de alunos de uma disciplina, implemente uma função que exiba na tela o número de matrícula, o nome, a turma e a média de um aluno e imprima uma mensagem que informe se o aluno está ou não aprovado.

```
typedef struct aluno Aluno;
struct aluno {
      char nome[81];
      char matricula[8];
      char turma;
      float p1;
      float p2;
      float p3;
};
```

Assuma que o critério para aprovação é dado pela média das três provas (p1, p2 e p3). A função recebe como parâmetro o aluno. Essa função deve obedecer ao protótipo:

```
void imprime_aluno (Aluno aluno);
```

Questão 12) Usando alguma estrutura Aluno criada por você, escreva um programa que, inicialmente, peça para o usuário informar quantos alunos deseja cadastrar. Em seguida, aloque dinamicamente um vetor de Aluno e leia os dados para popular o vetor. Depois, para cada aluno, exiba os dados e libere a memória.

Questão 13) Crie uma chamada Funcionario com os campos nome e salário. Aloque dinamicamente um vetor com um tamanho informado pelo usuário via teclado. Calcule e exiba a média salarial e o funcionário com maior salário.

Questão 14) Crie uma estrutura chamada Turma que possui os seguintes campos: nome da turma, quantidade de alunos e um ponteiro para vetor de Aluno, definido na questão 12. Escreva um programa que aloque dinamicamente a turma e com alunos. Seu programa deve ler e exibir os dados da turma.

Questão 15) Usando os conhecimentos aprendidos em sala, implemente um programa que use struct, typedef, vetores e malloc para cadastrar clientes de uma loja. Cada cliente deve conter nome, CPF, quantidade de compras e um ponteiro para um vetor de float (valores das compras). Seu programa deve calcular e exibir o total gasto por cliente.