

Lista Exercícios

Tipos Estruturados

1. Crie um struct do tipo carro de acordo com a figura 1.

```
1 # define MAX 100
2 typedef struct{ /* CARRO é o nome do tipo de estrutura */
3   char NomeDono [MAX]; /* string */
4   struct { /* um struct dentro de outro */
5     char RuaeNum [MAX]; /* Nome da rua e numero, string */
6     char Bairro [MAX];
7     char CEP [MAX];
8     char Cidade [MAX];
9     char Telefone [MAX];
10  } Endereco; /* nome de um campo, que é struct também */
11  char Modelo [MAX];
12  int Ano;
13  int Km; /* quilometragem atual */
14  char Fabricante [MAX]; /* nome do fabricante */
15  char Cor [MAX];
16  int NumPortas; /* número de portas */
17  int GasOuAlc; /* 1==gasolina, 2==alcool, 3==flex */
18  float Preco; /* preço atual de mercado, em reais */
19  char Chapa [MAX];
20 } CARRO; /* note o ; aqui. CARRO é o nome do tipo */
```

- Declarar um vetor chamado frota com até NUM_MAX elementos que sejam do tipo CARRO.
 - Atribuir (ou ler) os valores dos campos de todos os carros no vetor frota, de uma empresa locadora de carros.
 - Elabore um programa que ordena o vetor frota pelo número da placa.
 - Elabore um programa que lê um valor de placa, busca no vetor frota, utilizando busca binária, um carro com esta placa e, caso exista esse carro, lista todos os seus campos.
2. Considere uma estrutura para representar um ponto no espaço 2D e outra para representar um retângulo, composto por dois pontos que são, respectivamente, o vértice inferior esquerdo (vie) e o vértice superior direito (vsd). Crie uma função que recebe um retângulo e um ponto e verifica se esse ponto está dentro do retângulo. 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* vie, Ponto* vsd, Ponto* p);

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

float dot(Vetor* v1, Vetor* v2);

4. Considere as declarações a seguir para representar o cadastro de alunos de uma disciplina e implemente uma função que imprima o número de matrícula, o nome, a turma e a média de todos os alunos aprovados na disciplina.

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

typedef struct aluno Aluno;

Assuma que o critério para aprovação é dado pela média das três provas (p1, p2, p3). A função recebe como parâmetro o número de alunos e um vetor de ponteiros para os dados dos alunos. Essa função deve obedecer ao seguinte protótipo:

void imprime_aprovados(int n, Aluno turmas);**

5. Considere as declarações do tipo Aluno do exercício anterior e implemente uma função que tenha como valor de retorno a média final obtida pelos alunos de uma determinada turma. A nota final de cada aluno é dada pela média das três provas.

float media_turma(int n, Aluno turmas, char turma);**

6. Considere que uma empresa precisa armazenar os seguintes dados de um cliente:
- Nome completo com no máximo 50 caracteres;
 - renda mensal do cliente;
 - ano de nascimento;
 - possui ou não carro.

Defina um tipo e uma estrutura para armazenarem estes dados e escreva um programa que leia estes dados armazene-os em uma variável e em seguida os imprima.

7. Considerando a mesma estrutura do exercício anterior, escreva um programa que leia os dados de 100 clientes e imprima:
- quantos clientes têm renda mensal acima da média;
 - quantos clientes têm carro;
 - quantos clientes nasceram entre 1960 (inclusive) e 1980 (exclusive).
8. Escrever um programa que utilize structs e ponteiro para struct e imprima o conteúdo das variáveis da struct.
9. Fazer um programa que simule uma pilha push pop, usando structs. Um exemplo de entrada poderia ser o seguinte:

empilha C
empilha B
empilha A

desempilha A
desempilha B
desempilha C

Em uma pilha, o primeiro elemento a entrar é o último a sair e o último a entrar é o primeiro a sair.

10. Escreva um programa que solicite o nome e telefone de uma pessoa e grave essas informações num vetor de uma estrutura que contem esses dados (nome e telefone). O programa deve ter três opções apenas: uma que adiciona um novo dado, outra que lista todos os dados atualmente armazenados na memória e outra que sai do programa. Esse vetor de estrutura deve ter apenas 10 elementos e fornecer uma mensagem de erro caso o usuário tente adicionar mais pessoas que este máximo permitido.

11. Seja uma estrutura para descrever os carros de uma determinada revendedora, contendo os seguintes campos:

marca: string de tamanho 15

ano: inteiro

cor: string de tamanho 10

preço: real

a) Escrever a definição da estrutura carro.

b) Declarar o vetor vetcarros do tipo da estrutura definida acima, de tamanho N definido pelo usuário (usar malloc) e global.

Crie um menu para:

c) Definir uma função para ler o vetor vetcarros.

d) Definir uma função que receba um preço e imprima os carros (marca, cor e ano) e retorne um novo vetor com esses carros.

e) Defina uma função que leia a marca de um carro e retorne um novo vetor com todos os carros dessa marca (preço, ano e cor).

f) Defina uma função que leia uma marca, ano e cor e informe se existe ou não um carro com essas características. Se existir, informar o preço.

12. Seja uma estrutura para descrever os livros de uma loja, contendo os seguintes campos:

título: string de tamanho 15

autor: string de tamanho 20

estilo: string de tamanho 10

código: inteiro

preço: real

a) Escrever a definição da estrutura livro

Seja o seguinte trecho de programa:

```
// declaração de variáveis
```

```
struct livro vetlivro[50];
```

```
char ch; int numlivros = 0; // numero de livros
```

```
// definição da menu:
```

```
printf ( "Digite I para incluir um livro. \n" );
```

```
printf ( "Digite L para listar o nome de todos os livros. \n" );
```

```
printf ( "Digite A para procurar livro por autor. \n" );
```

```
printf ( "Digite T para procurar livro por título. \n" );
```

```
printf ( "Digite E para procurar livro por estilo. \n" );
```

```
printf ( "Digite M para calcular a média de preço de todos os livros. \n" );
```

```
printf ( "Digite S para sair \n";
```

```
}
```

b) Escrever as funções para atender o menu.

13. Elaborar um algoritmo que auxilie no controle de uma fazenda de gado que possui um total de N (lido do teclado) cabeças de gado. A base de dados é formada por um conjunto de estruturas (registros) contendo os seguintes campos referente a cada cabeça de gado:

- código: código da cabeça de gado,
- leite: número de litros de leite produzido por semana,
- alim: quantidade de alimento ingerida por semana - em quilos,
- nasc: data de nascimento - mês e ano,
- abate: 'N' (não) ou 'S' (sim).

O campo nasc. é do tipo struct data que por sua vez, possui dois campos:

- o mês
- o ano

Elaborar funções para:

- a) Ler a base de dados (código, leite, alim, nasc.mês e nasc.ano), e armazenar em um vetor de estruturas.
- b) Preencher o campo abate, considerando que a cabeça de gado irá para o abate caso:
 - tenha mais de 5 anos, ou;
 - produza menos de 40 litros de leite por semana, ou;
 - produza entre 50 e 70 litros de leite por semana e ingira mais de 50 quilos de alimento por dia.

Crie o menu de opções para:

- c) Devolver a quantidade total de leite produzida por semana na fazenda.
- d) Devolver a quantidade total de alimento consumido por semana na fazenda.
- e) Devolver a quantidade total de leite que vai ser produzido por semana na fazenda, f) Devolver a quantidade total de alimento que vai ser consumido por semana na fazenda
- g) Devolver número de cabeças de gado que irão para o abate.
- h) Sair do programa.

14. Seja um algoritmo para controlar os produtos do estoque de um supermercado. Para cada produto, tem-se os seguintes campos:

nome: string de tamanho 15
setor: caracter
quantidade: inteiro
preço: real //preço por unidade do produto

- a) Escrever a definição da estrutura produto.
- b) Declarar o vetor estoque do tipo da estrutura definida acima, de tamanho 100 e global.
Crie um menu para:
 - a) Definir um bloco de instruções para ler o vetor estoque.
 - b) Definir um bloco de instruções que receba um setor e devolva o número de diferentes produtos desse setor.

- c) Definir um bloco de instruções que calcule e devolva o total de capital investido em produtos do supermercado.
- d) Sair do Programa.

15. Seja um sistema destinado a controlar despesas e dados de um condomínio de apartamentos. Os dados de cada apartamento são armazenados em um vetor de estruturas. Para cada apartamento tem-se os seguintes dados:

- nome (do responsável)
- número (do apartamento)
- área (em m²)
- número de moradores
- valor (a ser pago no mês)

- a) Definir a estrutura acima
- b) Declarar um vetor de estruturas (global), supondo que o condomínio tem 40 apartamentos.

Crie um menu para:

- a) Escrever um bloco de instruções responsável por ler o vetor acima, exceto o campo valor.
- b) Escrever um bloco de instruções que retorne a área total do condomínio.
- c) Escrever um bloco de instruções que receba a área total do condomínio; leia o total de despesas do mesmo e calcule para cada apartamento o valor a ser pago no mês. Esse valor é proporcional à área do apartamento.
- d) Fazer um bloco de instruções para imprimir os dados do apartamento que tem o maior número de moradores. Em caso de haver mais de um, imprimir todos.
- e) Escrever um bloco de instruções para sair do programa.

16) Fazer um programa para simular uma agenda de telefones. Para cada pessoa deve-se ter os seguintes dados:

- Nome
- E-mail
- Endereço (contendo campos para Rua, numero, complemento, bairro, cep, cidade, estado, país)
- Telefone (contendo campo para DDD e número)
- Data de aniversário (contendo campo para dia, mês, ano)
- Observações : Uma linha (string) para alguma observações especial.

- a) Definir a estrutura acima.
- b) Declarar a variavel agenda (vetor) com capacidade de agendar até 100 nomes.
- c) Criar o menu para

- 1) Definir um bloco de instruções busca por primeiro nome: Imprime os dados da pessoa com esse nome (se tiver mais de uma pessoa, imprime para todas)
- 2) Definir um bloco de instruções busca por mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse mês.
- 3) Definir um bloco de instruções busca por dia e mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse dia e mês.

- 4) Definir um bloco de instruções insere pessoa: Insere por ordem alfabética de nome.
- 5) Definir um bloco de instruções retira pessoa: retira todos os dados dessa pessoa e desloca todos os elementos seguintes do vetor para a posição anterior.
- 6) Definir um bloco de instruções imprime agenda com as opções:
 - imprime nome, telefone e e-mail
 - imprime todos os dados.

O programa deve ter um menu principal oferecendo as opções acima.