

Lista IV – AEDSI

Structs

Prof. Fernando Augusto Teixeira

Baseada na lista de exercícios do Prof. Rone Ilídio

1) Implemente uma *struct* denominada Retangulo. Seus atributos serão altura e largura (*float*). Crie um *programa* onde o usuário insira a altura e a largura de duas variáveis do tipo retângulo. Este programa deve então exibir as áreas dos dois retângulos.

2) Crie um programa que possua uma função chamada *calcuraArea*. Ela deve receber como parâmetro uma variável do tipo Triângulo (*struct* com base e altura, ambos *float*) e retornar a área desse Triângulo. Crie 3 variáveis desse tipo e faça o usuário preencher todos esses valores. Exiba todos os dados, inclusive a área, do triângulo com maior área.

3) Implemente um programa que possua uma *struct* denominada *viagem*, com os campos: *distancia* (*float*) e *consumo*(*float*). Crie uma função denominada *kmlitro* que receba uma variável do tipo *viagem* e retorne quantos quilômetros foram rodados por cada litro de combustível consumido, ou seja, retorna o resultado da divisão de *distancia* por *consumo*. Crie um vetor do tipo *viagem* (1000 posições). Faça o usuário preencher todos os dados de todas as posições do vetor. Ao final, exiba tais dados na tela, inclusive o retorno da função para cada uma das posições do vetor. Exemplo, o resultado pode ser impresso da seguinte forma:

Distancia: 96	Consumo: 9,6	Média de consumo: 10
Distancia: 10	Consumo: 2	Média de consumo: 5

4) Crie uma *struct* denominada *solucao*, com dois campos: *elemento* (*char*[20]) e *porcentagem* (*float*). Crie um programa com duas variáveis do tipo solução. Faça o usuário preencher todos dos dados dessas variáveis. Exiba na tela todos os dados da variável que possuir maior *porcentagem*.

5) Crie um programa que possua uma *struct* de nome Aluno, com os campos nome (*char*[20]) e nota (*float*). Crie um vetor do tipo Aluno (10 posições). Faça o usuário preencher tal vetor com valores em todos os campos, depois exiba o nome e a nota de todos os alunos.

6) Crie um programa com uma *struct* chamada Produto. Tal *struct* deve possuir 3 campos: nome (*char*[20]), preco (*float*) e estoque (*int*). Faça com que o usuário preencha todos os dados desse vetor. Ao final, exiba os dados do produto mais caro e do produto mais barato.

7) Crie um programa que possua uma *struct* denominada Projeto, com os campos nome(*char*[20]), horastrabalhadas (*int*) e valorhora (*float*). Crie uma função denominada

valorProjeto, que recebe uma variável do tipo Projeto e retorne o resultado da multiplicação dos campos horastrabalhadas e valorhora. Crie um vetor do tipo Projeto. Crie um menu da seguinte forma:

- 1 - Inserir um projeto
- 2 - Exibir os dados do projeto cujo nome é informado pelo usuário
- 3 – Listar todos dados dos projetos, inclusive o resultado da função valorProjeto()
- 4 – Sair

Exigência: da uma das opções deve chamar uma função

8) Crie um programa que possua um *struct* denominado *Veículo*, com os seguintes campos:

- chassi – char [20]
- placa - char[7]
- fabricacao – int

O programa deve exibir o seguinte *menu*:

- 1 – Inserir um veículo
- 2 – Buscar veículo pela placa
- 3 – Sair

Se o usuário escolher a opção 1, UMA (só uma, não várias) nova variável do tipo Veiculo deve ser inserida em um vetor desse mesmo tipo (100 posições). Se a opção 2 for escolhida, o programa deve pedir que o usuário informe uma placa, buscar e exibir todos os dados do veículo que possuir a placa informada. Se ele escolher a opção 3, o programa deve ser encerrado.

Exigência 1: a inserção deve ser realizada dentro de uma função denominada *insereveiculo*, que recebe como parâmetro uma variável do tipo veículo, insere essa variável no vetor e retorna *true*. Se o vetor estiver cheio, a função retorna *false*.

Exigência 2: a busca através da placa deve ser realizada por uma função denominada *buscaporplaca*, que recebe a placa informada pelo usuário e retorna uma variável do tipo veículo.

9) Crie um programa que possua uma *struct* denominada *Aluno*. Ela deve possuir os seguintes campos:

- *nome*: char (50)
- *prova1*: float
- *prova2*: float

Os campos *prova1* e *prova2* receberão as notas do aluno na primeira e na segunda prova, respectivamente. Crie um vetor de 40 posições do tipo *Aluno*. Faça o usuário preencher todos os campos de todas as posições do vetor. Ao final, o programa deve exibir todos os alunos que passaram e todos os alunos que não passaram, separadamente. Considere que passou aquele aluno cuja soma das duas notas seja maior que 60 pontos.

Exigência: o cadastro de um novo aluno no vetor deve ser feito por uma função que recebe como parâmetro o nome e as notas.

10) Crie um programa que permita o cadastro de Cliente e seus respectivos Endereços e Projetos. Cliente, Endereço e Projetos devem ser structs com os seguintes dados:

- *Endereço*
 - *rua: string*
 - *bairro: string*
 - *numero: int*
 - *cidade: string*
- *Projeto*
 - *nome: string*
 - *valorProjeto: float*
 - *valorPago: float*
- *Cliente*
 - *nome: string*
 - *endereco: Endereco*
 - *projeto[10]: Projeto*

O programa deve permitir executar as seguintes opções:

1. Incluir Cliente/Endereço
2. Incluir Projetos
3. Listar Clientes/Endereço
4. Listar Projetos de um Cliente
5. Imprimir valor total de Projetos de todos os clientes
6. Imprimir valor recebido de Projetos de todos os clientes
7. Imprimir valor a receber de Projetos de todos os clientes
8. Sair

Teste seu programa com pelos menos 3 clientes e 2 projetos em cada cliente.