**3ª Lista - Algoritmos e Estruturas de Dados I**

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* (100 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:

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* (*string*) e *porcentagem* (*float*). Crie um programa com um vetor do tipo *solucao*. Faça o usuário preencher todos dos dados de todas as posições de vetor. Exiba na tela todos os dados ordenados pelo campo *elemento*.

5) Crie um programa com uma *struct* chamada Produto. Tal *struct* deve possuir 3 campos: nome (string), 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.

6) Crie um programa que possua uma struct denominada *Projeto*, com os campos nome(string), 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 valor de projeto (resultado da função valorProjeto())

4 – Sair

Exigência: cada uma das opções deve chamar uma função

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

* chassi – string
* placa - string
* 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.