

2ª Lista de Algoritmos e Estruturas de Dados II

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:

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

4) 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.

5) Crie uma agenda telefônica onde os dados são salvos em uma lista encadeada. Crie e utilize uma *struct* denominada contato com dois campos: nome (*string*) e telefone (*long*). Implemente as operações: inserir, listar todos os nomes e telefones, buscar telefone a partir do nome e excluir a partir do nome. Utilize lista simplesmente encadeada com cabeça.

6) Implemente um programa que controle os navios que estão desembarcando mercadorias em um porto. Crie uma lista encadeada a partir de uma *struct* denominada navio, com os campos: nome (*string*) e carga (*float*). Cada navio no porto deve representar um elemento na

lista encadeada. Implemente as funções: inserir navio, listar todos os dados dos navios e excluir navio. A operação de exclusão deve eliminar o navio cujo nome foi fornecido pelo usuário. No entanto, antes de ser eliminando, os dados desse navio devem ser exibidos na tela.

7) Faça um programa para manipular uma pilha :

- a) crie uma pilha;
- b) exiba o seguinte menu de opções:

EDITOR DE PILHA

- 1 - EMPILHAR
 - 2 - DESEMPILHAR
 - 3 - EXIBIR ELEMENTO DO TOPO
 - 4 - EXIBIR A PILHA
 - 5 - ESVAZIAR A PILHA
- DIGITE SUA OPÇÃO:

Tal programa deve sempre realizar as seguintes funções:

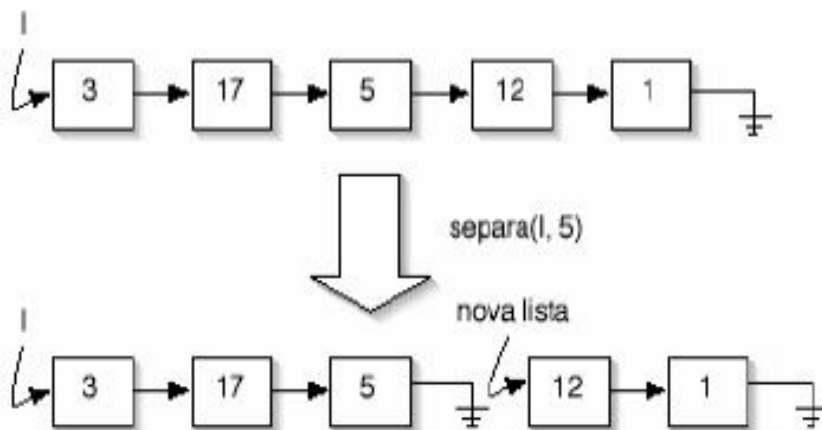
- a) ler a opção do usuário;
- b) executa a opção escolhida;
- c) exibir novamente o menu.

8) Em uma determinada biblioteca, todos os livros devolvidos são empilhados antes de serem colocados de volta nas prateleiras. Crie um programa que auxilie a bibliotecária a manter o controle para tais livros. Ele deve possuir uma pilha implementada sobre uma lista encadeada. Crie as funções: empilhar mais um livro e desempilhar um livro. Toda vez que chegar um livro na biblioteca ele deve ser colocado na pilha real. Consequentemente, seu nome deve ir para pilha criada no computador. Quando um livro é retirado da pilha real, seu nome deve ser retirado da pilha do computador. Se todos os livros forem retirados da pilha, o computador deve exibir a mensagem "Pilha vazia". Sempre exiba na tela o nome do livro que está no topo da pilha.

9) Crie um programa que mantenha a ordem dos atendimento em uma repartição pública. Ele deve implementar uma fila sobre uma lista encadeada. Tal fila deve simular a fila real dessa repartição. Crie as funções: enfileirar uma pessoa (pede o nome e coloca a pessoa no final da fila) e desenfileirar uma pessoa (retira a primeira pessoa da fila e exibe seu nome na tela do computador). Considere como informação útil somente o primeiro nome de cada pessoa. Exiba na tela quando a fila estiver vazia.

10) Crie um programa que possua 3 filas, cuja estrutura de dados de suas células é definida por uma struct com os seguintes campos: nome (string) e idade (int). Faça o usuário preencher a primeira fila com nomes mulheres e a segunda com nomes de homens. A partir de tais dados, preencha a terceira fila com os nomes de homens e mulheres que tenham idade maior ou igual a 18 anos. Exiba o conteúdo da terceira fila.

11) Considerando listas simplesmente encadeada de valores inteiros, implemente uma função que receba como parâmetro uma lista encadeada (ponteiro de início) e um valor inteiro n, e divida a lista em duas, de tal forma que a segunda lista comece no primeiro nó logo após a primeira ocorrência de n na lista original. A figura a seguir ilustra essa separação:



Crie um programa que recebe o usuário os valores da primeira lista, chame a função para dividi-la e exiba na tela as suas listas. (Fonte: Dept. Informática - PUC-Rio)