

Nome do aluno: \_\_\_\_\_ Data: 08/10/2012  
Resolva as questões abaixo identificando-as claramente na folha de respostas. Mantenha o silêncio na sala (mantendo desligado aparelhos eletrônicos). A interpretação das questões faz parte da prova. Horário limite: 15:50

(10 pontos) 1) O que é algoritmo?

(30 pontos) 2) Uma agenda telefônica é armazenada como uma lista encadeada (simples ou dupla). Um elemento desta agenda é apresentado abaixo. Crie as funções inicializa, insere, apaga, pesquisa e mostra. A função insere deve inserir elementos de forma que estes permaneçam ordenados em ordem crescente de **telefone**. A função apaga( char nome[]) deve apagar elementos baseado no nome do cliente e função pesquisa (char nome[]) deve pesquisar clientes baseado também no nome e retorna o endereço do bloco encontrado ou NULL caso não exista. A função mostra (struct tipo\_elemento \*ptr) mostra na tela o conteúdo de um registro (baseado no endereço que foi informado).

```
1 struct tipo_elemento {
2     char nome[20];
3     int telefone;
4     char endereco[20];
5     struct tipo_elemento *proximo;
6 };
```

(30 pontos) 3) Durante as aulas de estruturas de dados foram estudados alguns algoritmos de ordenação, entre estes, o método de ordenação por seleção. Quando estes métodos foram estudados, utilizou-se vetores para implementá-los. É claro que o método pode ser adaptado para ordenar elementos numa lista encadeada. Assim, pede-se que você implemente a função ordenada(Tipo\_Lista \*\*L) que ordena os elementos de uma lista encadeada utilizando o método de **seleção**. Note que você deve adaptar o método para funcionar como um lista encadeada e **não** copiar os elementos para um vetor auxiliar e ordená-los. Considere que as funções inicializa e insere estão disponíveis e os elementos já foram inseridos.

(30 pontos) 4) Um aluno muito esperto foi contratado para implementar módulos de uma biblioteca que fornece suporte às estruturas de Pilhas e Filas. Desta forma, um programador poderá utilizar a biblioteca para escrever programas que necessitem tanto das rotinas de Pilhas quanto das de Filas. Ocorre que esse aluno já desenvolveu à algumas semanas um módulo PILHAS que contém todas as estruturas de dados e suas respectivas funções. Assim, ele teve a idéia de ao invés de criar um módulo Filas desde a estaca zero, utilizar as funções de Pilhas para **emular** as funções de Filas e criar um módulo filas.c e filas.h (que internamente utilizam chamadas as rotinas de Pilhas). Construa a(s) estrutura(s) necessárias e implemente as rotinas de inserção e remoção de elementos na fila.