

2ª avaliação de Estrutura de Dados e Laboratório de Programação II – 10/11/2016

ALUNO (A) _____ Matrícula: _____

Turma de Estrutura de Dados: _____ Lab. de Prog II: _____

Leia atentamente as instruções abaixo:

- Fazer o download do arquivo Avaliacao2EDLab2016.zip do site do curso.
- Descompactá-lo em um diretório de sua máquina. Este arquivo contém todos os códigos para o desenvolvimento da prova. Você precisa saber em qual arquivo está a resposta de cada questão, pois ele deverá ser enviado ao servidor no final da prova.
- A resposta de cada questão deve, **obrigatoriamente**, estar entre cada par de marcador (//Qi, //-Qi). Assim, a questão 1 deve estar entre //Q1 e //-Q1, a questão 2 entre //Q2 e //-Q2 e assim por diante. Não remover tais marcadores de questão da sua prova. Caso sua solução tenha mais de uma função, elas devem estar entre esses marcadores.
- Colocar no arquivo main.cpp seu nome completo, e as turmas de ED e Lab. II.
- A prova é individual e sem qualquer tipo de consulta.
- Antes de sair do laboratório, enviar ao servidor – usando a janela de *upload* – cada arquivo de código que contém as respostas das questões da sua prova. Aguarde um momento e você verá as respostas de cada questão da sua prova.

Questões:

- 1) (25 Pontos) Considere o TAD `ListaCont`, que representa uma lista contígua de números inteiros em ordem crescente. Implemente a operação `insere(int valor)` desse TAD para inserir o `valor` na lista. Ao final da inserção a lista deve permanecer em ordem. Imprima uma mensagem de erro caso não seja possível inserir valor.

```
void ListaCont::insere(int valor);
```

- 2) (25 Pontos) Implemente a operação `removeMaior()` do TAD `ListaDupla` para remover o nó com maior valor de uma lista duplamente encadeada. A operação deve percorrer a lista uma única vez. Imprima uma mensagem de erro caso não seja possível remover o nó.

```
void ListaDupla::removeMaior();
```

- 3) (25 Pontos) Implemente a função `ordenaDecrescente(int *vet, int tam)` para ordenar de forma decrescente o vetor `vet` com os seus próprios `tam` elementos. Para isso, utilize os TADs `ListaCont` (Questão 1) e `PilhaEncad`.

```
void ordenaDecrescente(int* vet, int tam);
```

- 4) (25 Pontos) Implemente a operação `concatenaInicio(FilaEncad *fe)` do TAD `FilaEncad` para mover todos os elementos de `fe` para o início da fila implícita (objeto que chama a operação `concatenaInicio()`). Observar que ao final da operação, `fe` deve ficar vazia. A ordem de grandeza da complexidade dessa operação deve ser $O(1)$.

```
void FilaEncad::concatenaInicio(FilaEncad *fe);
```