

**1ª Avaliação Teórica: Estrutura de Dados - 03/05/2017 - (Profª Tamara)**

**Nome:** \_\_\_\_\_ **RA:** \_\_\_\_\_

Instruções: Leia atentamente todas as questões. Dê respostas claras e objetivas. Não é permitido consultar aos colegas ou qualquer tipo de material, caso isso ocorra, a(s) prova(s) será(ão) recolhida(s) e atribuído(s) valor(es) ZERO! A duração da avaliação é de no máximo 2h. A interpretação do problema também está sendo avaliada. Não serão válidas respostas fora do espaço designado (vide folha anexo os espaços para repostas). Faça com letra legível, o que eu não entender será desconsiderado. **Boa prova! Qualquer dúvida levante a mão e aguarde instruções.**

**Deve-se utilizar Linguagem C nas respostas relativas as implementações.**

1) [2,0] Assinale (V) para indicar que a afirmação é VERDADEIRA e (F) para FALSA. Se verdadeira, apenas reescreva a frase na folha de respostas; mas, se falsa, reescreva a sentença tornando-a verdadeira.

OBS: A cada duas erradas, anula-se uma certa. E, se assinalar (F) e não justificar corretamente, será considerado ERRADA a alternativa.

- Toda lista linear possui a ordem lógica dos elementos (a ordem ‘vista’ pelo usuário) sendo a mesma ordem física (em memória principal).
- Na estrutura de dados do tipo lista linear ordenada, todo elemento novo que é inserido torna-se o primeiro elemento da lista.
- Lista linear ligada, utilizando implementação estática, possui o procedimento de alocação/desalocação de memória para os registros sob demanda.
- Lista linear ligada duplamente encadeada possui o procedimento de alocação/desalocação de memória para os registros sob demanda.
- Em uma estrutura de lista, o último elemento armazenado deve ser o primeiro a ser removido.

2) Considere a lista linear duplamente encadeada (ordenada crescentemente), faça os procedimentos/funções (veja estrutura abaixo, porém fica a seu critério utilizá-la ou não; se não utilizar, defina a nova estrutura):

- A) **[1,0]** Inicializar a estrutura
- B) **[1,0]** Retornar a quantidade de elementos válidos e exibi-los  
(no mesmo procedimento/função)
- C) **[1,0]** Remoção do maior elemento
- D) **[2,0]** Inserção ordenada (crescentemente)
- E) **[1,0]** Exibir o maior elemento e fornecer a quantidade de elementos válidos  
(no mesmo procedimento/função)

```
typedef int TIPOCHAVE;

typedef struct{
    TIPOCHAVE chave;
}REGISTRO;

typedef struct aux{
    REGISTRO reg;
    struct aux* prox;
    struct aux* ant;
}ELEMENTO;
```

3) [2,0] Faça uma função que receba três listas, duas já preenchidas em ordem crescente e preencha a última com os valores das duas primeiras em ordem crescente.

1	4	5	10	11			
---	---	---	----	----	--	--	--

→ **Lista1**

5	8	7	11				
---	---	---	----	--	--	--	--

→
Lista2[illegible]

**4) [1,0 - Bônus]** Escreva um programa que remova todos os elementos com chaves pares de uma lista encadeada.  
DICA: operador % retorna o resto da divisão (se 0, o elemento é par)