

Universidade Federal da Fronteira Sul - UFFS  
Campus Chapecó  
Ciência da Computação  
Estrutura de Dados  
Prof.: Denio Duarte

## Instruções

1. Coloque apenas a sua MATRÍCULA na folha resposta.
2. Aparelhos eletrônicos desligados (ou no modo silencioso).
3. Responda TODAS as questões na folha resposta, enumerando as mesmas na ordem que melhor lhe convier.
4. As questões podem ser resolvidas a lápis porém o professor se reserva a não aceitar reclamações oriundas da correção das questões.
5. Consulta permitida apenas a cola oficial.

## Avaliação 02 - Recuperação

Use os tipos abaixo para desenvolver seus códigos:

```
struct tdado {                | struct tsent {
    int cod;                  |     struct tdado *first;
    char nome[30];           |     struct tdado *last;
    struct tdado *next, *prev; | };
};                             |

typedef struct tdado dado;
typedef struct tsent sentinela;
```

1. (4 pt) A diferença entre dois conjuntos (representados por listas encadeadas) gera um terceiro conjunto (outra lista encadeada) com os elementos pertencentes à primeira lista que não estão na segunda. Faça uma função que receba duas listas e retorne uma nova lista (representada por um sentinela - primeiro e último elemento da lista) com a divisão das listas passadas como parâmetros. O cabeçalho da função deve ser:  
`sentinela *Division(sentinela *s1, sentinela *s2)`, lembrando que um elemento da lista é um ponteiro do tipo `dado`. Verificar apenas o campo `cod`.
2. (3 pt) Escreva a função `dado *searchChange(sentinela *s1, int scod)` que procura na lista o código passado como parâmetro (`scod`) e, caso encontrá-lo, retorna o endereço do elemento além de promovê-lo para o primeiro elemento da lista. Se não encontrá-lo, a lista fica inalterada e é retornado `NULL`.

3. (3 pt) Na folha resposta, coloque a letra e o que deve ser colocado no espaço (linha tracejada) para a função funcionar. O novo elemento (**new**) já foi alocado e populado. A função insere um elemento da lista de forma ordenada pelo **cod**.

```
void insertOrder(sentinela *list, (a)____ *new)
{
    dado *aux;
    if (list->first==(b)_____)
    {
        list->first=new;
        list->last=new;
    } else {
        for (aux=list->first;aux!=NULL && new->cod > (c)_____;aux=aux->next);
        if (aux==NULL) {
            list->last->(d)_____=new;
            new->(e)_____=list->last;
            list->last=(f)_____;
            (g)_____;
        }
        if ((h)_____=list->first) {
            new->next=(i)_____;
            list->first->prev=(j)_____;
            list->first=new;
            return;
        }
        new->(k)_____=aux->prev;
        new->(l)_____=aux;
        aux->(m)_____->next=new;
        aux->(n)_____=new;
    }
}
```

**BOA PROVA & BOA SORTE.**