



LISTA DE EXERCÍCIOS – ALGORITMOS E ESTRUTURA DE DADOS III

Lista de Exercícios 6

DOCENTE: Thiago França Naves

DATA: __/__/__

ALUNO: _____

OBS: Todos os exercidos devem ser feitos utilizando o conceito de Lista Encadeada e Lista Encadeada Circular, com o projeto “ProjListaDinEncadDupla” disponibilizado junto com o material da aula.

- 1) Faça uma função que remova um aluno dada a sua respectiva matricula. Sua função deve reajustar os ponteiros *ant* e *prox* assim que remover o elemento, não esqueça de verificar se a elemento que será removido está em uma célula que é a primeira ou a última ou alguma célula do meio da lista, para cada posição o ajuste dos ponteiros é diferente.
- 2) Faça uma função que insira um novo elemento na lista de forma ordenada. Utilize a função *insere_lista_ordenada* da lista encadeada como base para criar essa função.
- 3) Faça um função que conte quantos alunos tem uma nota qualquer $n1$ igual, com o seguinte protótipo *int conta_lista_nota(Lista* li, int n1)*. A função deve utilizar apenas o ponteiro *li* passado como parâmetro, não pode ser criado nenhum outro ponteiro auxiliar. Ao final além de informar a quantidade alunos com a mesma nota, o ponteiro *li* deve estar apontado para a primeira célula “cabeça” da lista.
- 4) Refaça as funções de inserir e remover no início e fim para que a lista comporte-se de forma circular. No fim, faça uma função que para percorrer a lista inteira utilizando o comportamento circular da mesma.
- 5) Faça uma estrutura do tipo Fila utilizando o conceito de lista duplamente ou simplesmente encadeada, você pode escolher qual irá utilizar. É preciso criar as funções Enfileirar e Desenfileirar. Essas funções devem utilizar o conceito de nó descritor (*nodesc*), que é uma estrutura que controla onde está a célula inicial da lista e a célula final, a inserção e remoção na fila deve utilizar apenas os nós que estão na estrutura do nó descritor. Abaixo segue as duas estruturas que serão usadas, a função que inicializa o nó descritor e o cabeçalho das funções.

```

struct fila
{
    int info;
    struct fila *prox;
    struct fila *ant;
};

struct nodesc
{
    struct fila *ini;
    struct fila *fim;
};

int inicializa_noDesc(noDesc *no)
{
    *no = (noDesc)malloc(sizeof(struct nodesc));
    //coloca a fila no estado de vazia
    if(!(*no))
    {
        return 0;
    }
    else
    {
        (*no)->ini = NULL;
        (*no)->fim = NULL;
        return 1;
    }
}

int enfileirar(noDesc *n, int elem)

int desenfileirar(noDesc *n, int *elem)

```