



Atividade Avaliada 3 - Entrega pelo SIGAA

Data Entrega : até o dia 20/05 – Valor: 2.0 pontos no Semestre

1. Com base nas estruturas (Tarefa e LDE) e nos conceitos de Lista Duplamente Encadeada(LDE), implemente as funções:

```
typedef struct tarefa
{
    char descricao [30];
    int id, prioridade, concluida;
    struct tarefa *anterior;
    struct tarefa *proximo;
}Tarefa;

typedef struct LDE
{
    Tarefa *primeiro;
    Tarefa *ultimo; //facultativo
    int num;
}LDE;
```

*Figura 1: Estruturas de Lista
Duplamente Encadeada*

```
Tarefa* criaTarefa(int id);
LDE* criaListaLDE();
void mostraTarefa(Tarefa tf);
void insereInicio(LDE *lt, Tarefa *tf);
void insereFim(LDE *lt, Tarefa *tf);
void inserePosicao(LDE *lt, Tarefa *tf, int pos);
Tarefa* removeInicio(LDE *lt);
Tarefa* removeFim(LDE *lt);
Tarefa* removePosicao(LDE *lt, int pos);
void mostraListaED(LDE lt);
void mostraListaDE(LDE lt);
void apagaLista(LDE *lt);
void apagaElemento(Tarefa *tf);
void menu(LDE *lt);
```

Figura 2: Protótipos das Funções de Lista

- A) Adicionar um elemento no fim da Lista (LDE *lt, Tarefa *tf). Função sem retorno.
- B) Adiciona um elemento no início da Lista (LDE *lt, Tarefa *tf). Função sem retorno.
- C) Adiciona um elemento em uma dada posição "pos" (LDE *lt, Tarefa *tf, int pos). Retornar o código (1 para sucesso e 0 para Falha caso posição inválida);
- D) Remove um elemento no início da Lista (LDE *lt);. Retornar o endereço do elemento (*Tarefa) ou NULL para elemento inválido (lista vazia);
- E) Remove um elemento no final da Lista (LDE *lt);. Retornar o endereço do elemento (*Tarefa) ou NULL para elemento inválido (lista vazia);
- F) Remove um elemento de uma dada posição "pos" (LDE *lt, int pos);. Retornar o endereço do elemento (*Tarefa) ou NULL para elemento inválido ou lista vazia;
- G) Buscar um elemento contido na Lista (LDE *lt, char [] mod) (por modelo), Retornar a posição do elemento;
- H) Informa a quantidade de elementos da Lista (LDE *lt,). Retornar a quantidade de elementos;
- I) Mostrar a lista de elementos (LDE *lt);
- J) Apaga elemento, recebe o endereço do elemento e apaga o registro do sistema (FREE);
- K) Apagar a lista de elementos(LDE *lt); Deve remover todos os elementos da lista e apagar cada elemento; A lista ficará vazia após o processamento.

2. Implemente um sistema de menu para gerenciar as funções de Listas Simplesmente Encadeadas LDE, conforme protótipo das funções (Figura 2):

```
int menu(LSE *ls){
    printf("1 - Inserir um Elemento no Início:\n");
    printf("2 - Inserir um Elemento no Fim:\n");
    printf("3 - Inserir um Elemento na Posição:\n");
    printf("4 - Remove um elemento no Início:\n");
    printf("5 - Remove um elemento no Fim:\n");
    printf("6 - Remove um elemento na Posição:\n");
    printf("7 - Mostra Lista:\n");
    printf("8 - Apaga um Elemento da Lista\n");
    printf("9 - Apaga a Lista\n");
    printf("0 - Fim do Programa - Lista de Chamada:\n");
}
```

Figura 3: Menu de operações

3. Implemente um sistema para gerenciar Tarefas do dia. Crie uma lista com várias tarefas pré-cadastradas e diferentes prioridades (mínimo de 15 tarefas), após mostre:
 1. Todos os elementos com a mesma prioridade;
 2. Todos os elementos ordenados por prioridade (modo crescente);
 3. Os elementos que a tarefa já foi concluída;
 4. Verifique existência de tarefa repetida (descrição igual).

Bom trabalho!!!