



## **Atividade Avaliada 2 - Entrega pelo Moodle**

**Data Entrega : até o dia 26/09 – Valor: 2.0 pontos no Semestre**

1. Com base nos conceitos de Lista Simplesmente Encadeada(LSE), implemente uma agenda de contatos de telefones, utilize as estruturas abaixo:

```
typedef struct contato{
    char nome[20];
    char telefone [15];
    char tipo;
    struct contato *proximo;
}Contato;

typedef struct lse{
    Contato *primeiro;
    int n_elementos;
}LSE;
```

2. Implemente as funções abaixo para controlar a agenda telefônica:
  - A) Adicionar um contato no fim da Lista (LSE \*ls, Contato \*novo). Função sem retorno.
  - B) Adiciona um contato no início da Lista (LSE \*ls , Contato \*novo). Função sem retorno.
  - C) Adiciona um contato em uma dada posição “pos” (LSE \*ls, Contato \*e, int pos). Retornar o código (1 para sucesso e 0 para falha, caso posição inválida);
  - D) Remove um contato no início da Lista (LSE \*ls). Retornar o endereço do contato removido ou NULL para contato inválido (lista vazia);
  - E) Remove um contato no final da Lista (LSE \*ls). Retornar o endereço do contato ou NULL para contato inválido (lista vazia).
  - F) Remove um contato de uma dada posição “pos” (LSE \*ls, int pos). Retornar o endereço do contato removido ou NULL para contato inválido ou lista vazia;
  - G) Buscar um contato contido na Lista (LSE \*l, char [ ] nome) (por nome), Retornar a posição do contato caso seja encontrado ou -1 para contato não encontrado;
  - H) Retorna a quantidade de contatos da Lista (LSE \*l). Retornar a quantidade de contatos;
  - I) Mostrar a lista de contatos (LSE \*l);
  - J) Apaga contato, recebe o endereço do contato e apaga o registro do sistema;
  - K) Divide a lista, recebe uma lista (LSE \*l, int inicio, int fim), retorna um sublista com os contatos contidos entre os índices início e fim;  
Obs: A lista recebida por parâmetro (\*l) terá os elementos entre as posição **início e fim** retirados e inseridos na nova lista, que será retornada por parâmetro (ponteiro de lista LSE\*).

3. Implemente um sistema de menu para gerenciar as funções de Listas Simplesmente Encadeadas LSE, do item 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 1: Menu de operações*

4. Implemente um sistema para gerenciar o cadastro de contatos de um celular. O sistema deve controlar uma lista de contatos, cada nó terá as informações (nome,sobrenome, telefone, tipo).
- A) Os novos contatos por padrão serão inseridos em ordem alfabética, como base no seu nome.
- B) Implemente o menu do sistema de agenda:
- 1) Inserir novo contato( Insere um contato com base nome e em ordem alfabética);
  - 2) Remover contato (Remove um contato com base nome de um contato);
  - 3) Consultar um contato pelo nome (busca e mostra um contato por nome);
  - 4) Mostrar lista ordenada de contatos por nome;
  - 5) Criar o grupo da família. O grupo será uma nova lista como os elementos copiados da lista principal que possuem o atributo "tipo" com valor igual a "F" (Família).
- Os nós do grupo família serão criados a partir da cópia dos dados dos nós da lista principal.
  - Sempre que a opção for selecionada uma nova lista é criada.
  - Após a criação da lista, a função irá mostrar os elementos da nova lista (grupoda-Familia).

Bom trabalho!!!