**LISTAS LINEARES SIMPLEMENTE ENCADEADAS**

Considere a seguinte declaração:

struct noh {

int info;

struct noh \*prox;

};

typedef struct noh \*LLSE;

Utilizando a estrutura de dados (definida acima), faça um programa em linguagem C para manipular listas lineares simplesmente encadeadas. O programa deverá contemplar as seguintes funcionalidades:

1. Criar uma lista vazia (nula)
2. Inserir dados no início da lista;
3. Inserir dados no final da lista;
4. Inserir dados de forma ordenada na lista;
5. Inserir dados em uma posição informada na lista. Obs: caso não exista a posição um erro deve ser retornado pela função e o programa deverá trata-lo, informando ao usuário;
6. Remover nós do início da lista;
7. Remover nós do final da lista;
8. Remover nós pelo valor informado. Obs: caso hajam mais de um nó com o valor informado, remover todos;
9. Remover um nó pela posição informada. Obs: caso não exista a posição um erro deve ser retornado pela função e o programa deverá trata-lo, informando ao usuário;
10. Ordenar uma lista por substituição dos valores;
11. Ordenar uma lista por substituição dos endereços dos nós;
12. Inverter os nós da lista. Obs: Inversão dos ponteiros, NÃO trocar os valores dos respectivos nós;
13. Separe uma lista, a partir de um valor informado. Obs: ao final restarão duas listas, a primeira com os elementos do início da lista até o nó cujo valor foi informado e a segunda com os elementos após o nó informado;
14. Separe uma lista a partir do índice da posição informada. Obs: ao final restarão duas listas, a primeira com os elementos do início da lista até o nó cuja posição foi informada e a segunda com os elementos após o a posição informada;
15. Concatene duas listas. Obs: ao final restará apenas uma lista com os elementos da primeira, seguido dos elementos da segunda;
16. Intercale duas listas; Obs: a nova lista será criada com o primeiro nó da primeira lista seguido do primeiro nó da segunda lista, seguido do segundo nó da primeira lista, depois do segundo nó da segunda lista e assim sucessivamente. Caso restem nós em uma das listas originais, eles deverão ser inseridos no final da lista resultante.
17. Separe uma lista em duas, adotando o principio inverso da intercalação (p);
18. Trocar dois elementos de posição (a partir da informação da posição dos dois elementos);
19. Trocar dois elementos de posição (a partir dos seus valores);
20. Destruir a lista. Obs: Remover da memória todos os seus nós.