Para todos os exercícios dessa lista considere a seguinte declaração:

```
typedef struct lista{
 int info;
 struct lista *prox;
 }Lista;
```

1. Escreva uma função que recebe o ponteiro para uma lista encadeada, e retorna o número de células da lista.

2. Escreva uma função que recebe uma lista, como parâmetro, faz uma cópia da lista dada e retorna o ponteiro para a nova lista. Atenção, a lista resultante deve ser igual à original (os nós devem estar na mesma ordem).

3. Escreva uma função que receba, como parâmetros, duas lista L1 e L2 e concatene L1 no final de L2. Utilize o protótipo: Lista *concatena(Lista *L1, Lista *L2)

4. Escreva uma função que receba, como parâmetros, um vetor V com n elementos e construa uma lista encadeada com os n elementos do vetor. Se n=0 a função deve retornar uma lista vazia. Exemplo: Se o vetor dado for V={4, 1, 5, 9, 13, 26} a função deve retornar uma lista onde o primeiro elemento da lista é 4, o segundo é 1 e o último é 26.

5. Escreva uma função que receba, como parâmetros, uma lista e um número inteiro não negativo n, remova da lista os n primeiros nós e retorne a lista resultante.

6. Escreva uma função que receba, como parâmetros, uma lista e um número inteiro n, e divida a lista em duas, de forma que a segunda lista comece no primeiro nó logo após a ocorrência de n na lista original. A função deve retornar um ponteiro para a segunda subdivisão da lista original, enquanto L (cabeça da lista original) deve continuar apontando para o primeiro elemento da primeira subdivisão da lista.

7. Escreva uma função que inverte a ordem das células de uma lista encadeada (a primeira passa a ser a última, a segunda passa a ser a penúltima etc.). Faça isso sem usar espaço auxiliar; apenas altere os ponteiros.