



**Curso:** Análise e Desenvolvimento de Sistemas

**Série:** Módulo II

**Turno:** Noite

**Professor:** Jurandir Cavalcante Lacerda Júnior

### Lista de Exercícios – Estrutura de Dados Lista Encadeada

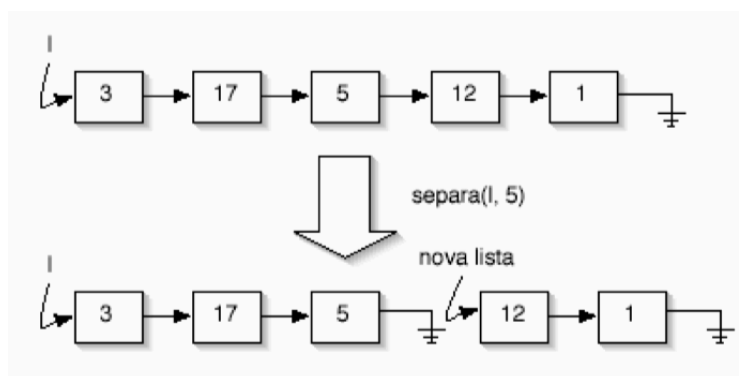
1. Implemente uma função que tenha como valor de retorno o comprimento de uma lista encadeada, isto é, que calcule o número de nós de uma lista. Essa função deve obedecer o protótipo:

```
int comprimento (Lista* l);
```

2. Considere listas encadeadas de valores inteiros e implemente uma função para retornar o número de nós da lista que possuem o campo INFO com valores maiores do que N. Essa função deve obedecer o protótipo:

```
int maiores (Lista* l, int n);
```

3. Considerando listas de valores inteiros, implemente uma função que receba como parâmetro uma lista encadeada e um valor inteiro n e divida a lista em duas, de tal forma que a segunda lista comece no primeiro nó logo após a primeira ocorrência de n na lista original. A figura a seguir ilustra essa separação:



Essa função deve obedecer ao protótipo:

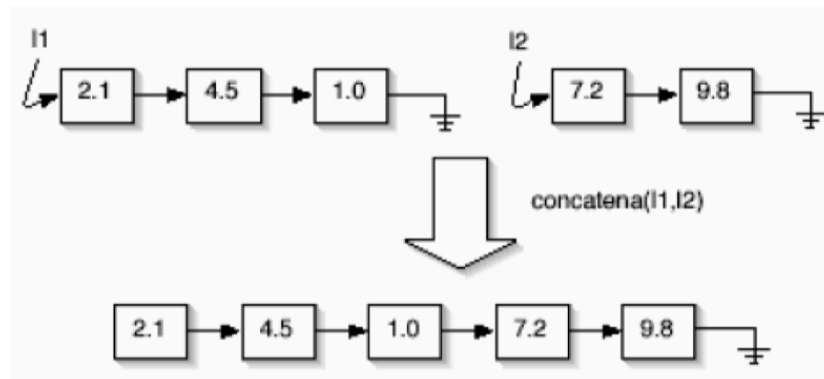
```
Lista* separa (Lista* l, int n);
```

A função deve retornar um ponteiro para a segunda sub-divisão da lista original, enquanto l deve continuar apontando para o primeiro elemento da primeira sub-divisão da lista.

4. Implemente uma função que tenha como valor de retorno o ponteiro para o último nó de uma lista encadeada. Essa função deve obedecer ao protótipo:

```
Lista* ultimo (Lista* l);
```

5. Implemente uma função que receba duas listas encadeadas de valores reais e retorne a lista resultante da concatenação das duas listas recebidas como parâmetros, isto é, após a concatenação, o último elemento da primeira lista deve apontar para o primeiro elemento da segunda lista, conforme ilustrado a seguir:



Essa função deve obedecer ao protótipo:

```
Lista* concatena (Lista* l1, Lista* l2);
```