

## Lista de exercícios 2

1 – 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  $n$ . Protótipo da função: `int maiores(Lista* l, int n);`.

2 – 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: `List* ultimo (List* l);`.

3 – Implemente uma função que recebe duas listas encadeadas de valores reais e retorna a lista resultante de concatenação das duas listas recebidas como parâmetro, isto é, após a concatenação, o último elemento da primeira lista deve apontar para o primeiro elemento da segunda lista. Protótipo da função: `List* concatena (List* l1, List* l2);`

4 – Considere listas de valores inteiros e implemente uma função que receba como parâmetros uma lista encadeada e um valor inteiro  $n$ , retire da lista todas as ocorrências de  $n$  e retorne a lista resultante. Protótipo: `List* retira_n(Lista* l, int n);`.

5 – Podemos dizer que uma lista encadeada não vazia é representada por um elemento seguido de uma sublista, ou seja, o segundo elemento da lista representa o primeiro elemento da sublista. Implemente uma função recursiva para imprimir os elementos de uma lista encadeada.

6 – Implemente as funções para inserir e retirar um elemento de uma lista circular duplamente encadeada.