

Listas Encadeadas: Rotação

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

Dadas duas listas encadeadas, escreva uma função que verifique se a segunda lista é uma rotação da primeira. Uma lista é uma rotação de outra se a última pode ser obtida da primeira removendo zero ou mais elementos de sua cauda e os inserindo em sua cabeça. Por exemplo, a lista 1->2->3->4 é uma rotação da lista 3->4->1->2. Por definição, uma lista é uma rotação de si mesma e uma lista vazia é uma rotação de outra lista vazia. A função deve ter a seguinte assinatura:

```
bool list_is_rotation(list_t* l1, list_t* l2);
```

Considere a seguinte definição de lista encadeada:

```
typedef struct list_node_t {  
    int data;           /*Dado da lista*/  
    struct list_node_t *next; /*ponteiro para o próximo elemento*/  
} list_node_t;  
  
typedef struct list_t {  
    list_node_t *head; /*Cabeça da Lista*/  
    list_node_t *tail; /*Cauda da Lista*/  
    size_t size;      /*tamanho da lista*/  
} list_t;
```

Assuma que as seguintes funções estão disponíveis:

```
size_t list_size(list_t *l); // retorna o tamanho da lista  
bool list_empty(list_t *l); // retorna verdadeiro sse a lista está vazia
```

Notas

Só é necessário implementar a função pedida no enunciado. Não é necessário realizar a leitura dos dados ou escrever a função `main`. Funções auxiliares, que tenham a ver apenas com a função solicitada, podem ser implementadas, caso deseje.