

Atividade Avaliativa - Listas Encadeadas

Exercício 1: Soma dos Elementos

Implemente uma função em C que calcule a soma de todos os elementos de uma lista encadeada. Use as funções fornecidas no header para manipular a lista.

Protótipo:

```
int list_sum(List *L);
```

Exercício 2: Inverter a Lista

Implemente uma função que inverta os elementos de uma lista encadeada, de modo que o primeiro elemento passe a ser o último e vice-versa.

Protótipo:

```
void reverse_list(List *L);
```

Exercício 3: Mesclar Duas Listas

Crie uma função que mescle duas listas encadeadas ordenadas em uma terceira lista também ordenada.

Protótipo:

```
List* merge_sorted_lists(List *L1, List *L2);
```

Exercício 4: Remover Elementos Duplicados

Escreva uma função que remova elementos duplicados de uma lista encadeada, mantendo apenas a primeira ocorrência.

Protótipo:

```
void remove_duplicates(List *L);
```

Exercício 5: Verificar Palíndromo

Implemente uma função que verifica se uma lista encadeada representa um palíndromo. Exemplo: para a lista 1 -> 2 -> 3 -> 2 -> 1, a função deve retornar 1 (verdadeiro).

Protótipo:

```
int is_palindrome(List *L);
```

Exercício 6: Rotacionar a Lista

Implemente uma função que rotacione a lista encadeada à esquerda em k posições. Por exemplo, a lista 1 -> 2 -> 3 -> 4 -> 5 rotacionada em 2 posições se torna 3 -> 4 -> 5 -> 1 -> 2.

Protótipo:

```
void rotate_list(List *L, int k);
```

Exercício 7: Intercalar Listas

Crie uma função que intercale os elementos de duas listas encadeadas, alternando os elementos das duas listas. Caso uma das listas termine antes, os elementos restantes da outra lista devem ser mantidos.

Protótipo:

```
List* intercalate_lists(List *L1, List *L2);
```

Exercício 8: Encontrar o K-ésimo Elemento

Implemente uma função que retorne o valor do k-ésimo elemento de uma lista encadeada. Se o índice for inválido, retorne um valor especial, como -1.

Protótipo:

```
int get_kth_element(List *L, int k);
```

Exercício 9: Dividir a Lista

Crie uma função que divida uma lista encadeada em duas listas: uma com os elementos pares e outra com os ímpares.

Protótipo:

```
void split_list(List *L, List **even_list, List **odd_list);
```

Exercício 10: Ordenar Lista

Implemente uma função para ordenar os elementos de uma lista encadeada em ordem crescente utilizando o algoritmo de sua escolha, como Bubble Sort ou Merge Sort.

Protótipo:

```
void sort_list(List *L);
```

Dicas de Implementação:

- Use as funções já definidas no header, como `list_insert` e `list_remove`, para manipular os elementos da lista.
- Teste cada função com diferentes casos de teste, incluindo listas vazias, listas com um único elemento e listas grandes.