

Estruturas de Dados (COMP0405)

Exercícios 3 (Lista Encadeada)

Professor: Alberto Costa Neto

Em todos os exercícios, deve-se partir da implementação apresentada em sala de aula, que pode ser encontrada no seguinte endereço:

<https://replit.com/@ALBERTOCOSTA3/ListaEncadeada>

1) Implementar uma função **inserirNoFinal** para incluir um item no final da lista. Para isso, utilize um apontador que mantenha o endereço do **último** elemento da lista. Observe que tendo este apontador, a inserção é rápida e fácil, já que o novo item será colocado após o último e será o novo **último** da lista. Fique atento para a remoção do último valor de uma lista. Neste caso, a remoção tem que ajustar o apontador **último** para o nó que anteriormente era o penúltimo.

2) Implementar uma função para remover o item de uma certa posição da lista. A função deve checar se a posição é válida, ou seja, se é referente a uma posição existente na lista. Caso a remoção seja bem-sucedida, a função retorna **true** e o parâmetro **item** contém o valor que foi removido da lista. Caso contrário, a função retorna **false** e a lista não será alterada. A função deve ter a assinatura abaixo:

```
bool removerDaPos(ITEM *item, int i, LISTA *l);
```

3) Implementar uma função **clonar** para criar uma lista nova contendo uma cópia dos itens da lista passada como parâmetro. A função deve ter a assinatura abaixo:

```
LISTA * clonar(LISTA *l);
```

4) Baseando-se nas funções disponíveis na Lista Encadeada, implemente uma **Lista Encadeada Ordenada**, isto é, que considera o valor dos itens como critério para manter os valores na ordem crescente, sendo o menor valor na cabeça da lista. Aproveite e otimize funções de busca e remoção para interromperem a operação quando chegarem a um valor maior do que o que se deseja buscar ou remover.