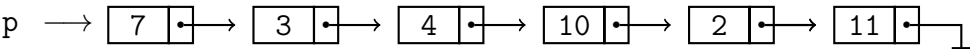


Imagine que o ponteiro **p** aponta para uma lista encadeada

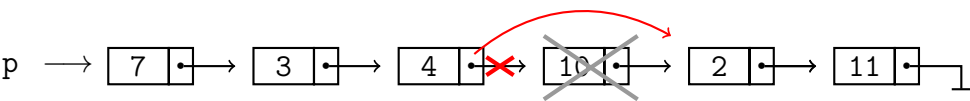


A tarefa é

→ *Remover o elemento k da lista*

Bom, aqui a gente precisa ter algum cuidado.

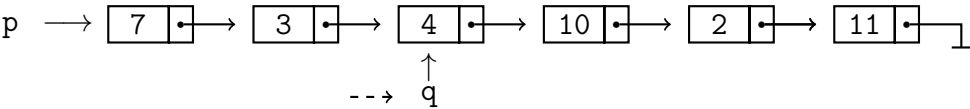
Abaixo nós temos o que precisa ser feito para remover o 10 da lista



Quer dizer, para remover o 10 é preciso modificar o registro do elemento que vem antes do 10.

E para modificar esse registro é preciso ter um ponteiro apontado para ele.

Então, para fazer a remoção do elemento **k**, a gente precisa levar um ponteiro auxiliar até o registro que fica antes do **k**



E daí, a gente realiza a operação de remoção.

A coisa fica assim

```
func opRemoção ( k, p )

    Se ( p == Nulo )      Retorna (p)                // o k não está lá ...

    Se ( p->num == k )    // o k é o primeiro elemento
        r = p          // segura ...
        p = p->prox     //
        free (r)        // e libera o registro do k
        Retorna (p)     //

    q = p;   achei = 0

    Enquanto (q->prox != Nulo )    // procura o k
        Se ( q->num == k )
            achei = 1;   Quebra
            q = q->prox

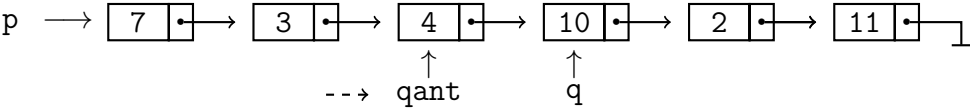
    Se ( achei == 0 )    // o k não está lá ...
        Retorna (p)

    Senão
        r = p->prox     // segura ...
        q->prox = r->prox
        free (r)        // e libera o registro do k
        Retorna (p)
```

Legal.

Mas, essa não é a única maneira de fazer as coisas.

Quer dizer, a gente também pode percorrer a lista com dois ponteiros, para chegar até o **k** e o elemento que fica atrás dele



E agora, é fácil fazer a remoção.

A coisa fica assim

```
func opRemoção2 ( k, p )

    Se ( p == Nulo )      Retorna (p)                // lista vazia

    q = p;   qant = Nulo

    Enquanto (q->prox != Nulo )    // procurando o k
        Se ( q->num == k )
            achei = 1;   Quebra
            q = q->prox

    Se ( q == Nulo )      Retorna (p)                // o k não está lá ...

    Senão
        Se ( qant == Nulo )    p = q->prox          // o k é o 1o elemento
        Senão                  qant->prox = q->prox

        free (q)

    Retorna (p)
```