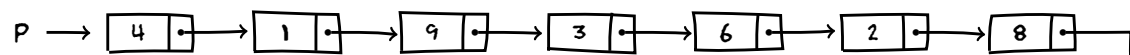


Removendo o elemento central

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



A tarefa é

- *remover o elemento central da lista*
- *(se houver algum elemento central ...)*

Quer dizer, se a lista tem uma quantidade ímpar de elementos, então existe um elemento central e ele deve ser removido.

Caso contrário, não existe um elemento central e não é preciso fazer nada.

Certo.

A única dificuldade é localizar o elemento central — *(caso haja um)*

Nós vamos resolver essa dificuldade com a seguinte esperteza

- nós vamos contar os elementos na ida e na volta
- e aquele elemento que tiver o mesmo número na ida e na volta é o elemento central

Para fazer a contagem na ida nós vamos utilizar um parâmetro.

E para fazer a contagem na volta nós vamos utilizar o valor de retorno da função.

A coisa fica assim

```
func REC-Rec (q, cont1)                                     // Remove o Elemento Central

    Se ( q == Nulo )   Retorna ( Nulo, 0 )                 // inicia a contagem da volta

    Senão
        px, cont2 = REC-Rec (q->prox, cont1+1)
        Se ( cont1 == cont2 )                               // eu sou o centro!
            free (q)
            Retorna ( px, cont2+1, )
        Senão
            q->prox = px                                     // re-conexão
            Retorna ( q, cont2+1, )
```