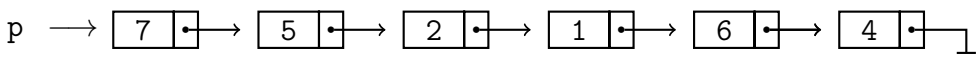


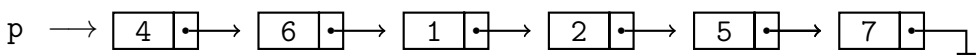
Imagine que o ponteiro **p** aponta para uma lista encadeada



A tarefa é

→ *inverter a ordem dos elementos da lista*

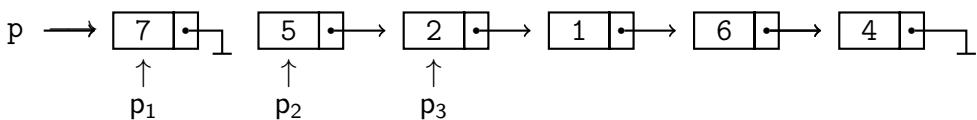
No exemplo acima, isso nos daria



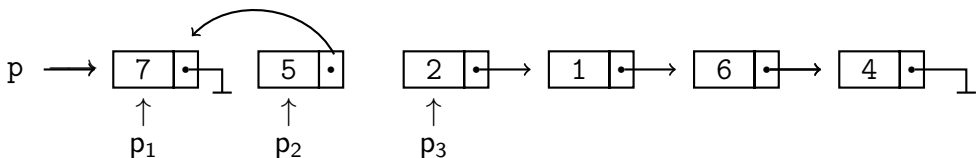
Mas, como é que a gente faz isso?

Bom, a nossa ideia é percorrer a lista com 3 ponteiros.

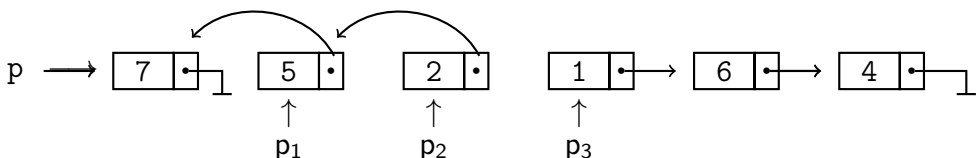
No início, o primeiro elemento aponta para `Nulo` — (*porque agora ele vai ser o último ...*)



E daí, a cada passo, o elemento apontado por **p₂** é desconectado do elemento da frente e reconectado ao elemento de trás



Daí os ponteiros andam, e a gente inverte o próximo elemento



No final, basta apontar o ponteiro **p** para aquele que era o último elemento — (*porque agora ele é o primeiro*)

A coisa fica assim

```
func invertLista (p)

    Se ( p == Nulo OU p->prox == Nulo )    Retorna (p)

    p1 = p;    p2 = p1->prox;    p3 = p2->prox           // posiciona os ponteiros

    p1->prox = Nulo                                       // o primeiro vira o último

    Enquanto (p3 != Nulo )
        p2->prox = q->prox
        p1 = p2;    p2 = p3;    p3 = p3->prox

    p2->prox = p1                                         // o último aponta para penúltimo

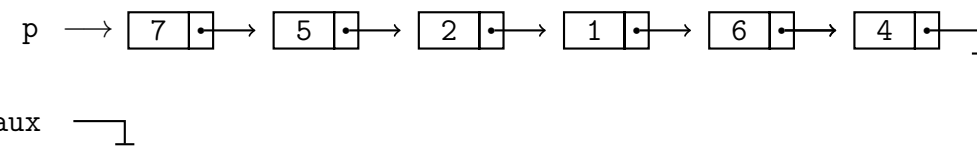
    p = p2                                                // e vira o primeiro elemento

    Retorna (p)
```

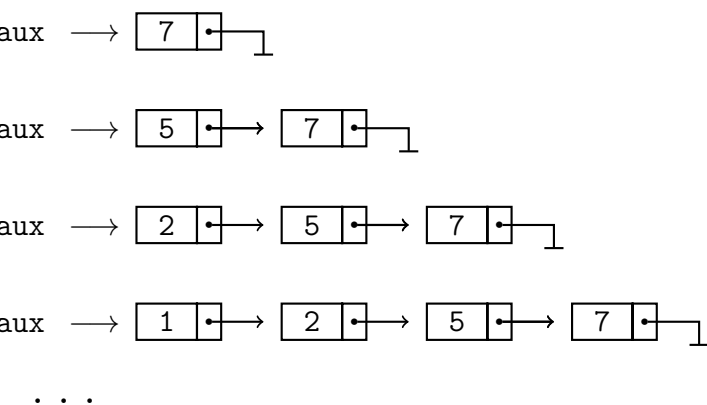
Mas, essa não é a única maneira de fazer as coisas.

Quer dizer, é possível fazer a coisa com apenas um ponteiro auxiliar.

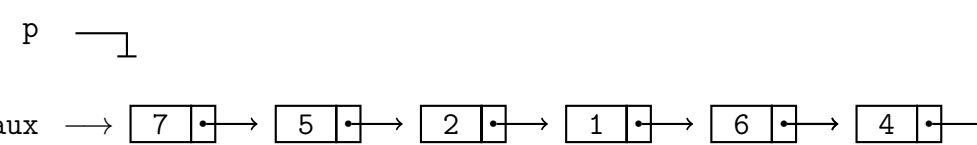
A ideia é que esse ponteiro começa apontando para `Nulo`



Daí, a gente vai sucessivamente removendo os elementos da lista **p**, e inserindo na lista **aux** — (*fazendo a inserção no início*)



Daí no final, a lista está completamente invertida



A coisa fica assim

```
func invertLista2 (p)

    Se ( p == Nulo OU p->prox == Nulo )    Retorna (p)

    aux = Nulo                               // a lista aux começa vazia

    Enquanto (p != Nulo )
        q = p;    p = p->prox               // tira da lista p
        q->prox = aux;    aux = q           // bota na lista aux

    Retorna (aux)
```