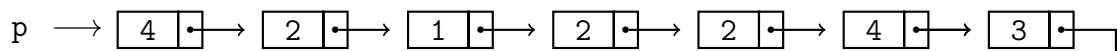


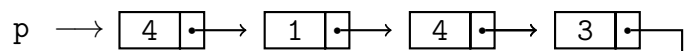
Imagine que o ponteiro **p** aponta para uma lista encadeada que pode conter elementos repetidos



A tarefa é

→ *remover todas as cópias do número k*

No exemplo acima, para $k = 2$, isso nos daria



Certo.

Aqui a gente vai utilizar outra vez a técnica de reconstruir a lista com o ponteiro **aux**.

Quer dizer, nós vamos remover os elementos de **p** um a um.

E daí,

- quando o elemento é diferente de **k**, ele é colocado na lista **aux**
- quando o elemento é igual a **k**, o seu registro é liberado

Só que dessa vez, para não inverter a ordem da lista, nós vamos fazer a inserção no final da lista **aux**.

E para fazer isso de maneira eficiente, nós vamos manter um ponteiro **ult** apontando para o último elemento de **aux**.

A coisa fica assim

```
func removeCópias ( k, p )

    Se ( p == Nulo )    Retorna (p)                                // já acabei ...

    aux = Nulo;    ult = Nulo

    Enquanto ( p != Nulo )

        q = p;    p = p->prox                                    // tira da lista p
        q->prox = Nulo
        Se ( q->num == k )                                        // Você é o kara?
            free (q)

        Senão                                                    // Não? então vai para a lista aux ...

            Se (aux == Nulo)
                aux = q;    ult = q                                // o primeiro a entrar em aux

            Senão
                ult->prox = q;    ult = q                        // e os próximos ...

    Retorna (aux)
```