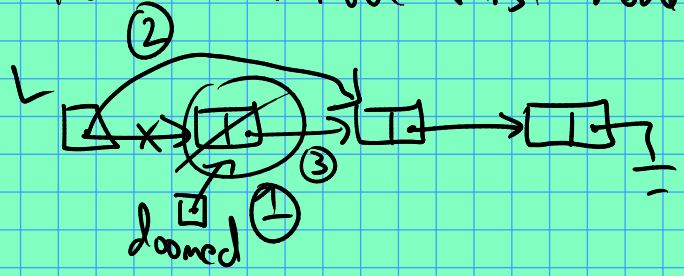
Exercise: clear a list (delete all nodes).

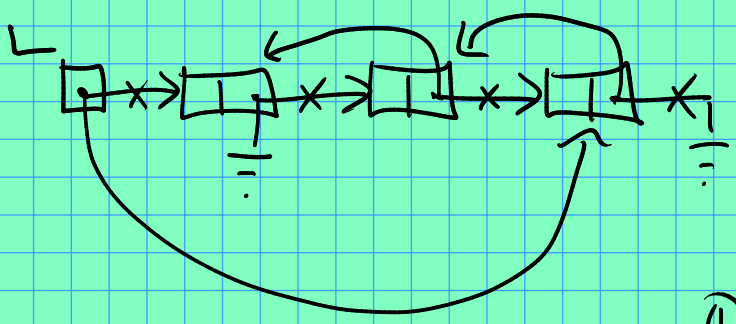
How to remove first node?




```
node * doomed = L;  // ①
L = L -> next;      // ②
delete doomed;      // ③
```

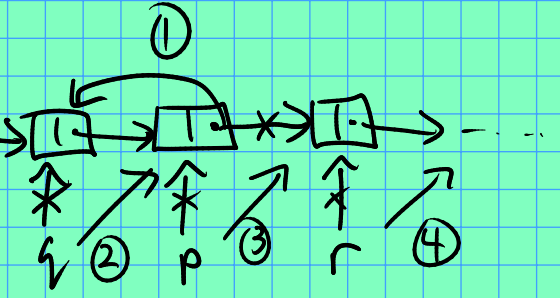
loop this to delete them all...

Exercise: reverse a list (without allocating any new nodes!)



Want to reverse all the arrows...

"Generic case":

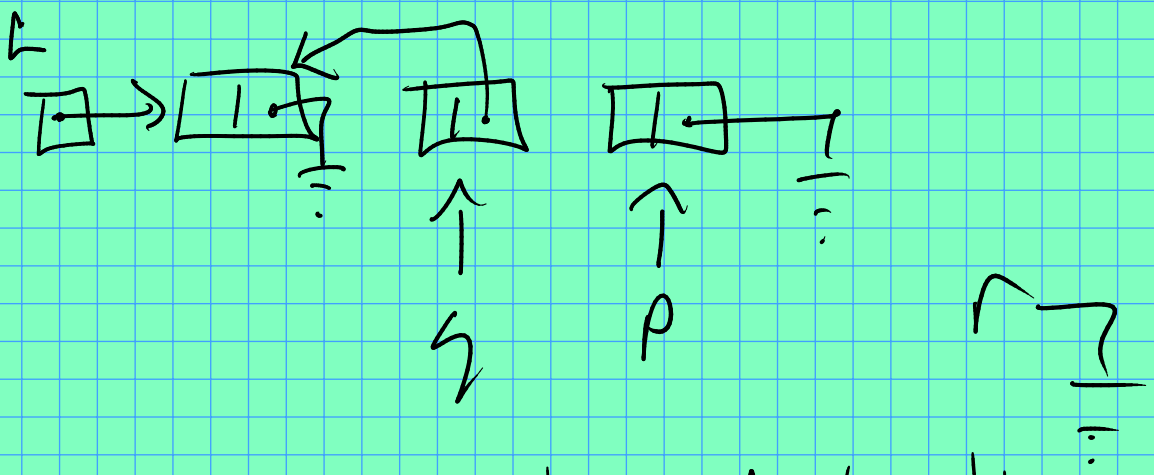


---

3 pointers should suffice. Can reverse p w/o losing rest of list...)

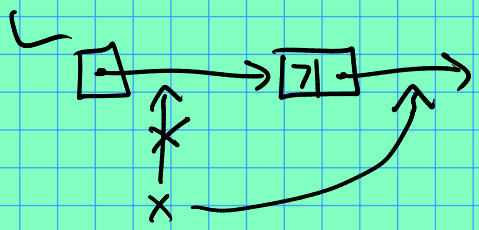
```
p -> next = q;  // ①
q = p;          // ②
p = r;          // ③
r = r -> next;  // ④
```

Trace on short list:



But our loop would end here!!
Last node not yet reversed.

Exercise: read list from stdin (in forward order) w/o a special case for the first node.



idea: store a pointer to the arrow where the next node should be attached.

```
node ** x = &L;
*x = new node;      // allocate new node; attach to end
(*x)->data = 7;     // set data
x = &(*x)->next;    // move x to end again.
```