③ (fill in data)

```
p -> next = new node;   // ①
p = p -> next;   // ②
p -> data = x;
p -> next = NULL;   // ③
```
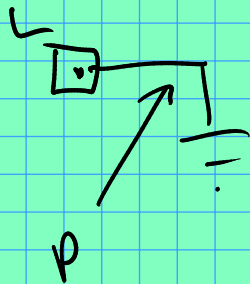
"invariant"
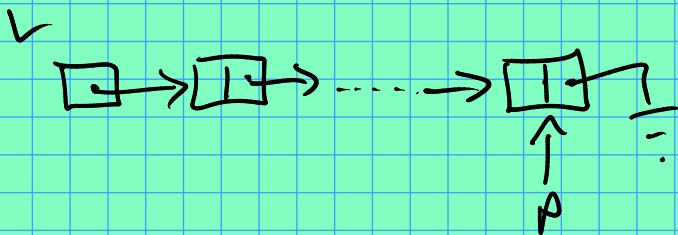
Issue: requires special case for first node, since L is not the .next do anything!
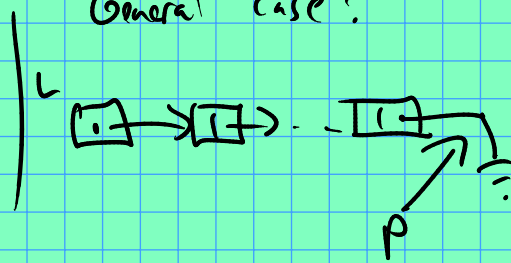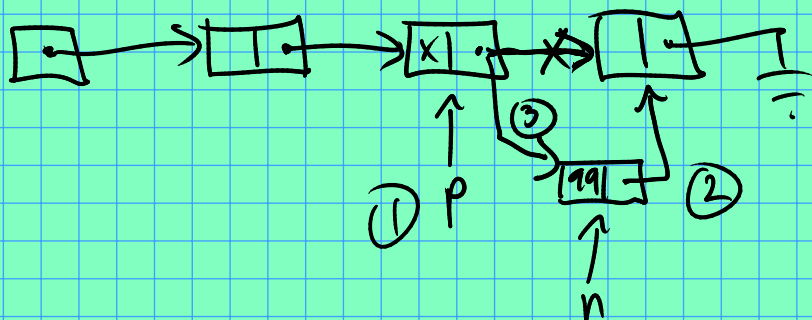
To remove special case:
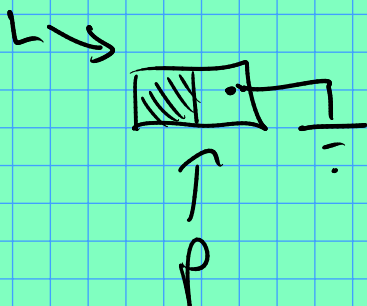
Alternative #2:

General case:


```
Node ** p = &L;
```
$*p \equiv$ next of very last node

```
*p = new node;
```

Exercise: search for an integer, and add a new node w/ data 99 after it.


```
// linear scan ....    //①

Node* n = new node;
n->data = 99;
n->next = p->next;   //②

p->next = n;   //③

//Note: this works even if p is
//pointing to the last node. (But does
                    require p ≠ NULL!)
```