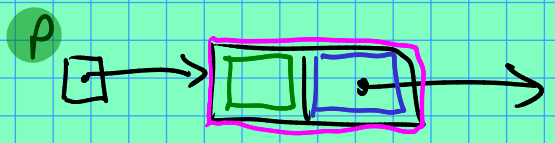
More about lists + dynamic memory.

Warm up: print a list.

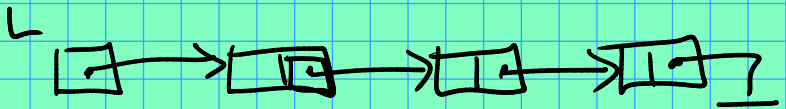



```
struct node {
    int data;
    node* next;
};
```

*p

(*p).data or p->data

(*p).next or p->next



L->data // first node's data

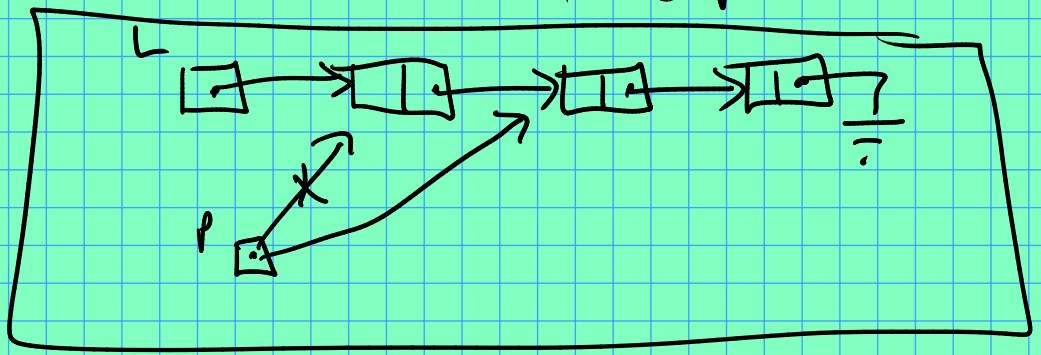
L->next->data // second node's data

≡ (L->next)->data

L->next->next->data // third node's data.

Gross :-

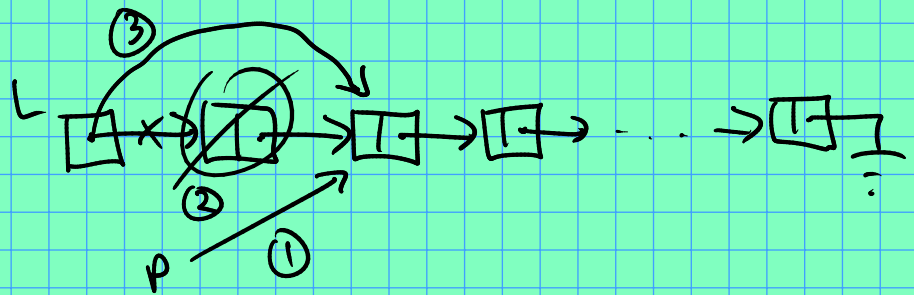
```
node* p = L; // pointer to current node
             // to be printed.
```




```
while (p != NULL) {
    cout << p->data << " ";
    p = p->next;
}
```

Exercise 2: clear a list (and free all the memory!!)

Warm up: how to remove first node?




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

// Note: make sure L not NULL for ①...

```
while (L != NULL) {
    node* p = L->next;  // ①
    delete L;  //②
    L = p;  //③
}
```

Finally, let's wrap this into a function:

```
void clearList (node*& L)
{
    // while loop goes here...

}
```