

# Insertion

- Consider the following linked list:

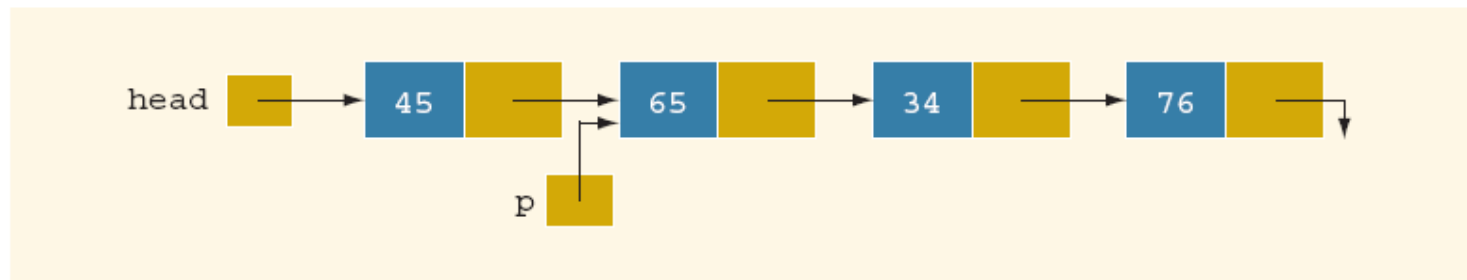


FIGURE 17-7 Linked list before item insertion

- A new node with `info` 50 is to be created and inserted after `p`

```
newNode = new nodeType; //create newNode
newNode->info = 50; //store 50 in the new node
```

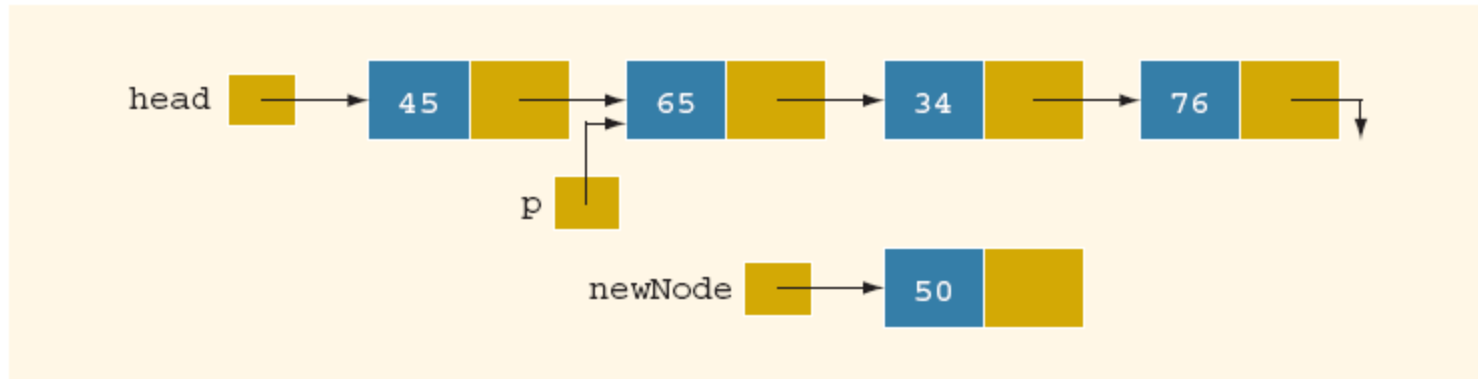


FIGURE 17-8 Create newNode and store 50 in it

```
newNode->link = p->link;
```

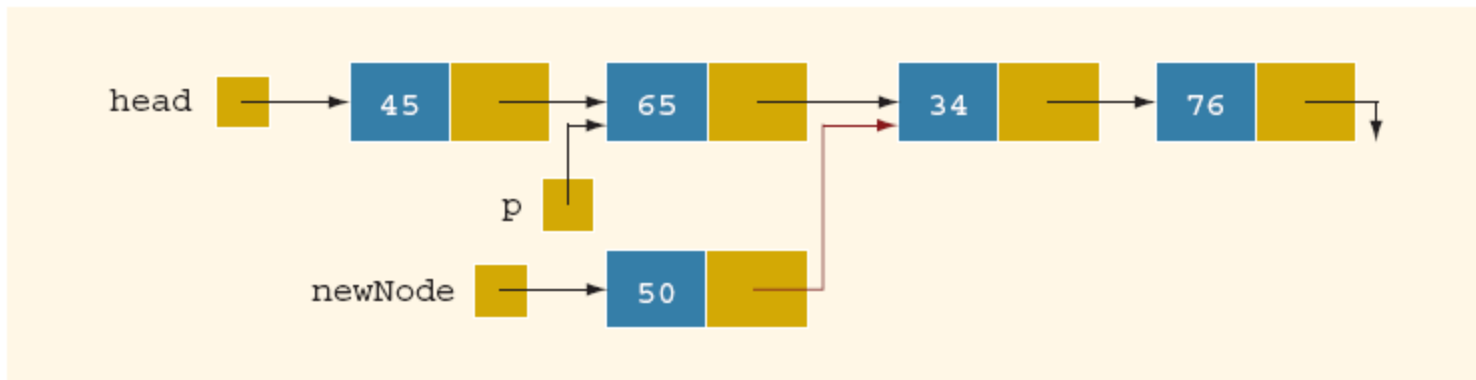


FIGURE 17-9 List after the statement `newNode->link = p->link;` executes

```
p->link = newNode;
```

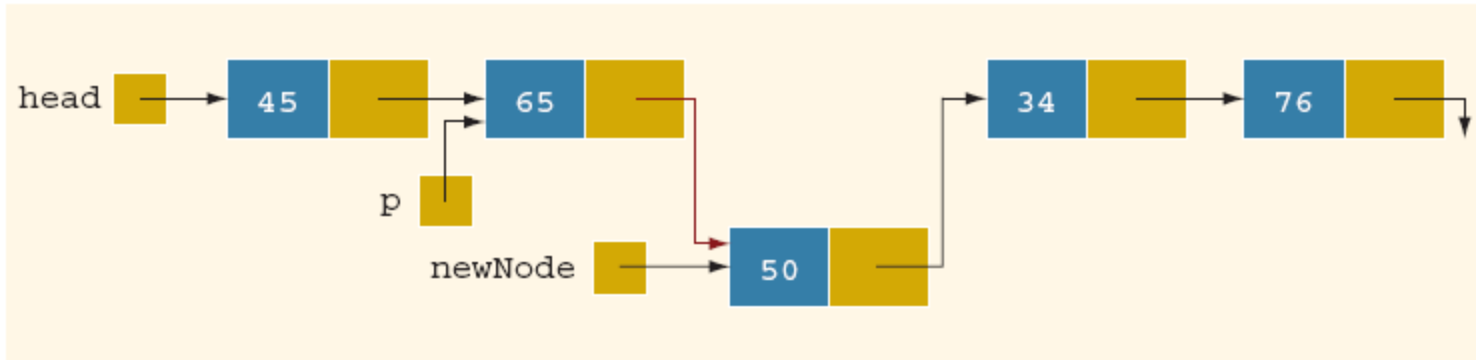


FIGURE 17-10 List after the statement `p->link = newNode` executes

The sequence of statements to insert the node is very important. Suppose that we reverse the sequence of the statements and execute the statements in the following order:

```
p->link = newNode;  
newNode->link = p->link;
```

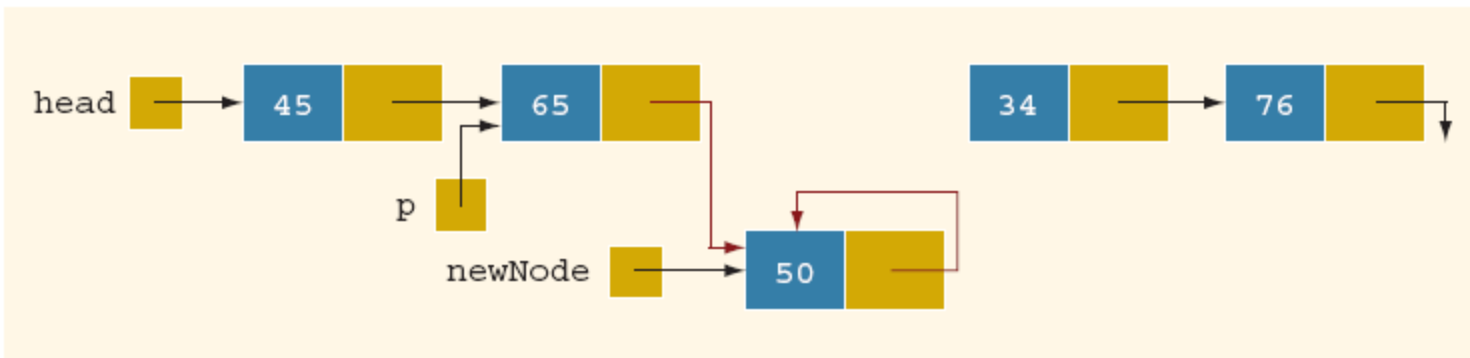
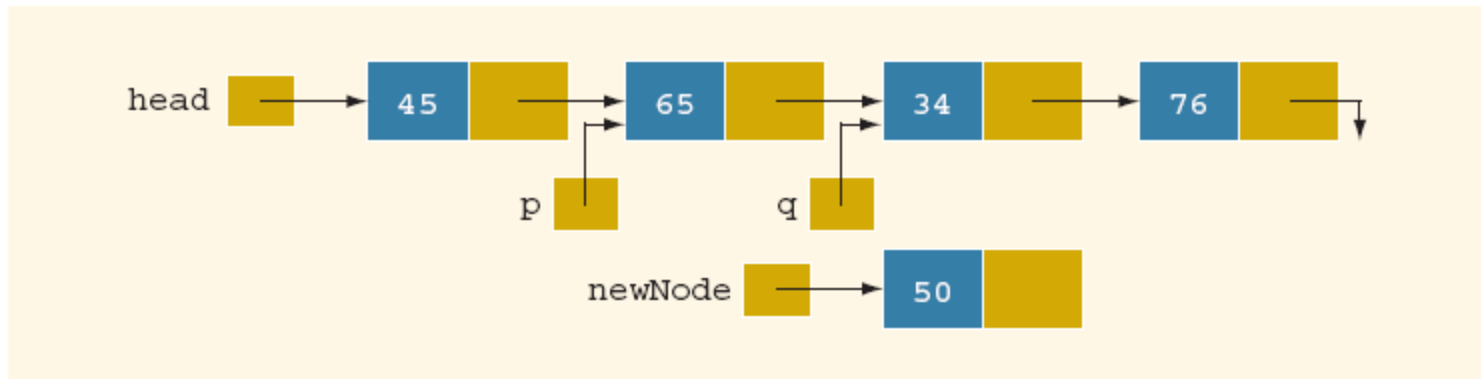


FIGURE 17-11 List after the execution of the statement `p->link = newNode;` followed by the execution of the statement `newNode->link = p->link;`

# Insertion (continued)

- Using two pointers, we can simplify the insertion code somewhat



- To **FIGURE 17-12** List with pointers *p* and *q*

```
newNode->link = q;  
p->link = newNode;
```

The order in which these statements execute does not matter

```
p->link = newNode;
```

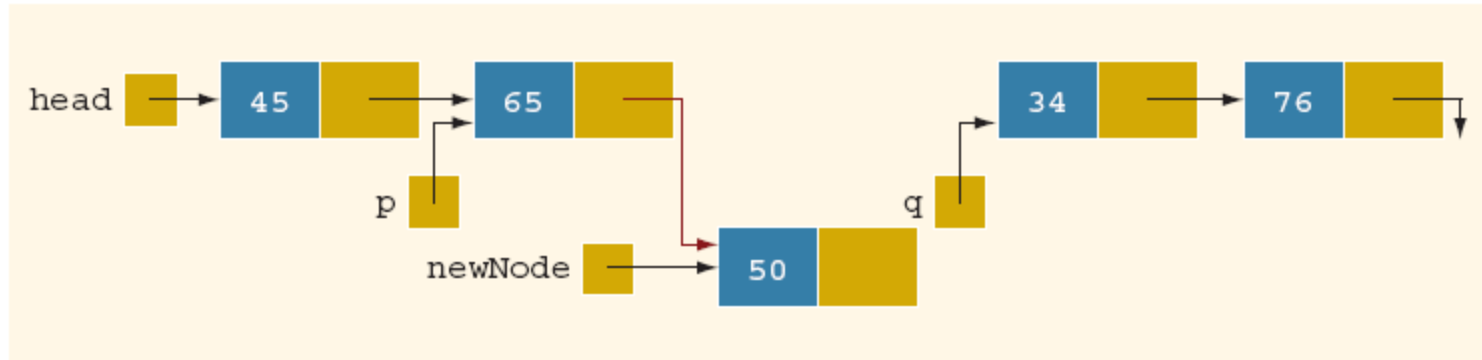


FIGURE 17-13 List after the statement `p->link = newNode;` executes

```
newNode->link = q;
```

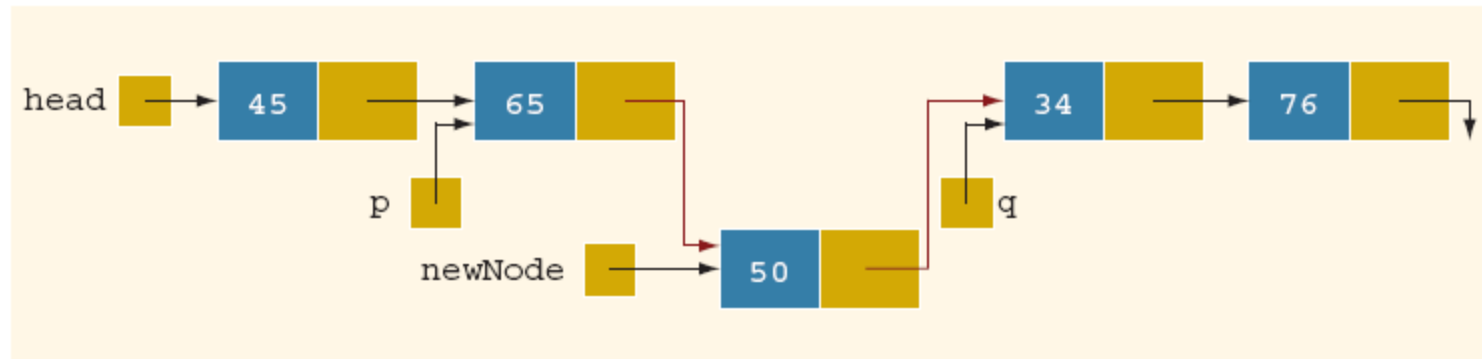


FIGURE 17-14 List after the statement `newNode->link = q;` executes

# Deletion

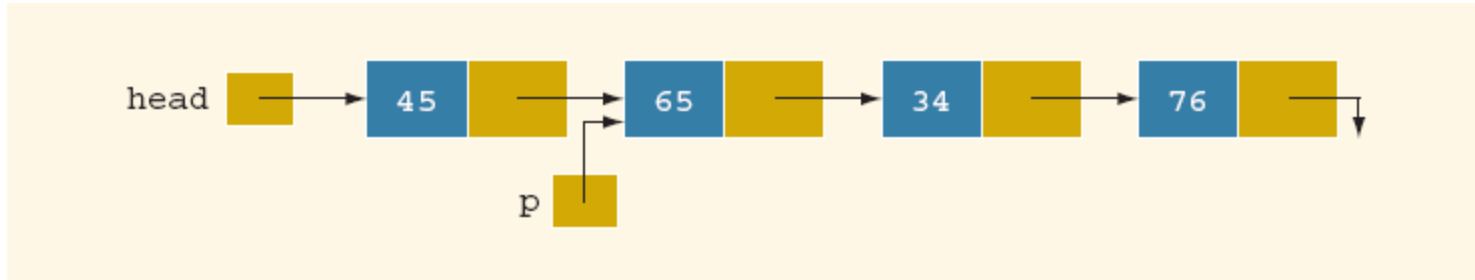


FIGURE 17-15 Node to be deleted is with info 34

```
p->link = p->link->link;
```

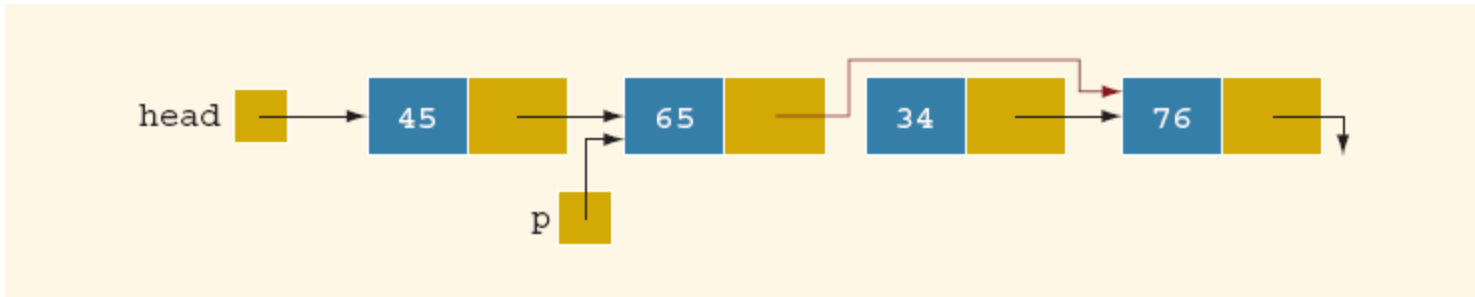


FIGURE 17-16 List after the statement `p->link = p->link->link;` executes

Node with `info 34` is removed from the list, but memory is still occupied; node is dangling

```
q = p->link;  
p->link = q->link;  
delete q;
```

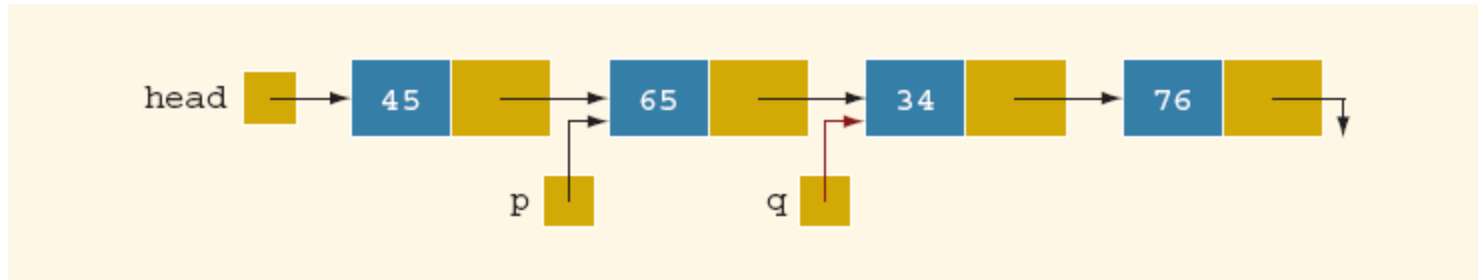


FIGURE 17-17 List after the statement `q = p->link;` executes

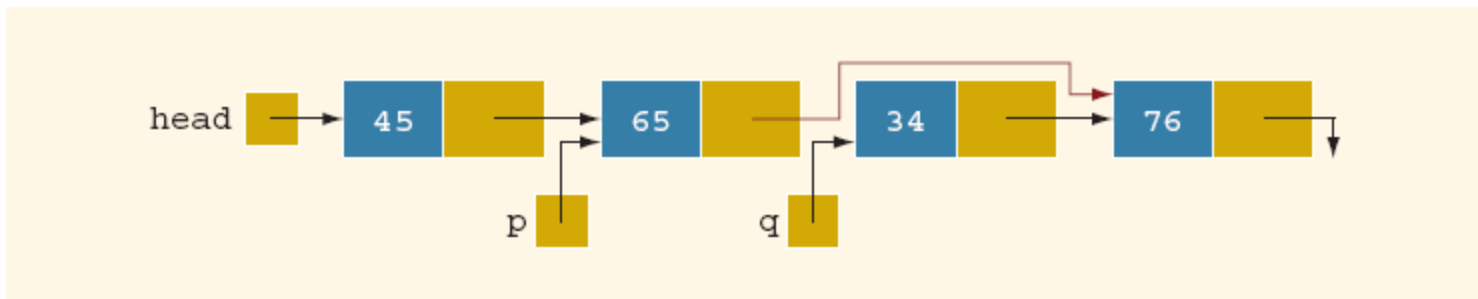


FIGURE 17-18 List after the statement `p->link = q->link;` executes

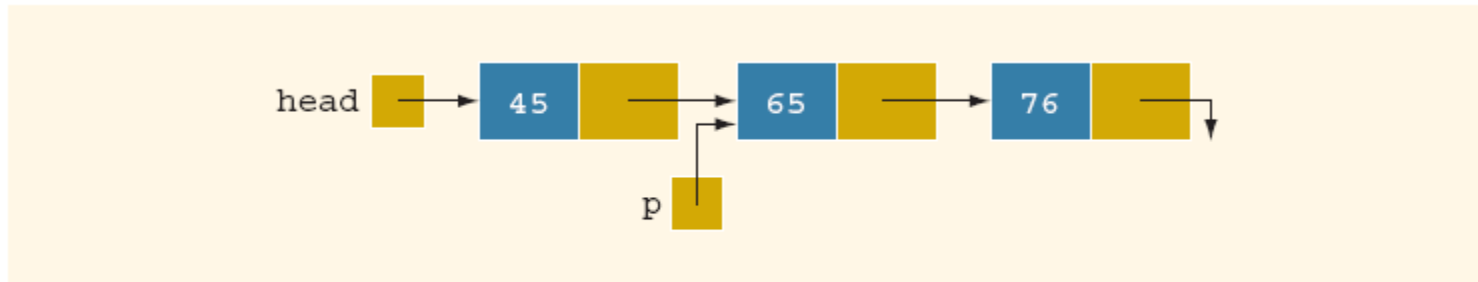


FIGURE 17-19 List after the statement `delete q;` executes