Linked Lists, continued

You cannot cycle through a linked list by using `p++` or `p--`.

To add a node into the middle of the list:

```
void AddItem(string newItem)
{
    if (our list is totally empty)
        //Just use our addToFront() method to add the new node
    else if (our new node belongs at the very top of the list)
        Just use our addToFront() method to add the new node
    else //new node belongs to somewhere in the middle of the list
    {
        //Use a traversal loop to find the node just ABOVE where you want to insert our new ite
m.
        //In a singly linked list, there's no way to go backward.
        //Allcoate and fill our new node with the item.
        //Link the new node into the list right after the ABOVE node.
    }
}
```

```
void AddItem(string newItem)
{
    if (head == nullptr //our list is totally empty)
        //Just use our addToFront() method to add the new node
        AddToFront(newItem);
    else if (//our new node belongs at the very top of the list)
        //Just use our addToFront() method to add the new node
        AddToFront(newItem);
    else //new node belongs to somewhere in the middle of the list
    {
```

```
        //Use a traversal loop to find the node just ABOVE where you want to insert our new ite
m.
        Node *p = head; //start with top node
        while(p->next != nullptr)
        {
            if(newItem >= p->value && newItem <= p->next->value
            /*p points just above where I want to insert*/)
                break;
            p = p->next; //move down one node
        }
        //the while loop takes us to the very last node
        //In a singly linked list, there's no way to go backward.
        //Allcoate and fill our new node with the item.
        Node *latest = new Node;
        latest->value = newItem;
        //Link the new node into the list right after the ABOVE node.
        latest->next = p->next; //link new node to the node below
        p->next = latest; //link above node to our new node
        //these two lines MUST be in this order


    }
}
```

Short-circuiting
If you have an if statement with multiple clauses, it'll run the leftmost clause first.