Linked list -- Sorted list (Ascending order)

```
void List::insert(ListItemType toAdd)
{
   nodePtr   prev, curr;
   nodePtr   newNode;

   // create new node
   newNode = new Node;
   assert(newNode);
   newNode->item = toAdd;

   prev=NULL;
   curr=head;
   while ((curr!=NULL)&&(curr->item < toAdd))
   {
        prev = curr;
        curr = curr->next;
   }

   // <case 1> insertion at the beginning of the list
   if (curr == head)
   {
       // add code here to perform insertion
       // at the head of the list
       newNode->next = head;
       head = newNode;
   }
   else // case2:insertion in the middle or end of list
   {
       // add code here
       newNode->next = curr;
       prev->next = newNode;
   }
}

void List::delete(ListIemType  toDelete)
{
      nodePtr curr, prev;

      if (head == NULL)
          cout << "The list is empty." << endl;
      else
      {
          prev= head;
          curr = head;
          while ((curr!=NULL) &&
                      (curr->item < toDelete))
          // can you switch the order of
```

```
        // the two conditions ??
        {
                prev= curr;
                curr = curr->next;
        }

        if  ((curr == head) &&
             (curr->item == toDelete))
        // delete from the head of the list
        {
                curr = head;
                head  = head->next;
                curr->next = NULL;
                delete curr;
                curr = NULL;

                size --;
        }
        else  if ((curr!=NULL)&&(curr != head) &&
                    (curr->item == toDelete))
        // found the node, prev points to
          // the node in front of "foundNode",
        // curr points to the "foundNode"
        {
                prev->next = curr->next;
                    // remove curr from the list
                curr->next = NULL;                      // delete the memory space
                delete curr;
                curr=NULL;

                size --;
        }
        else  // curr == NULL case
        {
                cout << toDelete << " is not in the list." << endl;
                cout << "Deletion operation not performed. " << endl;
        }
    }
}
```