

```

template <typename T>
void LinkedList<T>::insertOrder(const T& item)
{
    Node<T>* current;
    Node<T>* trailing=nullptr;
    bool found;
    Node<T>* another;
    another = new Node<T>;
    another->value = item;
    another->next = nullptr;
    if (first == nullptr) {
        first = another;
    }
    else {
        current = first;
        found = false;
        while (current != nullptr && !found) {
            if (current->value >= item) {
                found = true;
            }
            else {
                trailing = current;
                current = current->next;
            }
        }

        if (current == first) {
            another->next = first;
            first = another;
        }
        else {
            trailing->next = another;
            another->next = current;
        }
    }
}

```

```

template <typename T>
void LinkedList<T>::deleteNode(const T& item)
{
    Node<T>* current = first;
    Node<T>* previous = first;
    bool found = false;
    while (current != nullptr && !found) {
        if (current->value == item) {
            found = true;
        }
        else {
            current = current->next;
        }
    }
    if (found) {
        if (current == first) {
            first = current->next;
        }
        else {
            while (previous->next != current) {
                previous = previous->next;
            }
            previous->next = current->next;
        }
        delete current;
    }
}

```