

```
template <typename T>
void LinkedList<T>::insertOrder(const T& item)
{
    insertOrderRec(first, item);
}
```

```
template <typename T>
void LinkedList<T>::insertOrderRec(Node<T>*& listPtr,
                                   const T& item)
{
    if ((listPtr == nullptr) || (item < listPtr->value)) {
        Node<T>* tempPtr = listPtr;
        listPtr = new Node<T>;
        listPtr->value = item;
        listPtr->next = tempPtr;
    }
    else {
        insertOrderRec(listPtr->next, item);
    }
}
```

```

template <typename T>
void LinkedList<T>::deleteNode(const T& item)
{
    deleteNodeRec(first, item);
}

template <typename T>
void LinkedList<T>::deleteNodeRec(Node<T>*& listPtr,
                                const T& item)
{
    if (listPtr != nullptr) {
        if (item == listPtr->value) {
            Node<T>* tempPtr = listPtr;
            listPtr = listPtr->next;
            delete tempPtr;
        }
        else {
            deleteNodeRec(listPtr->next, item);
        }
    }
}

```