```cpp
#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* next = nullptr;;
};

Node* head = nullptr;

Node* newNode(int data) {
    Node* node = new Node;
    node->data = data;
    return node;
}


void insertAtHead(int data) {
    Node* node = newNode(data);
    if(head == nullptr) {
        head = node;
        return;
    }
    node->next = head;
    head = node;
}


void insertAtTail(int data) {
    if(head == nullptr) {
        insertAtHead(data);
        return;
    }
    Node* tail = head;
    while(tail != nullptr) {
           if (tail->next == nullptr) {
               Node* node = newNode(data);
               tail->next = node;
               break;
        }
           tail = tail->next;
    }
}


void insertAtMiddleAfter(int after, int data) {
    Node* prev = head;
    while(prev != nullptr) {
        if(prev->data == after) {
            Node* node = newNode(data);
            node->next = prev->next;
            prev->next = node;
            break;
        }
        prev = prev->next;
    }
}


void insertAtMiddleBefore(int before, int data) {
    if(head != nullptr && head->data == before) {
        insertAtHead(data);
        return;
    }

    Node* prev = head;
    while(prev != nullptr && prev->next != nullptr) {
        if(prev->next->data == before) {
            Node* node = newNode(data);
            node->next = prev->next;
            prev->next = node;
            break;
        }
        prev = prev->next;
    }
}

void deleteFromHead() {
    if(head == nullptr) {
        return;
    }

    Node* node = head;
    head = node->next;
    delete node;
}


void deleteFromTail() {
    if(head == nullptr) {
        return;
    }
    Node* prev = head;
    while(prev != nullptr && prev->next != nullptr) {
        if(prev->next->next == nullptr) {
            Node* node = prev->next;
            prev->next = nullptr;
            delete node;
            break;
        }
        prev = prev->next;
    }
}
```

```
107  void deleteFromMiddle(int data) {
108      if (head != nullptr && head->data == data) {
109          deleteFromHead();
110          return;
111      }
112      Node* prev = head;
113      while(prev != nullptr && prev->next != nullptr) {
114          if(prev->next->data == data) {
115              Node* node = prev->next;
116              prev->next = prev->next->next;
117              delete node;
118              break;
119          }
120          prev = prev->next;
121      }
122  }
123
124  void showForward() {
125      Node* node = head;
126      while(node != nullptr) {
127          cout<<" "<<node->data;
128          node = node->next;
129      }
130      cout<<endl;
131  }
132
133  int main() {
134
135      insertAtHead(10);
136
137      insertAtHead(20);
138
139      insertAtHead(30);
140
141      insertAtHead(40);
142
143      insertAtHead(50);
144
145      showForward();
146
147      insertAtMiddleAfter(30, 60);
148
149      showForward();
150
151
152      insertAtMiddleBefore(30, 70);
153
154      showForward();
155
156
157      insertAtTail(80);
158      showForward();
159
160      deleteFromHead();
161
162      showForward();
163
164      deleteFromTail();
165
166      showForward();
167
168      deleteFromMiddle(60);
169
170      showForward();
171
172      deleteFromMiddle(70);
173
174      showForward();
175
176
177      while(head != nullptr) {
178          deleteFromHead();
179          showForward();
180      }
181
182
183      return 0;
184  }
```