Design Linked List

```
class MyLinkedList {
    class Node {
    public:
        int val;
        Node* next;
        Node(int data, Node* nxt = NULL) {
            val = data;
            next = nxt;
        }
    };
    Node* head, * tail;
public:
    MyLinkedList() : head(nullptr), tail(nullptr) {
    }

    int get(int index) {
        int i = 0;
        Node* prev = head;
        while (i < index) {
            if (prev) {
                prev = prev->next;
            }
            else {
                break;
            }
            i++;
        }
        if (prev && i == index) {
            return prev->val;
        }
        else
            return -1;
    }

    void addAtHead(int val) {
        if (head == nullptr) {
            head = new Node(val);
            tail = head;
        }
        else {
            head = new Node(val, head);
        }
    }
```

```
void addAtTail(int val) {
    if (tail == nullptr) {
        addAtHead(val);
    }
    else {
        tail->next = new Node(val);
        tail = tail->next;
    }
}

void addAtIndex(int index, int val) {
    if (index == 0)
        addAtHead(val);
    else if (head) {
        int i = 0;
        Node* prev = head;
        Node* curr = head->next;

        while (i < index - 1) {
            if (curr) {
                prev = curr;
                curr = curr->next;
            }
            else {
                break;
            }
            i++;
        }
        if (prev && (i + 1) == index) {
            prev->next = new Node(val, curr);
            if (prev == tail) {
                tail = tail->next;
            }
        }

    }
}

void deleteAtIndex(int index) {
    Node* temp = nullptr;
    if (index == 0) {
        temp = head;
        head = head->next;
        delete temp;
    }
    else if (head){
        int i = 0;
```

```
            Node* prev = head;
            Node* curr = head->next;

            while (i < index - 1) {
                if (curr) {
                    prev = curr;
                    curr = curr->next;
                }
                else {
                    break;
                }
                i++;
            }
            if (curr) {
                prev->next = curr->next;
                if (curr == tail)
                    tail = prev;
                delete curr;
            }
        }
    }
};

/**
 * Your MyLinkedList object will be instantiated and called as such:
 * MyLinkedList* obj = new MyLinkedList();
 * int param_1 = obj->get(index);
 * obj->addAtHead(val);
 * obj->addAtTail(val);
 * obj->addAtIndex(index,val);
 * obj->deleteAtIndex(index);
 */
```