The given code is a program to create a linked list from a sequence of n integers.

```cpp
#include <bits/stdc++.h>
using namespace std;

/* Link list node */
class Node {
public:
        int key;
        Node* next;
};

void insert(Node** head_ref, int new_key);
void printList(Node* head);
void insertAt(int post, Node** head_ref, int new_key);
void remove(Node** head_ref, int key);
void readNnumbers(Node** head_ref);
void readNnumbers(Node** head_ref);
void processCommand(char cmd, Node** head_ref);

int main() {
        Node* head = NULL;
        readNnumbers(&head);

        char command;
        cin >> command;
        while (command != 'q'){
                processCommand(command, &head);
                cin >> command;
        }
        return 0;
}

void readNnumbers(Node** head_ref)
{
        int n, key;
        cin >> n;
        for (int i=0; i<n; i++){
                cin >> key;
                insert(head_ref, key);
        }
}

void insert(Node** head_ref, int new_key)
{
    /* allocate node */
    Node* new_node = new Node();

    /* put in the key */
    new_node->key = new_key;

    /* link the old list of the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref) = new_node;
}

void printList(Node* head)
{
        if (head == NULL) cout << "empty\n";

        while (head) {
                if (head->next == NULL)
                        cout << "[" << head->key << "] " << endl;
                else
                        cout << "[" << head->key << "] ";
                head = head->next;
        }
}

void processCommand(char cmd, Node** head_ref){
        if (cmd == 'i'){
                // *** Your code here!

        }
        else if (cmd == 'd'){
                // *** Your code here!
        }
}
```

```
void insertAt(int post, Node** head_ref, int new_key)
{
        // *** Your code here!

}

void remove(Node** head_ref, int key)
{
        // *** Your code here!

}
```

The program starts with reading N integers and creating a linked list of these N integers. Then it read commands to process the linked list until command "q" (for quitting the program) is found.

There are 3 other command codes "i" and "d" for insert and remove, respectively.

1) Command "i p k" is to call the function insertAt(p, arr, k), where p and k are 2 integers.
2) Command "d k" is to call the function remove(arr, k), where k is an integer.

You must implement 3 functions, i.e., processCommand(), insertAt() and remove() as follows.

1) Function processCommand(cmd) is to process command code "i" and "d".
2) Function insertAt(pos, ll, key) to add a new node to a linked list at pos^{th} node. For example, insertAt(1, ll, 1) must add a new node at 1st node. Insert(2, ll, 10) must add a new node at 10th node. However, if pos is higher than the current size of the linked list, the new node must be added to the end of the list. The listed list must be printed after the insertion.
3) Function remove(ll, key) to delete a node with key.
   a. If the key is found, that node must be deleted and "[key] deleted" must be printed.
   b. If key is not in the linked list, "[key] not existed" is printed.
   c. If the linked list is empty, "empty" is printed.

   Finally, the linked list must be printed after the deletion.

Your task is to implement functions insertAt(), remove(), and part of main() in the given code.

Input
Line 1 :   n   // an integer n, 0 < n < 20
Line 2 :   n integers with a space separated
Line 3-last line: commands

Output
Each line corresponds to a command in the input.

Example

Input	Output
3 1 2 3 i 1 4 i 10 5 q	[4] [3] [2] [1] [4] [3] [2] [1] [5]
3 1 2 3 d 2 d 2 d 3 d 1 q	2 deleted [3] [1] 2 not existed [3] [1] 3 deleted [1] 1 deleted empty
3 1 2 3 i 1 4 i 10 5 d 3 q	[4] [3] [2] [1] [4] [3] [2] [1] [5] 3 deleted [4] [2] [1] [5]