10) Linked list

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
    Node(int value) : data(value), next(NULL) {}
};

class LinkedList {
private:
    Node* head;

public:
    LinkedList() : head(NULL) {}

    ~LinkedList() {
        Node* current = head;
        while (current != NULL) {
            Node* nextNode = current->next;
            delete current;
            current = nextNode;
        }
        head = NULL;
    }

    void insertAtHead(int data) {
        Node* new Node = new Node(data);
        newNode->next = head;
        head = newNode;
        Node* current = head;
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return;
    }
    while (current != NULL)
    {
        cout << current->data << "->";
        current = current->next;
    }
    cout << "NULL" << endl;
}

void deleteNode(int val)
```

```
{
    if (head == NULL)
    {
        cout << "List is empty, cannot delete." << endl;
        return;
    }

    if (head->data == val)
    {
        Node* temp = head;
        head = head->next;
        delete temp;
        cout << "Node with value "
        << val << " deleted" << endl;
                return;
    }

    Node* current = head;
    Node* prev = NULL;

    while (current != NULL && current->data != val)
    {
        prev = current;
        current = current->next;
    }

    if (current == NULL)
    {
        cout << "Node with value " << val << " not found" << endl;
    }
    else
    {
        prev->next = current->next;
        delete current;
        cout << "Node with value " << val << " deleted" << endl;
    }
}

int main()
{
    LinkedList mylist;
    mylist.insertAtHead(30);
    mylist.insertAtHead(20);
    mylost.inseerAtHeas(10);
    cout << "Linked list after insertion:";
myList.printList ();
myList.delete Node (20);
cout << "Linked list after deleting 20:";
```

```
myList.printList ();
myList.delete Node (10);
cout << "Linked list after deleting 10:";
myList.printList ();
myList.delete Node (40);
myList.delete Node (30);
cout << "Linked list after deleting 30:";
myList.printList ();
return 0;
}
```