Singly Linked list

#include<iostream>

using namespace std;

struct Node{

    int data;

    Node \*next;

};

void insertatfirst(Node\* &head,int data){

    Node\* newNode=new Node();

    newNode->data=data;

    newNode->next=head;

    head=newNode;

}

void insertatspecified(Node\* &head,int target,int data){

    Node\* current=head;

    while(current!=NULL&&current->data!=target){

        current=current->next;

    }

    if(current==NULL){

        cout<<"not found target"<<endl;

        return;

    }

    Node\* newNode=new Node();

    newNode->data=data;

    newNode->next=current->next;

    current->next=newNode;

}

void insertatend(Node\* &head,int data){

    Node\* newNode=new Node();

    newNode->data=data;

    newNode->next=NULL;

    if(head==NULL){

        head=newNode;

        return;

    }

    Node\* current = head;

    while(current->next!=NULL){

        current=current->next;

    }

    current->next=newNode;

}

void deleteatfirst(Node\* &head){

    if(head==NULL){

        cout<<"list is empty";

        return;

    }

    Node\* temp = head;

    head= head->next;

    delete temp;

}

void deleteatspcified(Node\* &head,int target)

{

    if(head==NULL){

        cout<<"EMPty list";

        return;

    }

    Node\* current = head;

    Node\* prev=NULL;

    while(current!=NULL&&current->data!=target){

        prev=current;

        current=current->next;

    }

    if(current==NULL){

        cout<<"NOT FOUND";

        return;

    }

    prev->next=current->next;

    delete current;

}

void deleteatend(Node\* &head){

    if(head==NULL){

        cout<<"Empty list";

        return;

    }

    Node\* current=head;

    while(current->next->next!=NULL){

        current=current->next;

    }

    Node\* temp = current->next;

    delete temp;

    current->next=NULL;

}

// reverse the list

void reverselist(Node\* &head){

    Node\* prev = NULL;

    Node\* current = head;

    Node\* next =NULL;

    while(current!=NULL){

        next=current->next;

        current->next=prev;

        prev=current;

        current=next;

    }

    head=prev;

}

void  display(Node\*& head){

    Node\* current=head;

    while(current!=NULL){

        cout<<current->data<<" ";

        current=current->next;

    }

    cout<<endl;

}

int main(){

     Node\* head=NULL;

       insertatfirst(head,10);

       insertatfirst(head,20);

       insertatfirst(head,30);

       insertatfirst(head,40);

       insertatfirst(head,50);

       display(head);

       insertatspecified(head,30,25);

       display(head);

       insertatspecified(head,36,40);

       display(head);

       insertatend(head,60);

       display(head);

       insertatend(head,70);

       display(head);

       insertatend(head,800);

       display(head);

       deleteatfirst(head);

       display(head);

       deleteatspcified(head,60);

       display(head);

       deleteatend(head);

       display(head);

       reverselist(head);

       display(head);

    return 0;

}

Output;

PS C:\Users\gnave\OneDrive\Desktop\codes\output> cd 'c:\Users\gnave\OneDrive\Desktop\codes\output'

PS C:\Users\gnave\OneDrive\Desktop\codes\output> & .\'linkedlist.exe'

50 40 30 20 10

50 40 30 25 20 10

not found target

50 40 30 25 20 10

50 40 30 25 20 10 60

50 40 30 25 20 10 60 70

50 40 30 25 20 10 60 70 800

40 30 25 20 10 60 70 800

40 30 25 20 10 70 800

40 30 25 20 10 70

70 10 20 25 30 40