#include <iostream>

using namespace std;

class Node

{

public:

int data;

Node \*next;

Node(int d)

{

data=d;

next=NULL;

}

};

int length(Node \*head)

{

int len=0;

while(head!=NULL)

{

head=head->next;

len++;

}

return len;

}

void insertattail(Node \*&head,int data)

{

if(head==NULL)

{

head=new Node(data);

return;

}

Node \*tail=head;

while(tail->next!=NULL)

{

tail=tail->next;

}

Node \*n=new Node(data);

tail->next=n;

return;

}

void insertathead(Node \*&head,int data)

{

Node \*n=new Node(data);

n->next=head;

head=n;

}

void insertatmiddle(Node \*&head,int data,int pos)

{

if(head==NULL || pos==0){

insertathead(head,data);

}

else if(pos>length(head)){

insertattail(head,data);

}

else{

//Insert in the middle

//take pos-1 jumps

int jump=0;

Node \*temp=head;

while(jump<pos-1)

{

temp=temp->next;

jump++;

}

Node \*n=new Node(data);

n->next=temp->next;

temp->next=n;

}

}

void print(Node \*head)

{

while(head!=NULL)

{

cout<<head->data<<" ";

head=head->next;

}

}

void deletehead(Node \*&head,int pos)

{

//if nothing is present in the linkedlist

if(head==NULL)

{

return;

}

//delete head

if(pos==1)

{

Node \*temp=head;

head=head->next;

delete temp; //Permanently delete this node

return;

}

}

void deletetail(Node \*&head,int pos)

{

// Delete at tail

if(pos==length(head))

{

Node \*prev=NULL;

Node \*temp=head;

while(temp->next!=NULL)

{

prev=temp;

temp=temp->next;

}

delete temp;

prev->next=NULL;

}

}

void deleteatpos(Node \*&head, int pos)

{

//Delete inbetween head and tail

Node \*temp=head;

int z=1;

while(z<pos-1)

{

temp=temp->next;

z++;

}

delete temp->next;

temp->next=temp->next->next;

}

int main()

{

Node \*head=NULL;

insertathead(head,10);

insertathead(head,9);

insertathead(head,8);

insertathead(head,6);

insertathead(head,5);

insertathead(head,4);

insertattail(head,11);

insertatmiddle(head,7,3);

print(head);

cout<<endl;

deletehead(head,1);

print(head);

cout<<endl;

deleteatpos(head,4);

print(head);

cout<<endl;

deletetail(head,6);

print(head);

}

Output-

4 5 6 7 8 9 10 11

5 6 7 8 9 10 11

5 6 7 9 10 11

5 6 7 9 10