#include <iostream>

using namespace std;

struct Node{

char value;

struct Node\* next;

struct Node\* prev;

};

class Doublylinkedlist{

public:

Node \*head, \*curr;

Doublylinkedlist(){

head = NULL;

curr = NULL;

}

void Insertathead(char a){

Node\* bh = newnode();

if(head==NULL){

head=bh;

curr=bh;

}

temp->next = head;

head->prev = bh;

curr->next = bh;

head = bh;

}

void Insertattail(char a){

Node\* bh = newnode();

if(head==NULL){

head=bh;

curr=bh;

}

curr->next=bh;

temp->prev = curr;

curr = bh;

}

bool searchdata(char a){

Node \*m = head;

while(m != NULL){

if(m->value == a){

return true;

}

else {

m = m->next;

}

}

return 0;

}

void deletedata(char a){

if(head==NULL)

return;

if(a==1){

head = head->next;

if(head->next==NULL)

curr=NULL;

else

head->next->prev = NULL;

return;

}

Node \*bh;

Node \*bh1;

int i = 1;

bh = head;

while((i < a) && bh->next !=NULL){

bh = bh->next;

i++;

}

if(i==a){

bh1 = bh->prev;

bh1->next = bh->next;

if(bh->next==NULL)

curr=bh1;

else

bh->next->prev = bh1;

}

else

cout<<"Not Exist"<<endl;

}

void printforward(){

cout<<"Print forward order";

Node \*listdata;

listdata = head;

while(1) {

if(head==NULL || curr==NULL) break;

cout<<listdata->value<<" ";

if(listdata==curr) break;

listdata = listdata->next;

}

}

void printbackward(){

cout<<"Print backward order";

Node \*listdata;

listdata

while(1) {

if(head==NULL || curr==NULL) break;

cout<<listdata->value<<" ";

if(listdata->prev==NULL) break;

listdata=listdata->prev;

}

}

};

int main(){

head = NULL;

Insertathead(t);

Insertathead(h);

Insertathead(k);

searchdata(t);

Insertathead(l);

Insertathead(m);

deletedata(l);

printforward();

printbackward()

}