

```

#include<bits/stdc++.h>
using namespace std;
class node{
public:
    int info;
    node *link;
};
void creation(node **head,int data)
{
    node *ptr=new node();
    ptr->info=data;
    ptr->link=NULL;
    node *temp=*head;
    if(( *head)==NULL)
    {
        *head=ptr;
    }
    else
    {
        while(temp->link!=NULL)
        {
            temp=temp->link;
        }
        temp->link=ptr;
    }
}
void first_deletion(node **head)
{
    node *temp=*head;
    *head=temp->link;
    delete temp;
}
void last_deletion(node **head){
    node *temp=*head;
    node *ptr=NULL;
    while(temp->link!=NULL)
    {
        ptr=temp;
        temp=temp->link;
    }
    ptr->link=NULL;
    delete temp;
}
void delete_position(node **head,int pos)
{
    node *temp=*head;
    node *ptr=new node;
    if(pos==1)
    {
        *head=temp->link;
        delete temp;
    }
    else
    {
        while(--pos)
        {
            ptr=temp;
            temp=temp->link;
        }
        ptr->link=temp->link;
        delete temp;
    }
}
void delete_fixedData(node **head,int val)

```

```

{
    node *ptr=new node();
    node *temp=*head;
    if(val==temp->info){
        *head=temp->link;
        delete temp;
    }
    else{
        while(temp->info!=val){
            ptr=temp;
            temp=temp->link;
        }
        ptr->link=temp->link;
        delete temp;
    }
}

int after_deletion(node **head,int val,int n)
{
    node *ptr=new node;
    ptr->link=NULL;
    node *temp=*head;
    while(temp!=NULL)
    {
        if(temp->info==val)
        {
            if(temp->link==NULL)
            {
                cout<<"IMPOSSIBLE"<<endl;
                return n;
            }
            else
            {
                ptr=temp;
                ptr->link=temp->link;
                temp=temp->link;
                delete temp;
                return n-1;
            }
        }
        else
        {
            temp=temp->link;
        }
    }
}

void print(node **head)
{
    node *temp=*head;
    while(temp!=NULL)
    {
        cout<<temp->info<<" ";
        temp=temp->link;
    }
    cout<<endl;
}

int main()
{
    node *head=new node;
    head=NULL;
    int n,data,i,pos,val;
    cin>>n;
    for(i=0;i<n;i++)
    {
        cin>>data;
        creation(&head,data);
    }
}

```

```
print(&head);
/*cout<<endl;
first_deletion(&head);
print(&head);
cout<<endl;
last_deletion(&head);
print(&head);
cout<<endl;
cin>>pos;
delete_position(&head,pos);
print(&head);
cout<<endl;
cin>>val;
delete_fixedData(&head,val);
print(&head);*/
cin>>val;
n=after_deletion(&head,val,n);
print(&head);
}
```