#include<iostream>

#include<string>

using namespace std;

int count;

struct node

{

int prn;

string name;

node\* next;

};

class linklist{

public:

node\* head;

node\* tail;

linklist()

{

head=NULL;

tail=head;

}

void insertion();

void deletion();

void display();

};

void linklist::insertion(){

node\* nnode;

char ch,ch1;

int choice;

do{

cout<<"1.Insert Member \n 2.insert Head \n 3.insert tail \n"<<endl;

cout<<"enter your choice:";

cin>>choice;

switch(choice){

case 1:

//insertion of intermediate node

do{

node\* nnode;

nnode= new node;

cout<<"Enter PRN:";

cin>>nnode->prn;

cout<<"Enter name:";

cin>>nnode->name;

nnode->next=NULL;

if(head==NULL && tail==NULL){

head=tail=nnode;

}

else{

tail->next=nnode;

tail=nnode;

}

count++;

cout<<"do you want to continue to insert another member(y/n):";

cin>>ch;

}while(ch=='y');

break;

case 2:

//Insertion of head node

nnode=new node;

cout<<"Enter PRN:";

cin>>nnode->prn;

cout<<"Enter name:";

cin>>nnode->name;

nnode->next=NULL;

if(head==NULL && tail==NULL){

head=tail=nnode;

}

else{

nnode->next=head;

head=nnode;

}

count++;

break;

case 3:

//insertion of tail

nnode=new node;

cout<<"Enter PRN:";

cin>>nnode->prn;

cout<<"Enter name:";

cin>>nnode->name;

nnode->next=NULL;

if(head==NULL && tail==NULL){

head=tail=nnode;

}

else{

tail->next=nnode;

tail=nnode;

}

count++;

break;

}

cout<<"do you want to continue with another insertion(y/n):";

cin>>ch1;

}while(ch1=='y');

}

void linklist::deletion(){

int choice,key;

char ch1;

node\* temp;

node\* temp1;

do{

cout<<"Enter what do you want to delete:"<<endl;

cout<<"1.Head\n2.Member\n3.Tail\n4.No Deletion"<<endl;

cout<<"enter your choice:";

cin>>choice;

switch(choice){

case 1:

//deletion of head

temp=head;

head=head->next;

if(head == NULL) {

tail = NULL;

}

delete temp;

count--;

break;

case 2:

//Deletion of member

cout<<"Enter the PRN number you want to delete:";

cin>>key;

temp=head;

if(temp->prn == key) {

head = temp->next;

if(head == NULL) {

tail = NULL;

}

delete temp;

count--;

break;

}

while((temp->next)->prn!=key){

temp=temp->next;

}

temp1=(temp->next)->next;

delete temp->next;

count--;

temp->next = temp1;

break;

case 3:

//deletion of tail

temp=head;

if(temp->next == NULL) {

head = tail = NULL;

delete temp;

count--;

break;

}

while(temp->next!=tail){

temp=temp->next;

}

temp1=tail;

tail=temp;

delete temp1;

count--;

tail->next=NULL;

break;

case 4:

//No Deletion

cout<<"Exiting deletion function."<<endl;

return;

}

if(choice != 4) {

cout<<"do you want to continue with another deletion(y/n):";

cin>>ch1;

}

}while(ch1=='y');

}

void linklist::display(){

node\* temp;

temp=head;

while(temp!=NULL){

cout<<"PRN no:"<<temp->prn<<" Name:"<<temp->name<<endl;

temp=temp->next;

}

}

linklist concatenate(linklist a,linklist b){

a.tail->next=b.head;

return a;

}

int main()

{

linklist l1,l2,l3;

cout<<"\n Enter 1st linked list:"<<endl;

l1.insertion();

cout<<"\n 1st linked list"<<endl;

l1.display();

cout<<"\n";

l1.deletion();

l1.display();

cout<<"\n enter 2nd linklist:"<<endl;

l2.insertion();

cout<<"\n 2nd linked list"<<endl;

l2.display();

cout<<"\n";

l2.deletion();

l2.display();

cout<<"\n concatenated link list:"<<endl;

concatenate(l1,l2);

l1.display();

cout<<"\n Total count:"<<count;

}