**Pinnacle club.**

#include <iostream>

#include<cstring>

using namespace std;

struct node{

int prn;

string name;

node \*next;

};

class pinnacle{

public:

node \*create(node \*head){

node \*h,\*p;

h=head;

p=head;

string n;

int cnt,roll;

cout<<endl<<"Enter the number of members you want to add(Except President and secretory):";

cin>>cnt;

for(int i=0;i<cnt;i++){

p->next=new node;

p=p->next;

cout<<endl<<"Enter Name of member:";

cin>>n;

p->name=n;

cout<<endl<<"Enter PRN of member:";

cin>>roll;

p->prn=roll;

p->next=NULL;

}

p->next=new node;

p=p->next;

cout<<endl<<"Enter name of secretory:";

cin>>n;

p->name=n;

cout<<endl<<"Enter PRN of secretory:";

cin>>roll;

p->prn=roll;

p->next=NULL;

return head;

}

void print(node \*head){

node \*p=head->next;

cout<<endl<<"President name:"<<head->name;

cout<<endl<<"President Rollno:"<<head->prn;

while(p->next!=NULL){

cout<<endl<<"Name:"<<p->name;

cout<<endl<<"PRN No:"<<p->prn;

p=p->next;

}

cout<<endl<<"Secretory name:"<<p->name;

cout<<endl<<"Secretory Rollno:"<<p->prn;

}

int counts(node \*head){

int cnt=0;

while(head!=NULL){

cnt++;

head=head->next;

}

return cnt;

}

node \*add(node \*head){

node \*h,\*p,\*q;

h=p=head;

char ch;

int loc,roll;

string n;

cout<<endl<<"Do you want to update president(y/n):";

cin>>ch;

if(ch=='y' || ch=='Y'){

cout<<endl<<"Enter name of the updated president:";

cin>>n;

h->name=n;

cout<<endl<<"Enter PRN of the updated president:";

cin>>roll;

h->prn=roll;

}

cout<<endl<<"Do you want to update secretory(y/n):";

cin>>ch;

if(ch=='y' || ch=='Y'){

while(p->next!=NULL){

p=p->next;

}

cout<<endl<<"Enter name of the updated secretory:";

cin>>n;

p->name=n;

cout<<endl<<"Enter PRN of the updated secretory:";

cin>>roll;

p->prn=roll;

}

else{

cout<<endl<<"Enter location of member:";

cin>>loc;

for(int i=0;i<loc-1;i++){

p=p->next;

}

q=new node;

cout<<endl<<"Enter name of new member:";

cin>>n;

q->name=n;

cout<<endl<<"Enter PRN of new member:";

cin>>roll;

q->prn=roll;

p->next=q;

q->next=p->next;

}

return h;

}

node \*removes(node \*head){

node \*h,\*p,\*q;

h=p=head;

char ch;

int loc,roll;

string n;

cout<<endl<<"Do you want to update president(y/n):";

cin>>ch;

if(ch=='y' || ch=='Y'){

cout<<endl<<"Enter name of the updated president:";

cin>>n;

h->name=n;

cout<<endl<<"Enter PRN of the updated president:";

cin>>roll;

h->prn=roll;

}

cout<<endl<<"Do you want to update secretory(y/n):";

cin>>ch;

if(ch=='y' || ch=='Y'){

while(p->next!=NULL){

p=p->next;

}

cout<<endl<<"Enter name of the updated secretory:";

cin>>n;

p->name=n;

cout<<endl<<"Enter PRN of the updated secretory:";

cin>>roll;

p->prn=roll;

}

else{

cout<<endl<<"Enter location of member:";

cin>>loc;

for(int i=0;i<loc-1;i++){

p=p->next;

}

p->next=q;

q->next=q->next;

}

return h;

}

void reverses(node \*h){

if(h!=NULL){

reverses(h->next);

cout<<endl<<"Name:"<<h->name<<"\nPRN No:"<<h->prn;

}

}

node \*concat(node \*h1,node \*h2){

node \*head=h1;

while(head->next!=NULL){

head=head->next;

}

head->next=h2;

return h1;

}

};

int main()

{

node \*head1,\*head2;

int rollno,ch;

pinnacle p1;

string n;

cout<<endl<<"\*\*\*\*\*\*\*\*\*A Division\*\*\*\*\*\*\*";

head1=new node;

cout<<endl<<"Enter name of president:";

cin>>n;

head1->name=n;

cout<<endl<<"Enter PRN of president:";

cin>>rollno;

head1->prn=rollno;

head1->next=NULL;

head1=p1.create(head1);

p1.print(head1);

cout<<endl<<"\*\*\*\*\*\*\*\*\*B Division\*\*\*\*\*\*\*";

head2=new node;

cout<<endl<<"Enter name of president:";

cin>>n;

head2->name=n;

cout<<endl<<"Enter PRN of president:";

cin>>rollno;

head2->prn=rollno;

head2->next=NULL;

head2=p1.create(head2);

p1.print(head2);

do{

cout<<endl<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PINNACLE CLUB\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<endl<<"1.Add members...\n2.Delete members..\n3.count members...\n4.Display members..\n5.Display in reverse...\n6.Merge two clubs...\n7.Exit...";

cout<<endl<<"Enter your choice:";

cin>>ch;

switch(ch){

case 1:

char div;

cout<<endl<<"Enter division:";

cin>>div;

if(div=='A' || div=='a'){

head1=p1.add(head1);

}else if(div=='B' || div=='b'){

head2=p1.add(head2);

}else{

cout<<endl<<"Wrong choice!!!";

}

break;

case 2:

char div4;

cout<<endl<<"Enter division:";

cin>>div4;

if(div4=='A' || div4=='a'){

head1=p1.removes(head1);

}else if(div4=='B' || div4=='b'){

head2=p1.removes(head2);

}else{

cout<<endl<<"Wrong choice!!!";

}

break;

case 3:

char div1;

cout<<endl<<"Enter division:";

cin>>div1;

if(div1=='A' || div1=='a'){

cout<<endl<<"Number of members in A division's pinnacle club:"<<(p1.counts(head1)-2);

}else if(div1=='B' || div1=='b'){

cout<<endl<<"Number of members in B division's pinnacle club:"<<(p1.counts(head2)-2);

}else{

cout<<endl<<"Wrong choice!!!";

}

break;

case 4:

char div2;

cout<<endl<<"Enter division:";

cin>>div2;

if(div2=='A' || div2=='a'){

p1.print(head1);

}else if(div2=='B' || div2=='b'){

p1.print(head2);

}else{

cout<<endl<<"Wrong choice!!!";

}

break;

case 5:

char div44;

cout<<endl<<"Enter division:";

cin>>div44;

if(div44=='A' || div44=='a'){

p1.reverses(head1);

}else if(div44=='B' || div44=='b'){

p1.reverses(head2);

}else{

cout<<endl<<"Wrong choice!!!";

}

break;

case 6:

head1=p1.concat(head1,head2);

break;

case 7:

break;

}

}while(ch!=7);

}