/\*

Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of Second, third and final year students of department can be granted membership on request. Similarly one may cancel the membership of club. The list is sorted on PRN of student. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club member‘s information using singly linked list. Store student PRN and Name. Write functions to

a) Add and delete the members as well as president or even secretary.

b) Compute total number of members of club

c) Display members

d) Display list in reverse order using recursion

e) Two linked lists exists for two divisions. Concatenate two lists.

\*/

#include<iostream>

#include<string.h>

using namespace std;

#define TRUE 1

#define FALSE 0

#define SIZE 20

struct node

{

int PRN;

char name[SIZE];

struct node \*next;

}\*head;

class sll

{

public:

sll();

struct node \*create();

void display(struct node \*);

void total();

void reverse(struct node \*);

struct node \*insert\_President();

void insert\_Member();

void insert\_Secretary();

node \*remove();

node \*concat(struct node \*,struct node \*);

~sll();

};

sll::sll()

{

head=NULL;

}

sll::~sll()

{

node \*temp,\*temp1;

temp=head->next;

delete head;

while(temp!=NULL)

{

temp1=temp->next;

delete temp;

temp=temp1;

}

}

node \*sll::create()

{

node \*temp=NULL,\*New;

int val,flag;

char n[SIZE];

char ans='y';

flag=TRUE;

do

{

cout<<"\nEnter the PRN of Student ";

cin>>val;

cout<<"\nEnter Name of student::";

cin>>n;

New=new node;

if(New==NULL)

cout<<"Unable to allocate memory\n";

New->PRN=val;

strcpy(New->name,n);

New->next=NULL;

if(flag==TRUE) //Executed only for the first time

{

head=New;

temp=head;

flag=FALSE;

}

else

{ //temp last keeps track of the most recently created node

temp->next=New;

temp=New;

}

cout<<"\nDo you want to enter more elements?(Y/N)";

cin>>ans;

}while(ans=='Y'||ans=='y');

return head;

}

void sll::display(node \*head)

{

struct node \*temp;

temp=head;

if(temp==NULL)

{

cout<<"\nThe list is empty::";

return;

}

while(temp!=NULL)

{

cout<<"["<<temp->PRN<<","<<temp->name<<"]";

temp=temp->next;

}

}

void sll::total()

{

node \*temp;

int count=0;

temp=head;

if(temp==NULL)

{

cout<<"\nThe list is empty\n";

return;

}

while(temp!=NULL)

{

count++;

temp=temp->next;

}

cout<<"\nTotal number of members in club are "<<count;

}

void sll::reverse(node \*head)

{

if(head==NULL)

return;

reverse(head->next);

cout<<"{"<<head->PRN<<","<<head->name<<"}";

}

node \*sll::remove()

{

node \*temp,\*prev;

int key;

prev=new node;

temp=head;

cout<<"\nEnter the PRN of the node you want to delete:";

cin>>key;

while(temp!=NULL)

{

if(temp->PRN==key) //traverse till required to delete

break;

prev=temp;

temp=temp->next;

}

if(temp==NULL)

cout<<"\nNode not found";

else

{

if(temp==head)

head=temp->next;

else

prev->next=temp->next;

delete temp;

cout<<"\nThe number is deleted\n";

}

return head;

}

void sll::insert\_Secretary()

{

node \*New,\*temp;

New=new node;

cout<<"\nEnter the PRN of the student::";

cin>>New->PRN;

cout<<"\nEnter the name of the Student::";

cin>>New->name;

if(head==NULL)

head=New;

else

{

temp=head;

while(temp->next!=NULL)

temp=temp->next;

temp->next=New;

New->next=NULL;

}

cout<<"\nThe member is inserted\n";

}

void sll::insert\_Member()

{

int key;

node \*temp,\*New;

New=new node;

cout<<"\nEnter the PRN o the student::";

cin>>New->PRN;

cout<<"\nEnter the name of the student::";

cin>>New->name;

if(head==NULL)

{

head=New;

}

else

{

cout<<"\nEnter the PRN after which you want to insert the node::";

cin>>key;

temp=head;

do

{

if(temp->PRN==key)

{

New->next=temp->next;

temp->next=New;

break;

}

else

temp=temp->next;

}while(temp!=NULL);

}

cout<<"\nThe member is inserted\n";

}

node \*sll::insert\_President()

{

node \*New,\*temp;

New=new node;

cout<<"\nEnter the PRN of the student::";

cin>>New->PRN;

cout<<"\nEnter the name of the student:";

cin>>New->name;

if(head==NULL)

head=New;

else

{

temp=head;

New->next=temp;

head=New;

}

cout<<"\nThe member is inserted \n";

return head;

}

node \*sll::concat(node \*head1,node \*head2)

{

node \*temp;

temp=head1;

while(temp->next!=NULL)

temp=temp->next;

temp->next=head2;

cout<<"\nThe list are concatenated ::";

return head1;

}

int main()

{

sll s;

int choice,ch1;

char ans='y';

node \*start=NULL;

node \*start1,\*start2;

start1=NULL;

start2=NULL;

do

{

cout<<"\n1.Create";

cout<<"2.Display Members";

cout<<"\n3.Insert Member";

cout<<"\n4.Delete Members";

cout<<"\n5.Total numbers of members of club";

cout<<"\n6.Display list in reverse using recursion";

cout<<"\n7.concatnate two lists";

cout<<"\n8.Quit";

cout<<"\nEnter your choice (1-8)";

cin>>choice;

switch(choice)

{

case 1:

start=s.create();

break;

case 2:

s.display(start);

break;

case 3:

cout<<"\nThe Members are \n";

s.display(start);

cout<<"\nMenu";

cout<<"\n1.Insert President";

cout<<"\n2.Insert Members:";

cout<<"\n3.Insert Secratry:";

cout<<"\nEnter your choice::";

cin>>ch1;

switch(ch1)

{

case 1:

start=s.insert\_President();

break;

case 2:

s.insert\_Member();

break;

case 3:

s.insert\_Secretary();

break;

default:

cout<<"\nInvalid Choice";

}

break;

case 4:

start=s.remove();

break;

case 5:

s.total();

break;

case 6:

s.reverse(start);

break;

case 7:

cout<<"\nEnter the data for first division\n";

start1=s.create();

cout<<"\nEnter the data for second division\n";

start2=s.create();

start=s.concat(start1,start2);

break;

default:

cout<<"Invalid choice::";

}

cout<<"\nWant to go t main menu?";

cin>>ans;

}while(ans=='y'||ans=='Y');

return 0;

}

/\*

OUTPUT :

1.Create2.Display Members

3.Insert Member

4.Delete Members

5.Total numbers of members of club

6.Display list in reverse using recursion

7.concatnate two lists

8.Quit

Enter your choice (1-8)1

Enter the PRN of Student 1

Enter Name of student::ravi

Do you want to enter more elements?(Y/N)y

Enter the PRN of Student 2

Enter Name of student::sanjay

Do you want to enter more elements?(Y/N)y

Enter the PRN of Student 3

Enter Name of student::raj

Do you want to enter more elements?(Y/N)n

Want to go t main menu?y

1.Create2.Display Members

3.Insert Member

4.Delete Members

5.Total numbers of members of club

6.Display list in reverse using recursion

7.concatnate two lists

8.Quit

Enter your choice (1-8)2

[1,ravi][2,sanjay][3,raj]

Want to go t main menu?y

1.Create2.Display Members

3.Insert Member

4.Delete Members

5.Total numbers of members of club

6.Display list in reverse using recursion

7.concatnate two lists

8.Quit

Enter your choice (1-8)6

{3,raj}{2,sanjay}{1,ravi}

Want to go t main menu?y

1.Create2.Display Members

3.Insert Member

4.Delete Members

5.Total numbers of members of club

6.Display list in reverse using recursion

7.concatnate two lists

8.Quit

Enter your choice (1-8)6

{3,raj}{2,sanjay}{1,ravi}

Want to go t main menu?

\*/