7 )Department of Computer Engineering has student's 1/8 club named 'Pinnacle Club'. Students of Second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. 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

***CODE :-***

#include <iostream>

#include <string>

using namespace std;

struct Member

{

string prn;

string name;

Member\* next;

};

class PinnacleClub

{

private:

Member\* head;

Member\* tail;

public:

PinnacleClub()

{

head = nullptr;

tail = nullptr;

}

~PinnacleClub()

{

Member\* temp;

while (head != nullptr)

{

temp = head;

head = head->next;

delete temp;

}

}

void addMember(string prn, string name)

{

Member\* newMember = new Member{prn, name, nullptr};

if (tail)

{

tail->next = newMember;

tail = newMember;

} else {

head = tail = newMember;

}

}

void addPresident(string prn, string name)

{

Member\* newMember = new Member{prn, name, head};

head = newMember;

if (!tail)

{

tail = newMember;

}

}

void addSecretary(string prn, string name)

{

addMember(prn, name);

}

void deleteMember(string prn)

{

if (!head) return;

if (head->prn == prn)

{

Member\* temp = head;

head = head->next;

delete temp;

if (!head) tail = nullptr;

return;

}

Member\* temp = head;

while (temp->next && temp->next->prn != prn)

{

temp = temp->next;

}

if (temp->next)

{

Member\* toDelete = temp->next;

temp->next = temp->next->next;

if (toDelete == tail) tail = temp;

delete toDelete;

}

}

int totalMembers() const

{

int count = 0;

Member\* temp = head;

while (temp)

{

count++;

temp = temp->next;

}

return count;

}

void displayMembers() const

{

Member\* temp = head;

while (temp)

{

cout << "PRN: " << temp->prn << ", Name: " << temp->name << endl;

temp = temp->next;

}

}

void displayReverse(Member\* temp) const

{

if (temp == nullptr) return;

displayReverse(temp->next);

cout << "PRN: " << temp->prn << ", Name: " << temp->name << endl;

}

void displayReverseOrder() const

{

cout << "Members in Reverse Order:" << endl;

displayReverse(head);

}

void concatenate(PinnacleClub& otherClub)

{

if (!head)

{

head = otherClub.head;

tail = otherClub.tail;

} else if (otherClub.head)

{

tail->next = otherClub.head;

tail = otherClub.tail;

}

otherClub.head = nullptr;

otherClub.tail = nullptr;

}

};

int main()

{

PinnacleClub club1;

PinnacleClub club2;

club1.addPresident("P001", "John Doe");

club1.addMember("P002", "Alice Smith");

club1.addMember("P003", "Bob Johnson");

club1.addSecretary("P004", "Eve Davis");

club2.addPresident("P005", "Tom White");

club2.addMember("P006", "Lucy Black");

club2.addSecretary("P007", "Mia Green");

cout << "Club 1 Members:" << endl;

club1.displayMembers();

cout << "Total Members in Club 1: " << club1.totalMembers() << endl;

cout << "\nClub 2 Members:" << endl;

club2.displayMembers();

cout << "Total Members in Club 2: " << club2.totalMembers() << endl;

club1.concatenate(club2);

cout << "\nAfter concatenating Club 2 into Club 1:" << endl;

club1.displayMembers();

club1.displayReverseOrder();

club1.deleteMember("P003"); // Delete a middle member

club1.deleteMember("P004"); // Delete the secretary

club1.deleteMember("P001"); // Delete the president

cout << "\nAfter deleting some members:" << endl;

club1.displayMembers();

cout << "Total Members in Club 1: " << club1.totalMembers() << endl;

return 0;

}

***OUTUT :-***

