# Practical No 7

**Title: -** Write C++ program to maintain club member‘sinformation using singly linked list.

**Name:-** Sattyam Sagar Chavan

**Roll No:-** 73 **Class:-**SE AIDS **Sub:-**DSL

# INPUT:

#include <iostream> #include <string>

using namespace std; struct Node {

string prn; string name; Node\* next;

Node(string p, string n) : prn(p), name(n), next(nullptr) {}

};

class PinnacleClub { private:

Node\* head; Node\* tail; int count;

public:

PinnacleClub() : head(nullptr), tail(nullptr), count(0) {}

void addPresident(string prn, string name) { if (head != nullptr) {

cout << "President already exists." << endl; return;

}

head = new Node(prn, name); tail = head;

count = 1;

}

void addSecretary(string prn, string name) { if (tail != nullptr) {

cout << "Secretary already exists." << endl; return;

}

Node\* newNode = new Node(prn, name); if (head == nullptr) {

head = newNode; tail = head;

} else {

tail->next = newNode; tail = newNode;

}

count++;

}

void addMember(string prn, string name) { Node\* newNode = new Node(prn, name); if (head == nullptr) {

cout << "Add president first." << endl; delete newNode;

return;

}

if (tail == nullptr) { tail = newNode; head->next = tail;

} else {

tail->next = newNode; tail = tail->next;

}

count++;

}

void removeMember(string prn) { if (head == nullptr) return;

Node\* temp = head;

Node\* prev = nullptr;

if (head->prn == prn) { temp = head;

head = head->next; delete temp;

count--;

if (head == nullptr) tail = nullptr; return;

}

while (temp != nullptr && temp->prn != prn) { prev = temp;

temp = temp->next;

}

if (temp == nullptr) return;

if (temp == tail) { tail = prev;

}

prev->next = temp->next; delete temp;

count--;

}

void removePresident() {

if (head == nullptr) return;

Node\* temp = head;

head = head->next; delete temp;

count--;

if (head == nullptr) tail = nullptr;

}

void removeSecretary() {

if (head == nullptr) return;

Node\* temp = head;

Node\* prev = nullptr;

while (temp != nullptr && temp->next != nullptr) { prev = temp;

temp = temp->next;

}

if (temp == nullptr) return; if (prev != nullptr) {

prev->next = nullptr; tail = prev;

} else {

head = nullptr; tail = nullptr;

}

delete temp; count--;

}

void displayMembers() { Node\* current = head; while (current != nullptr) {

cout << "PRN: " << current->prn << ", Name: " << current->name << endl; current = current->next;

}

}

int getTotalMembers() { return count;

}

void concatenate(PinnacleClub& other) { if (head == nullptr) {

head = other.head; tail = other.tail;

} else if (other.head != nullptr) { tail->next = other.head;

tail = other.tail;

}

count += other.count; other.head = nullptr; other.tail = nullptr; other.count = 0;

}

~PinnacleClub() { Node\* current = head;

while (current != nullptr) { Node\* next = current->next; delete current;

current = next;

}

}

};

int main() { PinnacleClub club1; PinnacleClub club2;

string prn, name;

cout << "Enter PRN and Name for President of Club 1: "; cin >> prn >> name;

club1.addPresident(prn, name);

cout << "Enter PRN and Name for Secretary of Club 1: "; cin >> prn >> name;

club1.addSecretary(prn, name);

char addMore; do {

cout << "Enter PRN and Name for a new member of Club 1: "; cin >> prn >> name;

club1.addMember(prn, name);

cout << "Do you want to add another member? (y/n): "; cin >> addMore;

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

cout << "Enter PRN and Name for President of Club 2: "; cin >> prn >> name;

club2.addPresident(prn, name);

cout << "Enter PRN and Name for Secretary of Club 2: "; cin >> prn >> name;

club2.addSecretary(prn, name);

do {

cout << "Enter PRN and Name for a new member of Club 2: "; cin >> prn >> name;

club2.addMember(prn, name);

cout << "Do you want to add another member? (y/n): "; cin >> addMore;

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

cout << "\nClub 1 Members:\n"; club1.displayMembers();

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

cout << "\nClub 2 Members:\n"; club2.displayMembers();

cout << "Total Members in Club 2: " << club2.getTotalMembers() << endl; club1.concatenate(club2);

cout << "\nClub 1 Members after concatenation:\n"; club1.displayMembers();

cout << "Total Members in Club 1 after concatenation: " << club1.getTotalMembers() << endl;

club1.removePresident();

cout << "\nClub 1 Members after removing president:\n"; club1.displayMembers();

club1.removeSecretary();

cout << "\nClub 1 Members after removing secretary:\n"; club1.displayMembers();

return 0;

}

**OUTPUT:**



