#include <iostream>

#include <string>

using namespace std;

class node {

int prn;

string name;

node\* next;

public:

node(int x, string nm) {

prn = x;

next = NULL;

name = nm;

}

friend class list;

};

class list {

node\* start;

public:

list() { start = NULL; }

void create();

void display();

void insertAtBeginning();

void insertAtEnd();

void insertAfter();

void deleteAtFirst();

void deleteByValue();

void deleteAtEnd();

int computeTotal();

void sortlist();

void concatlist(list& q1);

void displayRev(node\* t);

bool reverseDisplay() {

if (start == NULL)

return false;

node\* temp = start;

displayRev(temp);

cout << "(President)";

return true;

}

};

void list::create() {

int no;

string nam;

if (start == NULL) {

cout << "Enter the PRN number: ";

cin >> no;

cout << "Enter the name: ";

cin >> nam;

start = new node(no, nam);

cout << "\n=====List is created========";

} else {

cout << "\nList is already created";

}

}

void list::display() {

node\* t = start;

if (start == NULL)

cout << "\nList is empty: ";

else {

cout << "\n======List======\n";

while (t != NULL) {

cout << t->prn << " " << t->name << "\n";

t = t->next;

}

}

}

void list::insertAtBeginning() {

int no;

string nam;

if (start == NULL) {

create();

} else {

cout << "\nEnter PRN Number: ";

cin >> no;

cout << "Enter Name: ";

cin >> nam;

node\* temp = new node(no, nam);

temp->next = start;

start = temp;

cout << "\nInserted " << temp->name << " at the beginning\n";

}

}

void list::insertAtEnd() {

int no;

string nam;

if (start == NULL) {

create();

} else {

cout << "\nEnter PRN Number: ";

cin >> no;

cout << "Enter Name: ";

cin >> nam;

node\* t = start;

while (t->next != NULL)

t = t->next;

node\* p = new node(no, nam);

t->next = p;

}

}

void list::insertAfter() {

int prev\_no;

cout << "\nEnter PRN No after which you want to insert: ";

cin >> prev\_no;

node\* t = start;

string nam;

int flag = 0, no;

while (t != NULL) {

if (t->prn == prev\_no) {

flag = 1;

break;

}

t = t->next;

}

if (flag == 1) {

cout << "\nEnter PRN Number: ";

cin >> no;

cout << "Enter Name: ";

cin >> nam;

node\* p = new node(no, nam);

p->next = t->next;

t->next = p;

} else {

cout << "\n" << prev\_no << " is not in the list";

}

}

void list::deleteAtFirst() {

if (start == NULL) {

cout << "\nClub is Empty";

} else {

node\* t = start;

start = start->next;

delete t;

cout << "\nPresident Deleted";

}

}

void list::deleteByValue() {

int no, flag = 0;

node \*t = start, \*prev = NULL;

if (start == NULL) {

cout << "\nList/Club is empty";

} else {

cout << "\nEnter PRN No. of member to be deleted: ";

cin >> no;

while (t != NULL) {

if (t->prn == no) {

flag = 1;

break;

}

prev = t;

t = t->next;

}

if (flag == 1) {

if (prev == NULL) { // If first node matches

start = t->next;

} else {

prev->next = t->next;

}

delete t;

cout << "\nMember with PRN No: " << no << " is deleted";

} else {

cout << "\nMember not found in List.\nPresident or Secretary cannot be deleted.";

}

}

}

int main() {

list l1;

l1.create();

l1.display();

l1.insertAtBeginning();

l1.insertAtEnd();

l1.insertAfter();

l1.display();

l1.deleteAtFirst();

l1.display();

l1.deleteByValue();

l1.display();

return 0;

}