#include<bits/stdc++.h>

using namespace std;

struct Node

{

pair<string,int> data;

Node \*next;

};

void insertMember(Node \*president, Node \*member)

{

Node \*temp = president;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = member;

}

void createMember(Node \*president)

{

string name;

int prn;

cout << "Enter the name-\n";

cin >> name;

cout << "Enter the prn-\n";

cin >> prn;

Node \*member = new Node;

member->data.first = name;

member->data.second = prn;

member->next = NULL;

insertMember(president, member);

}

void display(Node \*president)

{

Node \*temp1 = president;

while (temp1 != NULL)

{

cout << temp1->data.first << " " << temp1->data.second << endl;

temp1 = temp1->next;

}

}

void totalMembers(Node \*president)

{

int count = 0;

Node \*temp = president;

while (temp != NULL)

{

count++;

temp = temp->next;

}

cout << "The total members in the club are- " << count << endl;

}

void deleteMember(Node \*&president)

{

Node \*temp = president;

int position;

cout << "Enter the position of the member to be deleted-\n";

cin >> position;

int count = 0;

if (president == NULL)

{

cout << "The member linked list is already empty-\n";

}

else if (position == 0)

{

president = temp->next;

delete temp;

}

else

{

while (count != position-1)

{

temp = temp->next;

count++;

}

if (temp->next == NULL)

{

cout << "Element is not present in the member linked list" << endl;

}

Node \*todelete = temp->next;

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

delete todelete;

}

}

Node\* concatenate(Node \*president1, Node \*president2)

{

Node \*head = president1;

while (head->next != NULL)

{

head = head->next;

}

head->next = president2;

return president1;

}

int main()

{

Node \*president1 = new Node();

Node \*president2 = new Node();

string president1\_name;

int president1\_prn;

cout << "Enter president-1's name-\n";

cin >> president1\_name;

cout << "Enter president-1's prn-\n";

cin >> president1\_prn;

president1->data.first = president1\_name;

president1->data.second = president1\_prn;

president1->next = NULL;

int members1;

cout << "Enter number of members except president-1-\n";

cin >> members1;

while (members1 != 0)

{

createMember(president1);

members1--;

}

string president2\_name;

int president2\_prn;

cout << "Enter president-2's name-\n";

cin >> president2\_name;

cout << "Enter president-2's prn-\n";

cin >> president2\_prn;

president2->data.first = president2\_name;

president2->data.second = president2\_prn;

president2->next = NULL;

int members2;

cout << "Enter number of members except president-2-\n";

cin >> members2;

while (members2 != 0)

{

createMember(president2);

members2--;

}

char decider;

cout << "Enter Y for starting/continuing or N for exitting-\n";

cin >> decider;

while (decider == 'Y')

{

int choice;

int president\_choice;

cout << "1- Display the members\n2- Display total members\n3- Delete a member\n4- Concatenate the two member lists\n";

cin >> choice;

switch (choice)

{

case 1:

cout << "Enter 1 for president 1 and 2 for president 2-\n";

cin >> president\_choice;

if (president\_choice == 1)

{

display(president1);

break;

}

else

{

display(president2);

break;

}

case 2:

cout << "Enter 1 for president 1 and 2 for president 2-\n";

cin >> president\_choice;

if (president\_choice == 1)

{

totalMembers(president1);

break;

}

else

{

totalMembers(president2);

break;

}

case 3:

cout << "Enter 1 for president 1 and 2 for president 2-\n";

cin >> president\_choice;

if (president\_choice == 1)

{

deleteMember(president1);

break;

}

else

{

deleteMember(president2);

break;

}

case 4:

cout << "Concatenating the two member linked lists....\n";

concatenate(president1,president2);

break;

default:

break;

}

cout << "Enter Y for starting/continuing or N for exitting-\n";

cin >> decider;

}

cout << "The program has exitted!";

return 0;

}