```
/*Department of Computer Engineering has student's 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) Two linked lists exists for two divisions. Concatenate two lists.*/


#include <iostream>
using namespace std;

class Node {
public:
  int prn;
  string name, designation;
  Node *next;
};

class SLL {
public:
  Node *header = NULL;

  Node *create_new_node() {
    cout << "input: " << endl;
    Node *new_node = new Node();
    cout << "enter prn: " << endl;
    cin >> new_node->prn;
    cout << "enter name: " << endl;
    cin >> new_node->name;
    cout << "enter designation: " << endl;
    cin >> new_node->designation;
    new_node->next = NULL;
    return new_node;
  }

  void insertPresident() {
    Node *temp = create_new_node();
    if (header != NULL) {
      temp->next = header;
      header = temp;
    } else {
      header = temp;
    }
  }
```

```
void insertSecretary() {
  Node *temp1 = create_new_node();
  Node *temp2 = header;
  if (header == NULL) {
    header = temp1;
  } else {
    while (temp2->next != NULL) {
      temp2 = temp2->next;
    }
    temp2->next = temp1;
  }
}

void insertMember(int pos) {
  Node *temp1 = create_new_node();
  Node *temp2 = header;
  if (header == NULL) {
    header = temp1;
  } else {
    int i = 1;
    while (i < pos - 1) {
      temp2 = temp2->next;
      i++;
    }
    temp1->next = temp2->next;
    temp2->next = temp1;
  }
}

void deletePresident() {
  if (header != NULL) {
    header = header->next;
  } else {
    cout << "Doesn't exist" << endl;
  }
}

void deleteSecretary() {
  if (header != NULL) {
    Node *temp = header;
    Node *temp1;
    while (temp->next != NULL) {
      temp1 = temp;
      temp = temp->next;
    }
    temp1->next = NULL;
  } else {
      cout << "Doesn't exist" << endl;
    }
  }

  void deleteMember(int pos) {
    if (header != NULL) {
      Node *temp = header;
      Node *temp1;
      int i = 1;
      while (i < pos - 1 && temp->next != NULL) {
        temp = temp->next;
        i++;
      }
      temp1 = temp->next;
      temp->next = temp1->next;
    } else {
      cout << "Doesn't exist" << endl;
    }
  }

  void display() {
    int count = 0;
    Node *temp = header;
    while (temp != NULL) {
      cout << temp->prn << " ";
      if (temp->next != NULL) {
        cout << "->";
      }
      temp = temp->next;
      count++;
    }
    cout << "null" << endl;
    cout << "Total number of members: " << count << endl;
  }

  void concatenate(SLL &obj) {
    Node *temp2 = header;
    while (temp2->next != NULL) {
      temp2 = temp2->next;
    }
    temp2->next = obj.header;
  }
};

int main() {
  SLL s1, s2;
  int choice1, choice2;
```

```
while (true) {
  cout << "Choose one of the following divisions: " << endl;
  cout << "1. Division 1" << endl;
  cout << "2. Division 2" << endl;
  cout << "0. Exit" << endl;
  cin >> choice1;

  if (choice1 == 0) {
    break;
  }

  switch (choice1) {
  case 1:
    while (true) {
      cout << "Division 1 Operations:" << endl;
      cout << "1. Insert president" << endl;
      cout << "2. Insert secretary" << endl;
      cout << "3. Insert member" << endl;
      cout << "4. Delete president" << endl;
      cout << "5. Delete secretary" << endl;
      cout << "6. Delete member" << endl;
      cout << "7. Display" << endl;
      cout << "8. Concatenate" << endl;
      cout << "0. Return to division selection" << endl;
      cin >> choice2;

      if (choice2 == 0) {
        break;
      }

      switch (choice2) {
      case 1:
        s1.insertPresident();
        break;
      case 2:
        s1.insertSecretary();
        break;
      case 3:
        int pos;
        cout << "Enter the position: " << endl;
        cin >> pos;
        s1.insertMember(pos);
        break;
      case 4:
        s1.deletePresident();
        break;
      case 5:
        s1.deleteSecretary();
      break;
    case 6:
      int pos1;
      cout << "Enter the position: " << endl;
      cin >> pos1;
      s1.deleteMember(pos1);
      break;
    case 7:
      s1.display();
      break;
    case 8:
      s1.deleteSecretary();
      s2.deletePresident();
      s1.concatenate(s2);
      break;
    default:
      cout << "Invalid choice. Please try again." << endl;
    }
  }
  break;

case 2:
  while (true) {
    cout << "Division 2 Operations:" << endl;
    cout << "1. Insert president" << endl;
    cout << "2. Insert secretary" << endl;
    cout << "3. Insert member" << endl;
    cout << "4. Delete president" << endl;
    cout << "5. Delete secretary" << endl;
    cout << "6. Delete member" << endl;
    cout << "7. Display" << endl;
    cout << "8. Concatenate" << endl;
    cout << "0. Return to division selection" << endl;
    cin >> choice2;

    if (choice2 == 0) {
      break;
    }

    switch (choice2) {
    case 1:
      s2.insertPresident();
      break;
    case 2:
      s2.insertSecretary();
      break;
    case 3:
      int pos;
```

```
          cout << "Enter the position: " << endl;
          cin >> pos;
          s2.insertMember(pos);
          break;
        case 4:
          s2.deletePresident();
          break;
        case 5:
          s2.deleteSecretary();
          break;
        case 6:
          int pos1;
          cout << "Enter the position: " << endl;
          cin >> pos1;
          s2.deleteMember(pos1);
          break;
        case 7:
          s2.display();
          break;
        case 8:
          s2.deleteSecretary();
          s1.deletePresident();
          s2.concatenate(s1);
          break;
        default:
          cout << "Invalid choice. Please try again." << endl;
        }
      }
      break;

    default:
      cout << "Invalid division choice. Please try again." << endl;
    }
  }

  cout << "Exiting program..." << endl;
  cout << "Thank You" << endl;
  return 0;
}
```