**Assignment No: 2**

**Problem Statement:**

Department of Information Technology has a student's club named 'Pinnacle Club'. Students of the Second, third and final year of the department can be granted membership on request. Similarly, one may cancel the membership of the club. The first node is reserved for the president of the club and the last node is reserved for the secretary of the club. Write C program to maintain club member‟s information using a singly linked list. Store student PRN and Name. Write functions to a) Add and delete the members as well as the president or even secretary. b) Compute the total number of members of the club c) Display members of the club d) Display list in reverse order using recursion.

**Outcomes:**

Students will be able to:

1. Construct Linear data structures using arrays and linked lists.

**Code:**

#include <iostream>

using namespace std;

class Node

{

public:

string name, prn, role;

bool isPresent;

Node \*next;

Node \*prev;

Node(string name, string prn, string role, bool isPresent)

{

this->name = name;

this->prn = prn;

this->role = role;

this->isPresent = isPresent;

this->next = NULL;

this->prev = NULL;

}

};

class List

{

Node \*president, \*secretary;

int noOfMembers;

public:

List()

{

president = new Node("", "", "President", false);

secretary = new Node("", "", "Secretary", false);

noOfMembers = 0;

}

void addPresident()

{

cout << "\nEnter New President's Name: ";

cin >> president->name;

cout << "\n Enter New President's PRN No. : ";

cin >> president->prn;

president->isPresent = true;

cout << "\nNew President Added successfully !!!";

}

void removePresident()

{

cout << "\nDeleted President was: ";

cout << "\nPresident Name: " << president->name;

cout << "\nPresident PRN: " << president->prn;

president->name = "";

president->prn = "";

president->isPresent = false;

}

void addSecretary()

{

cout << "\nEnter New Secretary's Name: ";

cin >> secretary->name;

cout << "\n Enter New Secretary's PRN No. : ";

cin >> secretary->prn;

secretary->isPresent = true;

cout << "\nNew Secretary Added successfully !!!";

}

void removeSecretary()

{

cout << "\nDeleted Secretary was: ";

cout << "\nSecretary Name: " << secretary->name;

cout << "\nSecretary PRN: " << secretary->prn;

secretary->name = "";

secretary->prn = "";

secretary->isPresent = false;

}

void addMember()

{

noOfMembers++;

string name, prn;

cout << "\nEnter the Name of the member: ";

cin >> name;

cout << "\nEnter the PRN of the member: ";

cin >> prn;

Node \*newNode = new Node(name, prn, "Member", true);

cout << "Input";

if (noOfMembers == 1)

{

cout << "Pointers";

president->next = newNode;

secretary->prev = newNode;

newNode->next = secretary;

newNode->prev = president;

cout << "\nThe member is inserted successfully !!!";

return;

}

newNode->next = secretary;

newNode->prev = secretary->prev;

secretary->prev->next = newNode;

secretary->prev = newNode;

cout << "\nThe member is inserted successfully !!!";

return;

}

void removeMember()

{

if (noOfMembers == 0)

{

cout << "\nNo members present in the organization !!!";

return;

}

string searchPrn;

cout << "\nEnter the PRN to remove the member: ";

cin >> searchPrn;

Node \*ptr = president->next;

while (ptr != secretary && ptr->prn != searchPrn)

{

ptr = ptr->next;

}

if (ptr == secretary)

{

cout << "\nNo member found with the given PRN !!!";

return;

}

noOfMembers--;

ptr->prev->next = ptr->next;

ptr->next->prev = ptr->prev;

cout << "\nDeleted Member is: ";

cout << "\nName: " << ptr->name;

cout << "\nPRN: " << ptr->prn;

delete ptr;

return;

}

void displayPresident()

{

if (!president->isPresent)

{

cout << "\nThe President is not present !!!";

return;

}

cout << "\nPresident is: ";

cout << "\nPresident Name: " << president->name;

cout << "\nPresident PRN: " << president->prn;

return;

}

void displaySecretary()

{

if (!secretary->isPresent)

{

cout << "\nThe Secretary is not present !!!";

return;

}

cout << "\nSecretary is: ";

cout << "\nSecretary Name: " << secretary->name;

cout << "\nSecretary PRN: " << secretary->prn;

return;

}

void displayMembers()

{

if (noOfMembers == 0)

{

cout << "\nNo members present in the organization !!!";

return;

}

Node \*temp = president->next;

cout << "\nSr.No.\t\tPRN\t\tNAME\n";

int i = 1;

while (temp != secretary)

{

cout << i << "\t\t" << temp->prn << "\t\t" << temp->name << endl;

temp = temp->next;

i++;

}

return;

}

void printReverseMembers(Node \*node, int i)

{

if (node == president)

{

return;

}

cout << i << "\t\t" << node->prn << "\t\t" << node->name << endl;

i++;

printReverseMembers(node->prev, i);

}

void displayMembersReverse()

{

int i = 1;

if (noOfMembers == 0)

{

cout << "\nNo members present in the organization !!!";

return;

}

cout << "\nSr.No.\t\tPRN\t\tNAME\n";

printReverseMembers(secretary->prev, i);

}

int getNoOfMembers()

{

return noOfMembers;

}

};

int main()

{

List list;

int choice;

do

{

cout << "\n\n\n<<<<< <<< < MENU > >>> >>>>>" << endl;

cout << "1 | Add a member" << endl;

cout << "2 | Remove a member" << endl;

cout << "3 | Add a Secretary" << endl;

cout << "4 | Add a President" << endl;

cout << "5 | Remove a Secretary" << endl;

cout << "6 | Remove a President" << endl;

cout << "7 | Display members" << endl;

cout << "8 | Display members(Reverse Order)" << endl;

cout << "9 | Total number of members of the club" << endl;

cout << "10| Display President and Secretary" << endl;

cout << "0 | Exit" << endl;

cout << "\n Enter the choice: ";

cin >> choice;

switch (choice)

{

case 1:

list.addMember();

break;

case 2:

list.removeMember();

break;

case 3:

list.addSecretary();

break;

case 4:

list.addPresident();

break;

case 5:

list.removeSecretary();

break;

case 6:

list.removePresident();

break;

case 7:

list.displayMembers();

break;

case 8:

list.displayMembersReverse();

break;

case 9:

cout << "Total number of members: " << list.getNoOfMembers();

break;

case 10:

list.displayPresident();

list.displaySecretary();

break;

case 0:

cout << "\nShutting down...";

break;

default:

cout << "\nPlease Enter a valid choice...";

break;

}

} while (choice != 0);

}

**Output:**

1) 2)