Program:

*#include* <iostream>

*#include* <string>

*using* *namespace* std;

*class* list;

*class* node

{

*int* MIS;

    string name;

    node *\**next;

*public:*

    node(*int* *x*, string *nem*)

    {

        MIS *=* *x*;

        next *=* NULL;

        name *=* *nem*;

    }

*friend* *class* list;

};

*class* list

{

    node *\**start;

*public:*

    list()

    {

        start *=* NULL;

    }

*void* create();

*void* display();

*void* InsertPresident();

*void* InsertSecretary();

*void* InsertMember();

*void* DeletePresident();

*void* DeleteMember();

*void* DeleteSecretary();

*void* SortList();

*void* concat(list *&q1*);

*void* RevDisplay(node *\*t*);

*int* ContTotal();

*bool* DisplayReverse()

    {

*if* (start *==* NULL)

*return* false;

        node *\**temp *=* start;

        RevDisplay(temp);

*return* true;

    }

};

*void* list::RevDisplay(node *\*t*)

{

*if* (*t* *==* NULL)

*return*;

*else*

    {

        RevDisplay(*t*->next);

        cout *<<* "\nMIS NO:" *<<* *t*->MIS *<<* " Name: " *<<* *t*->name;

    }

}

*void* list::create()

{

*int* no;

    string StudName;

*if* (start *==* NULL)

    {

        cout *<<* "Enter MIS number: ";

        cin *>>* no;

        cout *<<* "Enter name: ";

        cin *>>* StudName;

        cout *<<* StudName;

        start *=* *new* node(no, StudName);

        cout *<<* "\n\*Added Successfully\*";

    }

*else*

    {

        cout *<<* "\nList is Already Created.";

    }

}

*void* list::display()

{

    node *\**t;

    t *=* start;

*if* (start *==* NULL)

        cout *<<* "\nList is Empty";

*else*

    {

        cout *<<* "\n\*\*\*\*\* List: \*\*\*\*\*\n";

*while* (t *!=* NULL)

        {

            cout *<<* t->MIS *<<* " " *<<* t->name *<<* " \n";

            t *=* t->next;

        }

    }

}

*void* list::InsertPresident()

{

*int* no;

    string StudName;

    node *\**temp;

*if* (start *==* NULL)

    {

        create();

    }

*else*

    {

        cout *<<* "\nEnter MIS number: ";

        cin *>>* no;

        cout *<<* "Enter name: ";

        cin *>>* StudName;

        temp *=* *new* node(no, StudName);

        temp->next *=* start;

        start *=* temp;

*//;*

        cout *<<* " President " *<<* temp->name *<<* "Inserted Successfully.";

    }

}

*void* list::InsertSecretary()

{

*int* no;

    string StudName;

    node *\**t;

*if* (start *==* NULL)

        create();

*else*

    {

        cout *<<* "\nEnter MIS number: ";

        cin *>>* no;

        cout *<<* "Enter name: ";

        cin *>>* StudName;

        t *=* start;

*while* (t->next *!=* NULL)

            t *=* t->next;

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

        t->next *=* p;

    }

    cout *<<* " Secretary Inserted Successfully.";

}

*void* list::InsertMember()

{

*int* prev\_no;

    cout *<<* "\nEnter Member MIS Number after do you want insert:";

    cin *>>* prev\_no;

    node *\**t;

    t *=* start;

    string StudName;

*int* flag *=* 0, no;

*while* (t *!=* NULL)

    {

*if* (t->MIS *==* prev\_no)

        {

            flag *=* 1;

*break*;

        }

        t *=* t->next;

    }

*if* (flag *==* 1)

    {

        node *\**p;

        cout *<<* "\nEnter MIS number: ";

        cin *>>* no;

        cout *<<* "Enter name: ";

        cin *>>* StudName;

        p *=* *new* node(no, StudName);

        p->next *=* t->next;

        t->next *=* p;

    }

*else*

    {

        cout *<<* "\n"

*<<* prev\_no *<<* " Not found.";

    }

    cout *<<* "Member added Successfully.";

}

*void* list::DeletePresident()

{

    node *\**t;

*if* (start *==* NULL)

        cout *<<* "\nClub is Empty";

*else*

    {

        t *=* start;

        start *=* start->next;

        t->next *=* NULL;

*delete* t;

        cout *<<* "\nPresident deleted successfully.";

    }

}

*void* list::DeleteMember()

{

*int* no, flag *=* 0;

    node *\**t, *\**prev;

*if* (start *==* NULL)

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

*else*

    {

        cout *<<* "\nEnter member MIS number to be deleted: ";

        cin *>>* no;

        t *=* start->next;

*while* (t->next *!=* NULL)

        {

*if* (t->MIS *==* no)

            {

                flag *=* 1;

*break*;

            }

            prev *=* t;

            t *=* t->next;

        }

*if* (flag *==* 1)

        {

            prev->next *=* t->next;

            t->next *=* NULL;

*delete* t;

            cout *<<* "\nMember: " *<<* no *<<* " is deleted successfully.";

        }

*else*

            cout *<<* "\nMember not Found.";

    }

}

*void* list::DeleteSecretary()

{

    node *\**t, *\**prev;

    t *=* start;

*if* (start *==* NULL)

        cout *<<* "\nEmpty..";

*else*

    {

*while* (t->next *!=* NULL)

        {

            prev *=* t;

            t *=* t->next;

        }

        prev->next *=* NULL;

*delete* t;

        cout *<<* "\nSecretary Deleted successfully.";

    }

}

*int* list::ContTotal()

{

    node *\**t;

*int* count *=* 0;

    t *=* start;

*if* (start *==* NULL)

    {

        cout *<<* "\nempty.";

*return* 0;

    }

*while* (t *!=* NULL)

    {

        count*++*;

        t *=* t->next;

    }

*return* count;

}

*void* list::SortList()

{

    node *\**i, *\**j, *\**last *=* NULL;

*int* tMIS;

    string tname;

*if* (start *==* NULL)

    {

        cout *<<* "\nempty.";

*return*;

    }

*for* (i *=* start; i->next *!=* NULL; i *=* i->next)

    {

*for* (j *=* start; j->next *!=* last; j *=* j->next)

        {

*if* ((j->MIS) *>* (j->next->MIS))

            {

                tMIS *=* j->MIS;

                tname *=* j->name;

                j->MIS *=* j->next->MIS;

                j->name *=* j->next->name;

                j->next->MIS *=* tMIS;

                j->next->name *=* tname;

            }

        }

    }

    cout *<<* "\n List is sorted.";

    display();

}

*void* list::concat(list *&q1*)

{

    node *\**t, *\**p;

    t *=* *q1*.start;

*if* (t *==* NULL)

    {

        cout *<<* "\nList 2 is empty";

*return*;

    }

    p *=* start;

*while* (p->next *!=* NULL)

    {

        p *=* p->next;

    }

    p->next *=* t;

*q1*.start *=* NULL;

    cout *<<* "\nAfter concatenationlist";

    display();

}

*int* main()

{

    list *\**l;

*int* choice, selectList;

    list l1, l2;

    l *=* *&*l1;

X:

    cout *<<* "Welcome to GHRCEM Club!" *<<* endl;

    cout *<<* "\n1.List 1";

    cout *<<* "\n2.List 2";

    cout *<<* "\nEnter choice: ";

    cin *>>* selectList;

*if* (selectList *==* 1)

    {

        l *=* *&*l1;

    }

*else* *if* (selectList *==* 2)

    {

        l *=* *&*l2;

    }

*else*

    {

        cout *<<* "\nWrong list Number.";

*goto* X;

    }

*do*

    {

        cout *<<* "\n";

        cout *<<* "\n\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*";

        cout *<<* "\n\* 1.Create                      \*";

        cout *<<* "\n\* 2.Insert President            \*";

        cout *<<* "\n\* 3.Insert Secretary            \*";

        cout *<<* "\n\* 4.insert Member               \*";

        cout *<<* "\n\* 5.Display ALL Member          \*";

        cout *<<* "\n\* 6.Delete President            \*";

        cout *<<* "\n\* 7.Delete Secretary            \*";

        cout *<<* "\n\* 8.Delete Member               \*";

        cout *<<* "\n\* 9.Count members               \*";

        cout *<<* "\n\* 10.Sort list                  \*";

        cout *<<* "\n\* 11.To concatenate two list    \*";

        cout *<<* "\n\* 12.Reverse Display            \*";

        cout *<<* "\n\* 13.Go back & select list      \*";

        cout *<<* "\n\* 0.Exit                        \*";

        cout *<<* "\n\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*";

        cout *<<* "\nEnter your choice:\t             ";

        cin *>>* choice;

        cout *<<* "\n";

*switch* (choice)

        {

*case* 1:

            l->create();

*break*;

*case* 2:

            l->InsertPresident();

*break*;

*case* 3:

            l->InsertSecretary();

*break*;

*case* 4:

            l->InsertMember();

*break*;

*case* 5:

            l->display();

*break*;

*case* 6:

            l->DeletePresident();

*break*;

*case* 7:

            l->DeleteSecretary();

*break*;

*case* 8:

            l->DeleteMember();

*break*;

*case* 9:

            cout *<<* "\nTotal members of Club: " *<<* l->ContTotal();

*break*;

*case* 10:

            l->SortList();

*break*;

*case* 11:

            l1.concat(l1);

*break*;

*case* 12:

            l->DisplayReverse();

*break*;

*case* 13:

*goto* X;

*break*;

        deafult:

            cout *<<* "Wrong input try again";

        }

    } *while* (choice *!=* 0);

    cout *<<* "\nThank you!!\n";

*return* 0;

}

Output :









