

PROBLEM STATEMENT:

To create ADT that implements the SET concept.

- a. Add (newElement) -Place a value into the set
- b. Remove (element) Remove the value
- c. Contains (element) Return true if element is in collection
- d. Size () Return number of values in collection Iterator () Return an iterator used to loop over collection
- e. Intersection of two sets
- f. Union of two sets
- g. Difference between two sets
- h.Subset

*/

```
#include <iostream>
#include<list>
#include<cstdlib>
using namespace std;

class set{

private:
    int num,flag=1;
public:
    list<int>l,l1,u,l,d;
    list<int>::iterator t,t1,t2,t3,t4;
void add();
void delete1(int);
void search(int);
void searchB(int);
void display();
void union1();
void Intersection();
void insert();
void Difference();
};
void set::insert()
{
    int n,m;
    cout<<"\nSET A:\n";
    cout<<"How many Elements You want Add in Set A:\n";
    cin>>n;
    cout<<"Enter Elements\n";

    for(int i=0;i<n;i++)
    {
        cin>>num;
        l.push_back(num);
    }
    cout<<"\nSET B:\n";
    cout<<"How many Elements You want Add in Set B:\n";
    cin>>m;
    cout<<"Enter Elements\n";

    for(int i=0;i<m;i++)
    {
```

```

        cin>>num;
        l1.push_back(num);
    }
}
void set::add()
{
    char c;

    cout<<"In Which Set do you want Add Element (A/B)\n";
    cin>>c;
    if(c=='A' || c=='a')
    {
        cout<<"Enter Elements\n";
        cin>>num;
        l.push_back(num);
        cout<<"\nElement Inserted\n";
    }
    else if(c=='B' || c=='b')
    {
        cout<<"Enter Elements\n";
        cin>>num;
        l1.push_back(num);
        cout<<"\nElement Inserted\n";
    }
    else
        cout<<"Invalid Set!!!";
}
void set::display()
{
    cout<<"The Elements for Set A:\n{t";
    for(t=l.begin();t!=l.end();t++)
    {
        cout<<*t<<"\t";
    }
    cout<<"}";
    cout<<"\n\n";
    cout<<"The Elements for Set B:\n{t";
    for(t1=l1.begin();t1!=l1.end();t1++)
    {
        cout<<*t1<<"\t";
    }
    cout<<"}";
}
void set::search(int key)
{
    for(t=l.begin(),t1=l1.begin();t!=l.end();t++,t1++)
    {
        if(*t==key || *t1==key)
        {
            cout<<"The Element is Present\n";
            flag=1;
            break;
        }
        else

```

```

        flag=0;
    }
    if(flag==0)
    {
        cout<<"The Element is not Present\n";
    }
}

void set::delete1(int key)
{
    if(l.empty() && l1.empty())
    {
        cout<<"The Set A & Set B is Empty\n";
    }
    else
    {
        search(key);
        if(flag==1)
        {
            l.remove(key);
            l1.remove(key);
            cout<<"Element Deleted\n";
        }
        else
            cout<<"Element not Deleted\n";
    }
}

}
void set::union1()
{
    int flag=0;
    for(t=l.begin(); t!=l.end(); t++)
    {
        u.push_back(*t);
    }
    for(t1=l1.begin(); t1!=l1.end(); t1++)
    {
        for(t2=u.begin(); t2!=u.end(); t2++)
        {
            if(*t1==*t2)
            {
                flag=0;
                break;
            }
            else
                flag=1;
        }
        if(flag==1)
        {
            u.push_back(*t1);
        }
    }
}

cout<<"The Union Set of A & B is : {\t";
for(t2=u.begin(); t2!=u.end(); t2++)
{

```

```

        cout<<*t2<<"\t";
    }
    cout<<"}";
}
void set::Intersection()
{
    for(t=l.begin();t!=l.end();t++)
    {
        for(t1=l1.begin();t1!=l1.end();t1++)
        {
            if(*t==*t1)
            {
                l.push_back(*t);
                break;
            }
        }
    }
    if(l.empty())
    {
        cout<<"There is no Common element in Set A & Set B\n";
    }
    else
    {
        cout<<"The Intersection Set of A & B is : {\t";
        for(t3=l.begin();t3!=l.end();t3++)
        {
            cout<<*t3<<"\t";
        }
        cout<<"}";
    }
}
void set::Difference()
{
    int flag=0;
    for(t=l.begin();t!=l.end();t++)
    {
        for(t1=l1.begin();t1!=l1.end();t1++)
        {
            if(*t==*t1)
            {
                flag=0;
                break;
            }
            else
                flag=1;
        }
        if(flag==1)
        {
            d.push_back(*t);
        }
    }
    if(d.empty())
    {
        cout<<"The Set A & Set B are Equal\n";
    }
    else

```

```

        {
            cout<<"The Difference Set of A & B is : {\t";
            for(t4=d.begin();t4!=d.end();t4++)
            {
                cout<<*t4<<"\t";
            }
            cout<<"}";
        }
    }
}

```

```

int main()
{
    set s;

    int ch,key;

    s.insert();
    while(1)
    {
        cout<<"\n\n-----\n";
        cout<<"\nSet Theory\n";
        cout<<"\n\n-----\n";
        cout<<"1.Add Element\n";
        cout<<"2.Delete Element\n";
        cout<<"3.Search Element\n";
        cout<<"4.Display\n";
        cout<<"5.Union\n";
        cout<<"6.Intersection\n";
        cout<<"7.Difference\n";
        cout<<"8.Exit\n";

        cout<<"Enter Your Choice: ";
        cin>>ch;

        switch(ch)
        {
            case 1:

                s.add();
                break;

            case 2:

                cout<<"Enter which Element to Deleted: ";
                cin>>key;
                s.delete1(key);
                break;

            case 3:

                cout<<"Enter the Element to be Searched : ";
                cin>>key;
                s.search(key);

                break;

            case 4:

```

```
        cout<<endl;
        s.display();

        break;

case 5:
    s.union1();
    break;
case 6:
    s.Intersection();
    break;
case 7:
    s.Difference();
    break;
case 8:
    cout<<"Exiting...";
    exit(1);
    break;
default:
    cout<<"Invalid Choice";

    }
    }
    return 0;
}
```