

ASSIGNMENT NO.1.

Aim :- To create ADT that implement the "set" concept.

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

Objective:- to study the different set operations.

Theory:-

Set is a container implemented in C++ language in STL and has a concept similar to how set is defined in mathematics. The facts that separates set from the other containers is that is it contains only the **distinct elements** and elements can be traversed in sorted order. Having the strong hold on sets is useful in competitive programming and solving algorithmic problems. The insertion and deletion in STL sets are discussed in this article. Sets have the most impact in mathematical set theory. These theories are used in many kinds of proofs, structures, and abstract algebra. Creating relations from different sets and codomains are also an important applications of sets.

In computer science, set theory is useful if you need to collect data and do not care about their multiplicity or their order. In databases, especially for relational databases, sets are very useful. There are many commands that finds unions, intersections, and differences of different tables and sets of data.

Program Code:-

```
#include<bits/stdc++.h>
using namespace std;

void display(char set1[100],char set2[100],int n,int m)
{
    cout << "\tSET 1 : " << endl;
    for(int i=0;set1[i]!='\0';i++)
    {
        cout << set1[i]<< " " ;
    }
}
```

```

        cout << endl;

        cout << "\tSET 2 : " << endl;
        for(int i=0;set2[i]!='\0';i++)
        {
            cout <<set2[i] << " " ;
        }
        cout << endl;
    }

void insert(char set1[100],char set2[100],int n,int m)
{
    int setno,i;
    cout << "\tIn which set do you wish to insert" << endl;
    cin >> setno;
    if(setno == 1)
    {
        for(i=0;i<set1[i]!='\0';i++);
        cout << "\tEnter the element" << endl;
        cin >> set1[i];
        set1[i+1]='\0';

    }
    else if(setno == 2)
    {
        for(i=0;set2[i]!='\0';i++);
        cout << "\tEnter the element" << endl;
        cin >> set2[i];
        set2[i+1]='\0';

    }
    else
    {
        cout << "\tINVALID SET NUMBER" << endl;
    }
    display(set1,set2,n,m);
}

void size(char set1[100],char set2[100],int n,int m)
{
    int lenSet1,lenSet2;
    for(lenSet1 = 0;set1[lenSet1]!='\0';lenSet1++);
    for(lenSet2 = 0;set2[lenSet2]!='\0';lenSet2++);
    cout << "\tSize of set 1 is : " << lenSet1 << endl;
    cout << "\tSize of set 2 is : " << lenSet2 << endl;
}

void contains(char set1[100],char set2[100],int n,int m)
{
    char temp;
    int flagset1 = 0;
    int flagset2 = 0;

```

```

    cout << "\tEnter the element you wish to search" << endl;
    cin >> temp;
    for(int i = 0 ;set1[i]!='\0';i++)
    {
        if ( set1[i] == temp)
        {
            cout << "Element is in set 1" << endl;
            flagset1 = 1;
        }
    }
    if(flagset1 == 0)
        cout << "\tElement not found" << endl;

    for(int i = 0 ;set2[i]!='\0';i++)
    {
        if ( set2[i] == temp)
        {
            cout << "Element is in set 2" << endl;
            flagset2 = 1;
        }
    }
    if(flagset2 == 0)
        cout << "\tElement not found" << endl;
}

void remove(char set1[100],char set2[100],int n,int m)
{
    int pos,setno;
    cout << "\tEnter the set number from which u wish to delete
element " << endl;
    cin >> setno;
    int i=1,len;
    if(setno == 1)
    {
        cout << "\tEnter the index from which you wish to delete
element" << endl;
        cin >> pos;
        for (len=0;set1[len]!='\0';len++);
        if(pos > len || pos < 1)
            cout << "\tINVALID POSITION" << endl;
        else
        {
            while(set1[pos-1]!='\0')
            {
                set1[pos-1] = set1[pos];
                pos++;
            }
        }
    }

    if(setno == 2)

```

```

        {
            cout << "\tEnter the position from which you wish to delete
element" << endl;
            cin >> pos;
            for (int len=0;set2[len]!='\0';len++)
            if(pos > len || pos < 1)
                cout << "\tINVALID POSITION" << endl;
            else
            {
                while(set2[pos-1]!='\0')
                {
                    set2[pos-1] = set2[pos];
                    pos++;
                }
            }
            display(set1,set2,n,m);
        }
    }

void intersection(char set1[100],char set2[100],int n,int m)
{
    for(int i = 0 ;set1[i]!='\0';i++)
    for(int j = 0 ;set2[j]!='\0';j++)
    {
        if(set1[i]==set2[j])
        {
            cout<<set1[i]<<" ";
        }
    }
    cout<<endl;
}

void union1(char set1[100],char set2[100],int n,int m)
{
    int j;
    for(int i=0;set1[i]!='\0';i++)
    {
        cout<<set1[i]<<" ";
    }
    for(int i = 0 ;set2[i]!='\0';i++)
    {
        bool flag=false;
        for( j = 0 ;set1[j]!='\0';j++)
        {
            if(set2[i]==set1[j])
            {
                flag=true;
            }
        }
        if(flag==false)
            cout<<set2[i]<<" ";
    }
}

```

```

}
void diff(char set1[100],char set2[100],int n,int m)
{
    int i,j,k,l,r;
    if(n>=m)
    {
        for(k=0;set1[i]!='\0';k++)
        {
            int flag=0;
            for( r = 0 ;set2[r]!='\0';r++)
            {
                if(set1[k]==set2[r])
                {
                    flag=1;
                }
            }
            if(flag==0)
            {
                cout<<set1[k]<<" ";
            }
        }
    }
    else
    {
        for(k=0;set2[k]!='\0';k++)
        {
            int flag=0;
            for( r = 0 ;set1[r]!='\0';r++)
            {
                if(set2[k]==set1[r])
                {
                    flag=1;
                }
            }
            if(flag==0)
            {
                cout<<set2[k]<<" ";
            }
        }
    }
}
void subset(char set1[100],char set2[100],int n,int m)
{
    int i;
    if(m>m)
    {
        cout<<"Set2 is not subset of set1";
    }
    else

```

```

    {
        for(i=0;i<m;i++)
            bool flag;
    }
}

int main()
{
    static char set1[100],set2[100],choice='y';
    int ch,i,n,m;
    cout << "\tEnter size of first set" << endl;
    cin >> n;
    cout << "\tEnter size of second set" << endl;
    cin >> m;
    cout << "\tEnter first set" << endl;
    for(i=0;i<n;i++)
    {
        cin >> set1[i];
    }

    cout << "\tEnter second set" << endl;
    for(i=0;i<m;i++)
    {
        cin >> set2[i];
    }

    while(choice == 'y')
    {
        cout << "\tMENU" << endl;
        cout << "\t1.Insert" << endl;
        cout << "\t2.Remove" << endl;
        cout << "\t3.Contains" << endl;
        cout << "\t4.Size" << endl;
        cout<<"\t5.Intersection"<<endl;
        cout<<"\t6.Union"<<endl;
        cout<<"\t7.Difference"<<endl;
        cin >> ch;
        switch(ch)
        {
            case 1:
                insert(set1,set2,n,m);
                break;
            case 2:
                remove(set1,set2,n,m);
                break;
            case 3:
                contains(set1,set2,n,m);
                break;
            case 4:
                size(set1,set2,n,m);

```

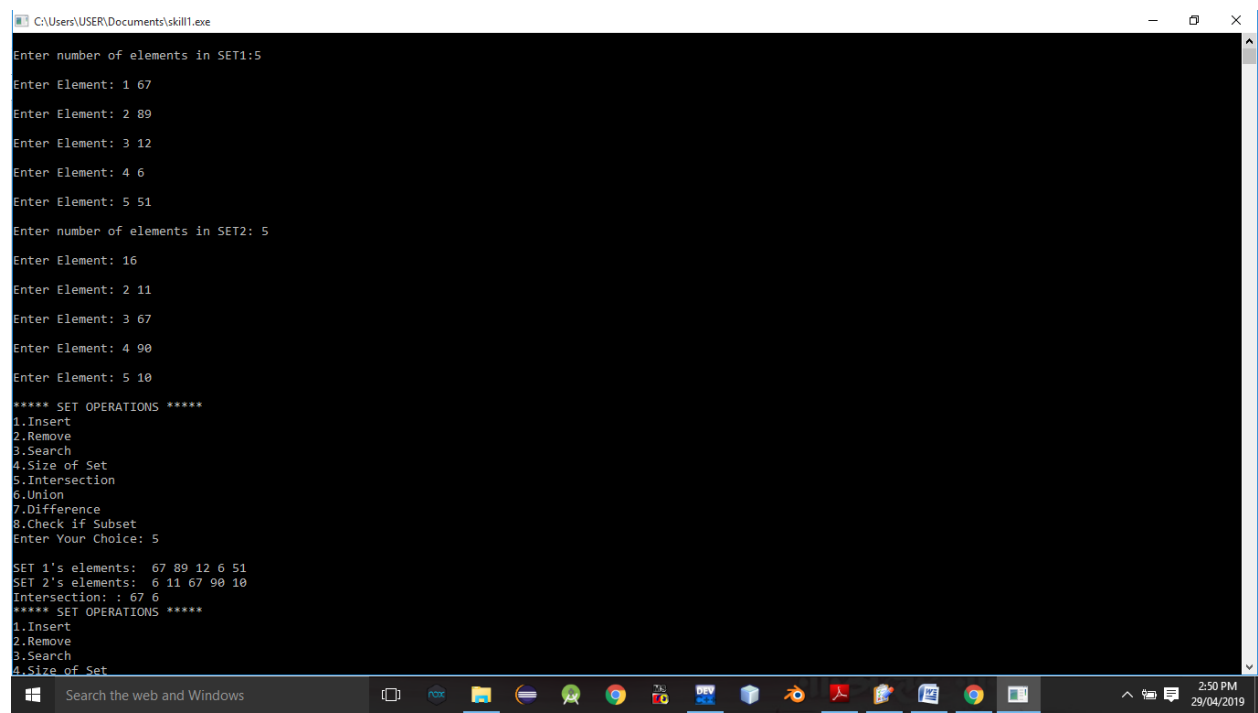
```
        break;
    case 5:
        intersection(set1,set2,n,m);
        break;
    case 6:union1(set1,set2,n,m);
        break;
    case 7:diff( set1,set2,n,m);
        break;
    default :
        cout << "\tINVALID CHOICE" << endl;

    cout << "\tDo you wish to continue" << endl;
    cout << "\tIf yes enter y" << endl;
    cin >> choice;

    }

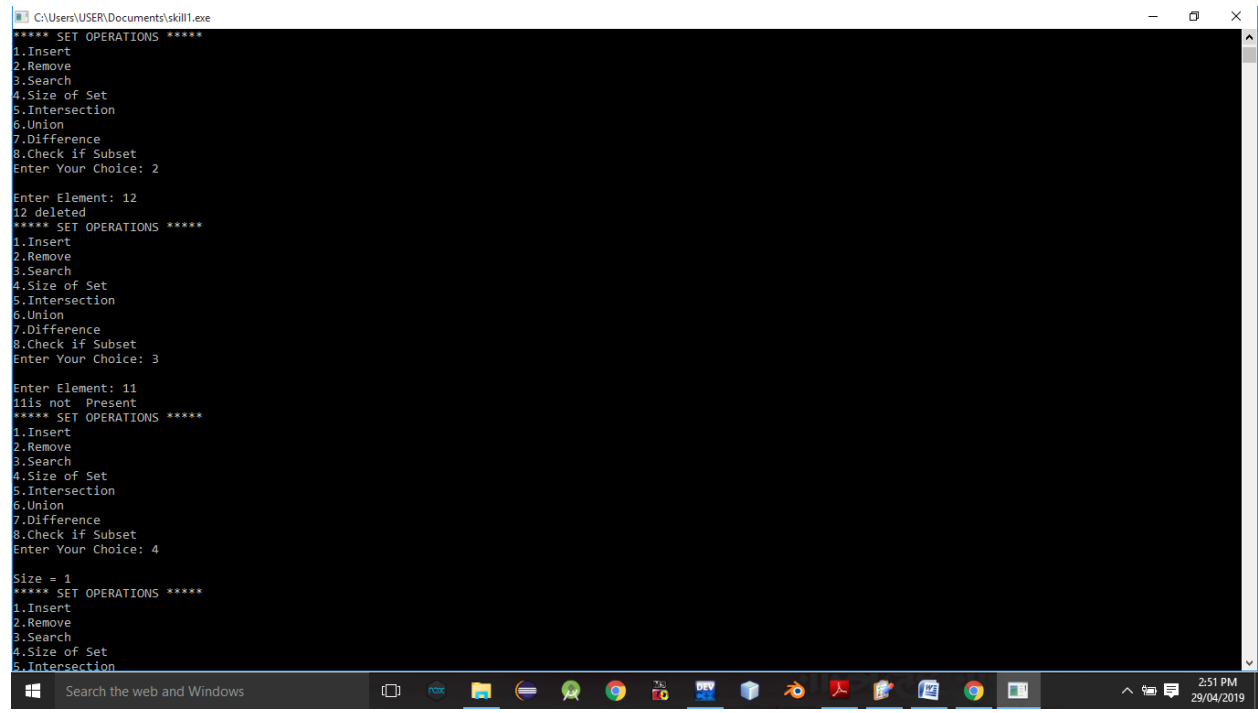
}
```

Output Screenshots:-



The screenshot shows a Windows command prompt window titled "C:\Users\USER\Documents\skill1.exe". The program prompts the user to enter the number of elements in SET1 (5) and SET2 (5). It then prompts for 5 elements for each set. SET1 elements are 67, 89, 12, 6, 51. SET2 elements are 16, 11, 67, 90, 10. The program displays the menu "***** SET OPERATIONS *****" with options 1 to 8. The user enters choice 5. The output shows SET 1's elements, SET 2's elements, and their intersection: 67 6. The taskbar at the bottom shows the time as 2:50 PM on 29/04/2019.

```
C:\Users\USER\Documents\skill1.exe
Enter number of elements in SET1:5
Enter Element: 1 67
Enter Element: 2 89
Enter Element: 3 12
Enter Element: 4 6
Enter Element: 5 51
Enter number of elements in SET2: 5
Enter Element: 16
Enter Element: 2 11
Enter Element: 3 67
Enter Element: 4 90
Enter Element: 5 10
***** SET OPERATIONS *****
1.Insert
2.Remove
3.Search
4.Size of Set
5.Intersection
6.Union
7.Difference
8.Check if Subset
Enter Your Choice: 5
SET 1's elements:  67 89 12 6 51
SET 2's elements:  6 11 67 90 10
Intersection: : 67 6
***** SET OPERATIONS *****
1.Insert
2.Remove
3.Search
4.Size of Set
```



```
C:\Users\USER\Documents\skill1.exe
***** SET OPERATIONS *****
1.Insert
2.Remove
3.Search
4.Size of Set
5.Intersection
6.Union
7.Difference
8.Check if Subset
Enter Your Choice: 2
Enter Element: 12
12 deleted
***** SET OPERATIONS *****
1.Insert
2.Remove
3.Search
4.Size of Set
5.Intersection
6.Union
7.Difference
8.Check if Subset
Enter Your Choice: 3
Enter Element: 11
11 is not Present
***** SET OPERATIONS *****
1.Insert
2.Remove
3.Search
4.Size of Set
5.Intersection
6.Union
7.Difference
8.Check if Subset
Enter Your Choice: 4
Size = 1
***** SET OPERATIONS *****
1.Insert
2.Remove
3.Search
4.Size of Set
5.Intersection
```

Conclusion:- Thus,we have studied different operations on set ADT.