ASSIGNMENT 1

AIM:-

To create ADT that implement the set concept.

A. ADD -place value in the set

B. Remove

C. Contains -return true if element is In the set

D. Size

E. Intersection of 2 sets

F. Union of 2 sets

G. Difference between 2 set

H. Subset

I.

OBJECTIVE:-

  To implement set concept and associated functions.

THEORY:-

Set theory, branch of mathematics that deals with the properties of well-defined collections of objects, which may or may not be of a mathematical nature, such as numbers or functions. The theory is less valuable in direct application to ordinary experience than as a basis for precise and adaptable terminology for the definition of complex and sophisticated mathematical concepts.

Set theory is a branch of mathematical logic that studies sets, which informally are collections of objects. Although any type of object can be collected into a set, set theory is applied most often to objects that are relevant to mathematics. The language of set theory can be used to define nearly all mathematical objects.

A **Set** is an unordered collection of objects, known as elements or members of the set.

An element 'a' belong to a set A can be written as 'a $\in$ A', 'a $\notin$ A' denotes that a is not an element of the set A.

Representation of a Set

A set can be represented by various methods. 3 common methods used for representing set:

1. Statement form.
2. Roaster form or tabular form method.
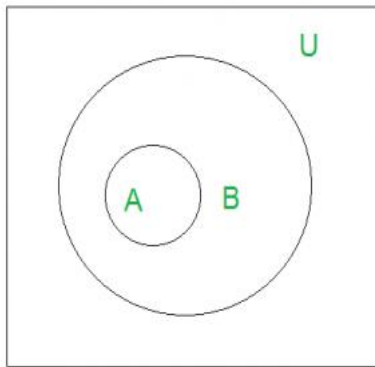3. Set Builder method.

Subset

A set A is said to be **subset** of another set B if and only if every element of set A is also a part of other set B.

Denoted by '⊆'.

'A ⊆ B ' denotes A is a subset of B.

To prove A is the subset of B, we need to simply show that if x belongs to A then x also belongs to B.
To prove A is not a subset of B, we need to find out one element which is part of set A but not belong to set B.

'U' denotes the universal set.
Above Venn Diagram shows that A is a subset of B.

Size of a Set
Size of a set can be finite or infinite.

For example

Finite set: Set of natural numbers less than 100.

Infinite set: Set of real numbers.

Size of the set S is known as **Cardinality number**, denoted as |S|.

Example: Let A be a set of odd positive integers less than 10.
Solution : A = {1,3,5,7,9}, Cardinality of the set is 5, i.e.,|A| = 5.

Union of the sets A and B, denoted $A \cup B$, is the set of all objects that are a member of A, or B, or both. The union of {1, 2, 3} and {2, 3, 4} is the set {1, 2, 3, 4} .

Intersection of the sets A and B, denoted $A \cap B$, is the set of all objects that are members of both A and B. The intersection of {1, 2, 3} and {2, 3, 4} is the set {2, 3} .

Set difference of U and A, denoted $U \setminus A$, is the set of all members of U that are not members of A. The set difference {1, 2, 3} \ {2, 3, 4} is {1} , while, conversely, the set difference {2, 3, 4} \ {1, 2, 3} is {4} . When A is a subset of U, the set difference $U \setminus A$ is also called the **complement** of A in U. In this case, if the choice of U is clear from the context, the notation A^c is sometimes used instead of $U \setminus A$, particularly if U is a universal set as in the study of Venn diagrams.

Symmetric difference of sets A and B, denoted $A \triangle B$ or $A \ominus B$, is the set of all objects that are a member of exactly one of A and B (elements which are in one of the sets, but not in both). For instance, for the sets {1, 2, 3} and {2, 3, 4} , the symmetric difference set is {1, 4} . It is the set difference of the union and the intersection, $(A \cup B) \setminus (A \cap B)$ or $(A \setminus B) \cup (B \setminus A)$.

Power set of a set A is the set whose members are all of the possible subsets of A. For example, the power set of {1, 2} is { {}, {1}, {2}, {1, 2} } .
S

ALGORITHM:-
1.INSERT IN SET
```
int insert(int *arr,int size){
     cout<<"Enter the number to be inserted : ";
     size++;
```

```
        cin>>arr[size-1];
        return size;
}
```

2.REMOVE ELEMENT FROM SET

```
int remove(int *arr,int size){
        cout<<"Enter the position of number to be removed : ";
        int pos;
        cin>>pos;
        for(int i = pos-1;i<size-1;i++){
            arr[i]=arr[i+1];
        }
        size--;
        return size;
}
```

3.SEARCH A ELMENT FROM SET

```
void contains(int * arr, int size){
            cout<<"Enter the number to be searched : "<<endl;
            int no,counter=0;
            cin>>no;
            for(int i=0;i<size;i++){
                if(arr[i]==no){
                    cout<<"The number "<<no<<" is present at position
"<<i+1<<endl;
                    counter++;
                }
            }
            if(counter==0){cout<<"Number you entered is not present in
the set."<<endl;}
}
```

4.SIZE OF SET

```
void size(int size1,int size2){
        cout<<"The size of Set 1 = "<<size1<<endl;
        cout<<"The size of Set 2 = "<<size2<<endl;
}
```

5.DISPLAY THE SET

```
void display(int * arr1,int *arr2,int size1,int size2){
        for(int i=0;i<size1;i++){
            cout<<arr1[i]<<" ";
        }
```

```cpp
        cout<<endl;
         for(int i=0;i<size2;i++){
              cout<<arr2[i]<<" ";
        }
        cout<<endl;

}
```

6.UNION OF 2 SET

```cpp
void _union(int * arr1,int *arr2,int size1,int size2){

        cout<<"Union is ";
        for(int i=0;i<size1;i++){
              cout<<arr1[i]<<" ";
        }
        for(int i=0;i<size2;i++){
              bool isSame=false;
              for(int j=0;j<size1;j++){
                    if(arr2[i]==arr1[j]){
                          isSame=true;
                    }
              }
              if(isSame==false){
                    cout<<arr2[i]<<"   ";
              }
        }

        cout<<endl;
}
```

7.INTERSECTION OF 2 SET

```cpp
void intersection(int * arr1,int *arr2,int size1,int size2){
        cout<<"intersection is : ";

        for(int i=0;i<size1;i++){
              for(int j=0;j<size2;j++){
                          if(arr1[i]==arr2[j]){
                                cout<<arr1[i]<<" ";
                          }
              }
        }
        cout<<endl;
```

```
}
```

8.DIFFERENCE OF 2 SET

```
void diff(int * arr1,int *arr2,int size1,int size2){
        cout<<"Difference is : ";
        for(int i=0;i<size1;i++){
                bool isDiff=true;
                for(int j=0;j<size2;j++){
                        if(arr1[i]==arr2[j]){
                                isDiff=false;
                        }
                }
                if(isDiff==true){
                        cout<<arr1[i]<<"   ";
                }

        }
        cout<<endl;
}
```

9.SUBSET

```
void isSubset(int * arr1,int *arr2,int size1,int size2){
        if(size2>size1){
                cout<<"set 2 is not a subset"<<endl;
        }
        else{
                int k=0;
                for(int i=0;i<size2;i++){
                        bool isMatched = false;
                        for(int j=0;j<size1;j++){
                                if(arr2[i]==arr1[j]){
                                        isMatched=true;
                                        k++;
                                }
                        }
                        if(isMatched==false){
                                cout<<"set 2 is not a subset"<<endl;
                        }
                }
                if(k==size2){
                        cout<<"set 2 is a subset of set 1"<<endl;
```

```
                }

        }
}
```

CODE:-

```cpp
#include<iostream>
using namespace std;

int insert(int *arr,int size){
    cout<<"Enter the number to be inserted : ";
    size++;
    cin>>arr[size-1];
    return size;
}
int remove(int *arr,int size){
    cout<<"Enter the position of number to be removed : ";
    int pos;
    cin>>pos;
    for(int i = pos-1;i<size-1;i++){
        arr[i]=arr[i+1];
    }
    size--;
    return size;
}

void contains(int * arr, int size){
        cout<<"Enter the number to be searched : "<<endl;
        int no,counter=0;
        cin>>no;
        for(int i=0;i<size;i++){
            if(arr[i]==no){
                cout<<"The number "<<no<<" is present at position "<<i+1<<endl;
                counter++;
            }
        }
```

```cpp
            if(counter==0){cout<<"Number you entered is not present in
the set."<<endl;}
}
void size(int size1,int size2){
     cout<<"The size of Set 1 = "<<size1<<endl;
     cout<<"The size of Set 2 = "<<size2<<endl;
}

void display(int * arr1,int *arr2,int size1,int size2){
     for(int i=0;i<size1;i++){
          cout<<arr1[i]<<" ";
     }
     cout<<endl;
      for(int i=0;i<size2;i++){
          cout<<arr2[i]<<" ";
     }
     cout<<endl;


}

void _union(int * arr1,int *arr2,int size1,int size2){

     cout<<"Union is ";
     for(int i=0;i<size1;i++){
          cout<<arr1[i]<<" ";
     }
     for(int i=0;i<size2;i++){
          bool isSame=false;
          for(int j=0;j<size1;j++){
               if(arr2[i]==arr1[j]){
                    isSame=true;
               }
          }
          if(isSame==false){
               cout<<arr2[i]<<"    ";
          }
     }

     cout<<endl;
}
```

```cpp
void intersection(int * arr1,int *arr2,int size1,int size2){
     cout<<"intersection is : ";

     for(int i=0;i<size1;i++){
          for(int j=0;j<size2;j++){
                    if(arr1[i]==arr2[j]){
                              cout<<arr1[i]<<" ";
                    }
          }
     }
     cout<<endl;

}

void diff(int * arr1,int *arr2,int size1,int size2){
     cout<<"Difference is : ";
     for(int i=0;i<size1;i++){
          bool isDiff=true;
          for(int j=0;j<size2;j++){
                    if(arr1[i]==arr2[j]){
                              isDiff=false;
                    }
          }
          if(isDiff==true){
               cout<<arr1[i]<<"    ";
          }

     }
     cout<<endl;
}

void isSubset(int * arr1,int *arr2,int size1,int size2){
     if(size2>size1){
          cout<<"set 2 is not a subset"<<endl;
     }
     else{
          int k=0;
          for(int i=0;i<size2;i++){
               bool isMatched = false;
```

```cpp
                for(int j=0;j<size1;j++){
                    if(arr2[i]==arr1[j]){
                        isMatched=true;
                        k++;
                    }
                }
                if(isMatched==false){
                    cout<<"set 2 is not a subset"<<endl;
                }
            }
            if(k==size2){
                cout<<"set 2 is a subset of set 1"<<endl;
            }

        }
}

int main(){
    cout<<"How many elements do you want to enter in first set :";
    int size1;
    cin>>size1;
    int arr1[100];
    cout<<"Enter the elements below :";
    for(int i=0;i<size1;i++){
        cin>>arr1[i];
    }
    cout<<"How many elements do you want to enter in second set :";
    int size2;
    cin>>size2;
    int arr2[100];
    cout<<"Enter the elements below :";
    for(int i=0;i<size2;i++){
        cin>>arr2[i];
    }
    char choice;
    do{
        cout<<"      M E N U     "<<endl;
        cout<<"1.insert"<<endl<<"2.remove"<<endl<<"3.contains
element?"<<endl<<"4.size"<<endl<<"5.display"<<endl<<"6.intersection"
<<endl<<"7.Union"<<endl<<"8.Difference"<<endl<<"9.is Subset "<<endl;
```

```
            int ch;
            cin>>ch;
            switch(ch){
                case 1 :
                    cout<<"In which set do you want to insert the
element ? "<<endl;
                    cout<<"1.set 1"<<endl<<"2.set 2"<<endl;
                    int no2;
                    cin>>no2;
                    if(no2==1){size1 = insert(arr1,size1);}
                    else{size2 = insert(arr2,size2);}
                break;
                case 2 : cout<<"from which set do you want to remove
the element ? "<<endl;
                    cout<<"1.set 1"<<endl<<"2.set 2"<<endl;
                    int no3;
                    cin>>no3;
                    if(no3==1){size1 = remove(arr1,size1);}
                    else{size2 = remove(arr2,size2);}
                break;
                case 3 :
                    cout<<"In which set do you want to search the
element ? "<<endl;
                    cout<<"1.set 1"<<endl<<"2.set 2"<<endl;
                    int no4;
                    cin>>no4;
                    if(no4==1){contains(arr1,size1);}
                    else{contains(arr2,size2);}
                break;
                case 4 : size(size1,size2);
                break;
                case 5 : display(arr1,arr2,size1,size2);
                break;
                case 6 : intersection(arr1,arr2,size1,size2);
                break;
                case 7 : _union(arr1,arr2,size1,size2);
                break;
                case 8 : diff(arr1,arr2,size1,size2);
                break;
                case 9 : isSubset(arr1,arr2,size1,size2);
```

```
                break;
                default : "Sorry!! Wrong choice.";    }
            cout<<"Do you want to continue?[y/n]";
            cin>>choice;
        }while(choice=='Y' || choice=='y');
        return 0;
}
```

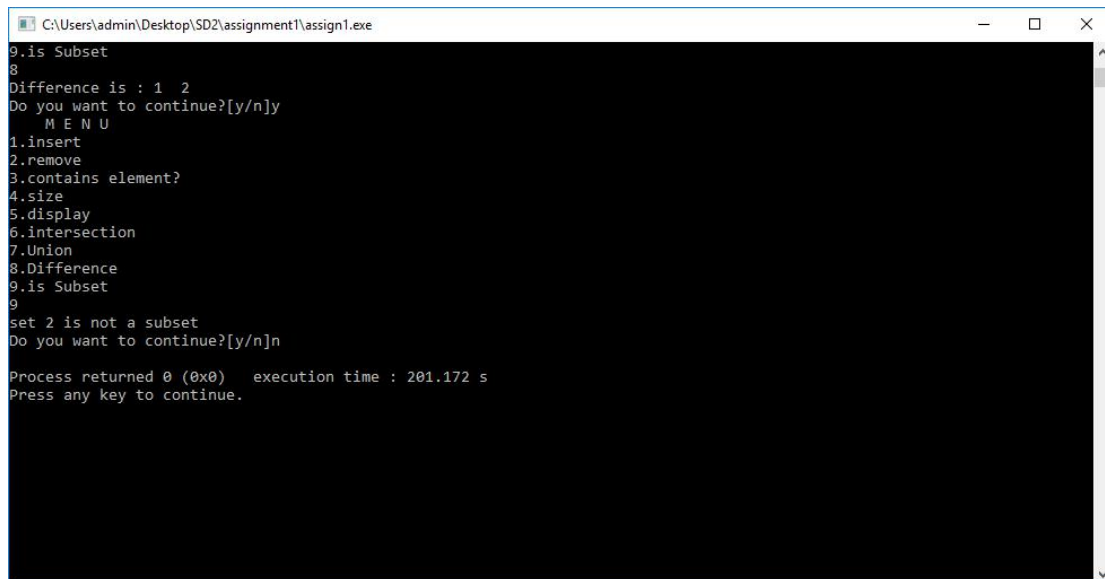**OUTPUT:-**

```
C:\Users\admin\Desktop\SD2\assignment1\assign1.exe                    —    □    ×
3.contains element?
4.size
5.display
6.intersection
7.Union
8.Difference
9.is Subset
3
In which set do you want to search the element ?
1.set 1
2.set 2
1
Enter the number to be searched :
4
The number 4 is present at position 4
Do you want to continue?[y/n]y
    M E N U
1.insert
2.remove
3.contains element?
4.size
5.display
6.intersection
7.Union
8.Difference
9.is Subset
4
The size of Set 1 = 4
The size of Set 2 = 3
Do you want to continue?[y/n]y
```

```
C:\Users\admin\Desktop\SD2\assignment1\assign1.exe                    —    □    ×
1.insert
2.remove
3.contains element?
4.size
5.display
6.intersection
7.Union
8.Difference
9.is Subset
5
1 2 3 4
3 4 5
Do you want to continue?[y/n]y
    M E N U
1.insert
2.remove
3.contains element?
4.size
5.display
6.intersection
7.Union
8.Difference
9.is Subset
6
intersection is : 3 4
Do you want to continue?[y/n]y
    M E N U
1.insert
2.remove
3.contains element?
```

```
C:\Users\admin\Desktop\SD2\assignment1\assign1.exe                    —    □    ×
2.remove
3.contains element?
4.size
5.display
6.intersection
7.Union
8.Difference
9.is Subset
7
Union is 1 2 3 4 5
Do you want to continue?[y/n]y
    M E N U
1.insert
2.remove
3.contains element?
4.size
5.display
6.intersection
7.Union
8.Difference
9.is Subset
8
Difference is : 1  2
Do you want to continue?[y/n]y
    M E N U
1.insert
2.remove
3.contains element?
4.size
5.display
```

CONCLUSION:-

We have successfully implemented set concept and related functions.