```
Set 1
Enter an element: 12
Enter an element: 34
Enter an element: 80
Enter an element: 12
Enter an element: -999

The set 1 is = {12, 34, 80}

Set 2
Enter an element: 16
Enter an element: 12
Enter an element: 8
Enter an element: -999

The set2 is = {16, 12, 8}

Set1 is not equal to Set2

The union is = {12, 34, 80, 16, 8} // order doesnt matter here
The intersection is = {12}

set1 - Set2 = {34, 80}
Set2 - Set1 = {16, 8}

Enter an element for inclusion test: 12
12 is in set1
Enter an element for inclusion test: 16
16 is in set2
```

==========

Grading Guidelines

1. Your set method must work with any set.
2. The maximum points for a program that has the wrong output is: 90%
3. The maximum points for a program that doesnt compile is: 70%
4. To receive maximum points all of the following methods must be successfully implemented.
   - Set-default constructor creates an empty set
   - add - adds an element to the set
   - contains - This method searches the set for the a specified element. It returns true if this array contains the specified element.
   - complement - This method returns the complement - For example A.complement(B) returns a set that contains elements of A
     that are not in B.
   - equals - returns true if two sets are equal.
   - intersection - This method returns the intersection of two sets - i.e the set of all elements that are in both A and B
   - remove - This method removes and element from the set. The method returns false, if the element doesnt exists.
   - union - returns the union of two sets
   - print - prints the set as a set. eg: {1, 2, 3}
5. Test your main method with multiple inputs to make sure the program works properly.