CS-211 : Discrete Math

Assignment 1

Set Operations :

Name : Mohamed Mashaal Mohamed Ali El-agha - No Team members ..

- **Assignment Implementation :**

Code and Implementation can be found over here :
https://github.com/MohamedMashaal/Set-Operation-Simulator

- **Problem Statement :**

  Write a program that takes an input a list of strings as a Universe, then takes another input a number of sets (that are subsets of the universe) then ask the user about the operations they want to perform:

  • Union of two sets

  • Intersection of two sets

  • Complement of a set

- **Used data structures :**

  - Java ArrayList .

- **Algorithms used documented :**

-Union :

Method takes 2 sets and returns the union .

```
(ArrayList1<String> , ArrayList2<String>){
ArrayList<String> uniRes ---> ArrayList1;
      for(String x : ArrayList2){
         if(!uniRes.contains(x))
            uniRes.add(x);
      }
      return uniRes;
}
```

-Intersection :

Method takes 2 sets and returns the intersection .

```
(ArrayList1<String> , ArrayList2<String>){
ArrayList<String> intRes ;
      for(String x : ArrayList1){
         if(ArrayList2.contains(x))
            intRes.add(x);
      }
      return return intRes;
```

-Complement :

Method takes 1 sets and returns the complement .

```
(ArrayList){
ArrayList<String> compRes ----> Universe ;
      for(String x : ArrayList)
            compRes.remove(x);
      return compRes ;
```

- **Design and Assumption ：**

Though Problem Statement assumed that universe gets inserted at first then subsets then the user gets to specify what to do .
The Assignment was implemented using **Java Swing** for a GUI and as an **Android App** too to allow User more freedom ,more functionality and to be more user friendly .
The GUI and Event Driven Programming with The ArrayLists allowed the user a more Dynamic Sets , User is allowed to add elements to the universe or the subsets whenever he requires it plus executing whatever functionality (union , intersection , complement) whenever he wants .

User is given the freedom to add duplicated elements to any set however any duplicate elements aren't considered as a check is being executed when the user adds an element to a specific set since sets don't allow repetition .
Set 0 is assumed to be the universe and subsets are indexed 1 , 2 , ... A label with such instruction has been set to inform the user of such thing .

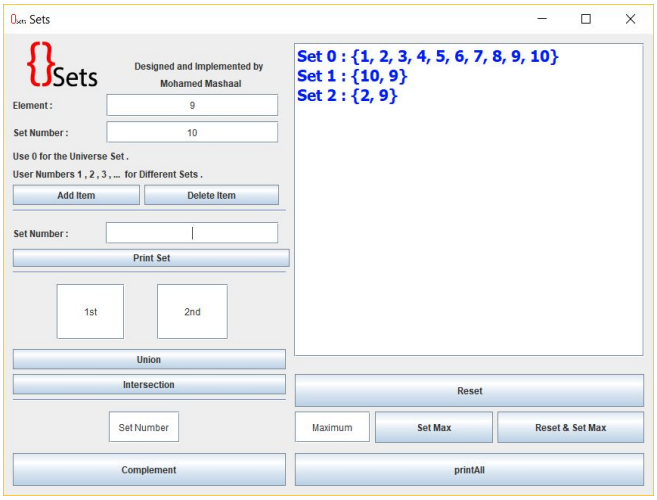
-- The Assignment was implemented at first using HashSets , then ArrayLists were used instead .(Both are included in the Assignment Implementation) .
P.S : The Assignment could implemented with the Help of the Hint as an ArrayList of Strings representing the Universe then Subsets implemented Using ArrayLists of bits mapping to Those Strings , Union can be achieved by Oring (||) two Subsets while Intersection can be achieved by Anding (&&) two Subsets as for the complement it could be achieved by XORing (^) Universe with the Subset .

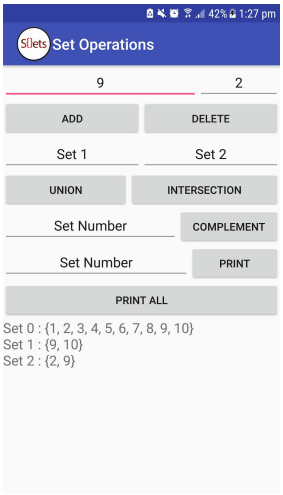
- **Sample Runs :**

- Set 0 : Represents the Universe {1,2,3,4,5,6,7,8,9,10}

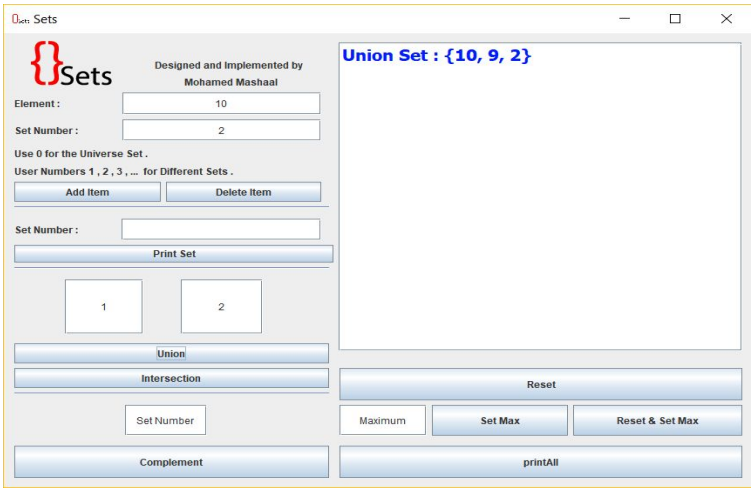
Set1 : {9,10}

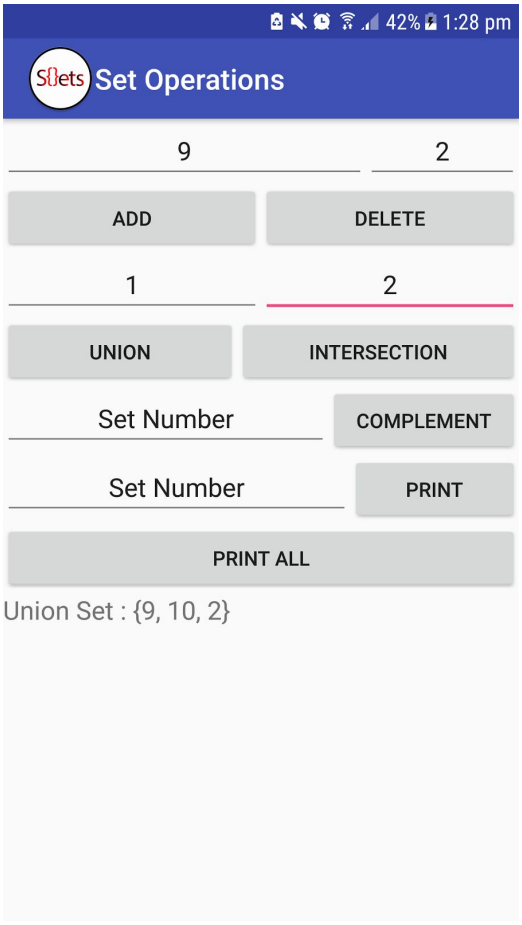
Set2: {2,9}



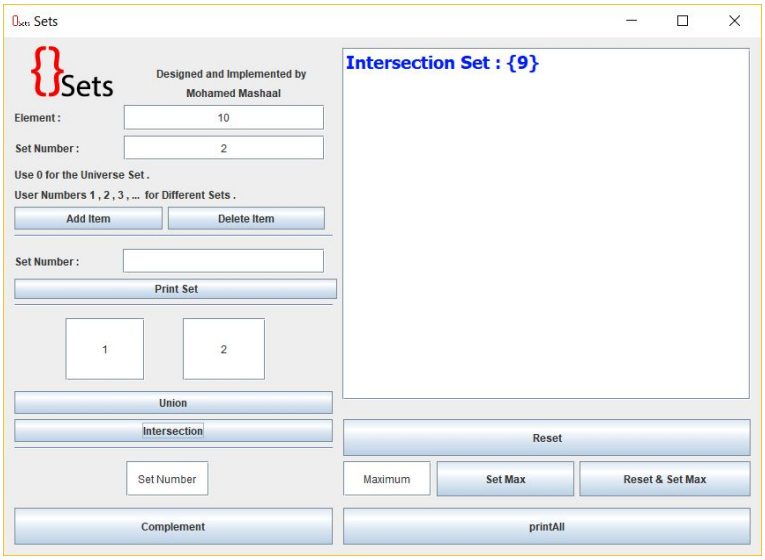


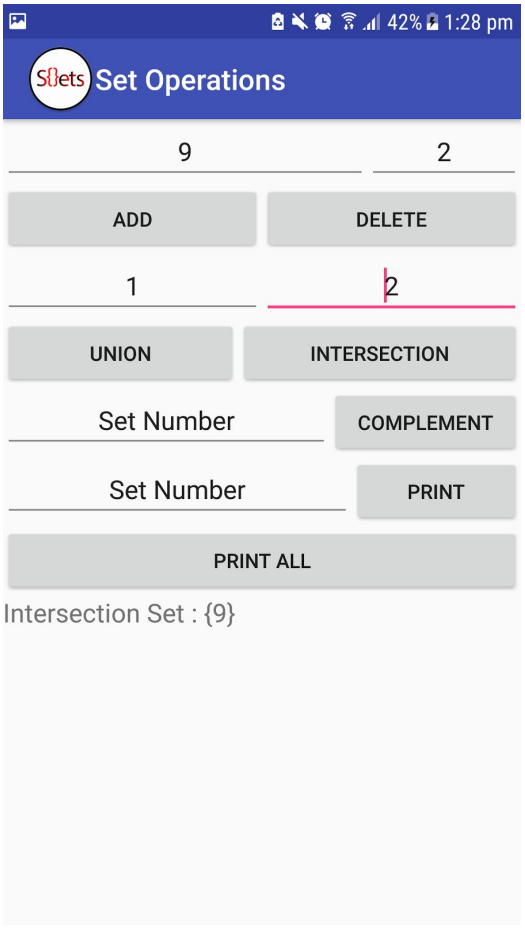
Set 1 & Set 2 Union : {2 , 9 , 10 }





Set 1 & Set 2 Intersection : {9}





Set 1 Complement : {1,2,3,4,5,6,7,8}