DISCRETE MATH

LAB 1

Name: Salah El Din Ahmad
El Tenehy
ID: 19015854
Name: Gamal Abdul Hameed Nassef
Newysar
ID: 19015550

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 (3 required features to be implemented in this assignment):

- Union of two sets
- Intersection of two sets
- Complement of a set

Used Data Structures:

1. Arrays
2. Array Lists
3. Strings

Algorithms used documented using pseudo code:

Function Union:

```
1. arr ← sub1
2. Loop1: Iterate over sub2:
3.     f ← true
4.     Loop 2: Iterate over sub1:
5.         If the element of sub1 is the same element in sub2,
           f ← false
6.     End loop 2
7.     If the flag is true, add the element to arr
8. End loop 1
9. Return arr
```

Function Intersection:

1. arr ← new array list
2. Loop1: Iterate over sub1:
3. Loop2: Iterate over sub2:
4. If the element of sub1 is the same element in sub2, add it to arr
5. End loop 2
6. End loop 1
7. Return arr

Function Complement:

1. arr ← new array list
2. Loop1: Iterate over uni(the Universe Set):
3. f ← true
4. Loop2: Iterate over sub:
5. If the element of sub is the same element in uni, f ← false
6. End loop 2
7. If the flag is true, add the element of the uni to arr
8. End loop 1
9. Return arr

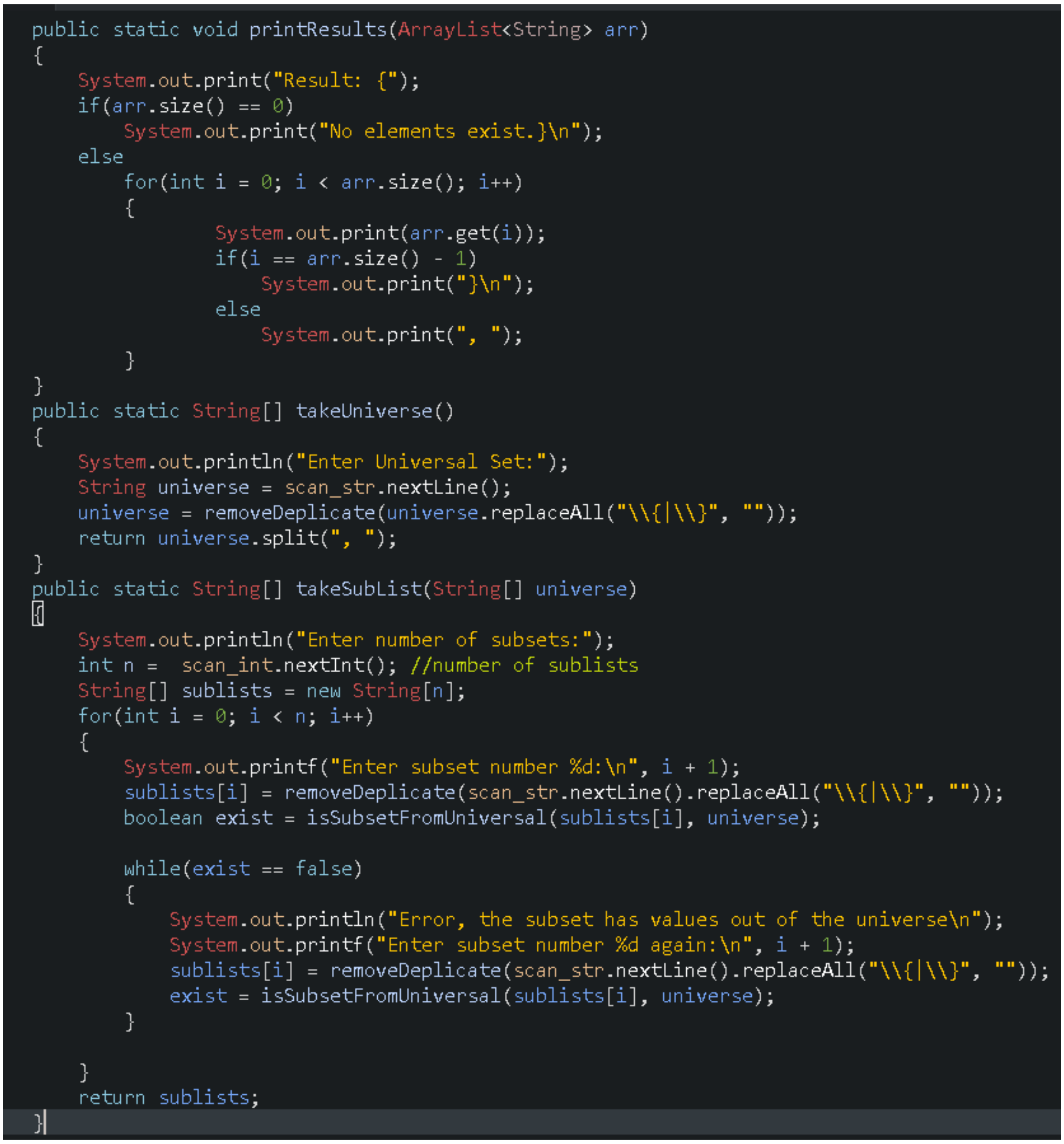
Function removeDulpicate:

1. Scan the Universal set
2. Loop1: Iterate over the set from index j = 0 to length
3. Loop2: Iterate over the set from index k = j + 1 to length
4. If(set[j] is the same as set[k]), remove set[j]
5. End loop2
6. End loop1

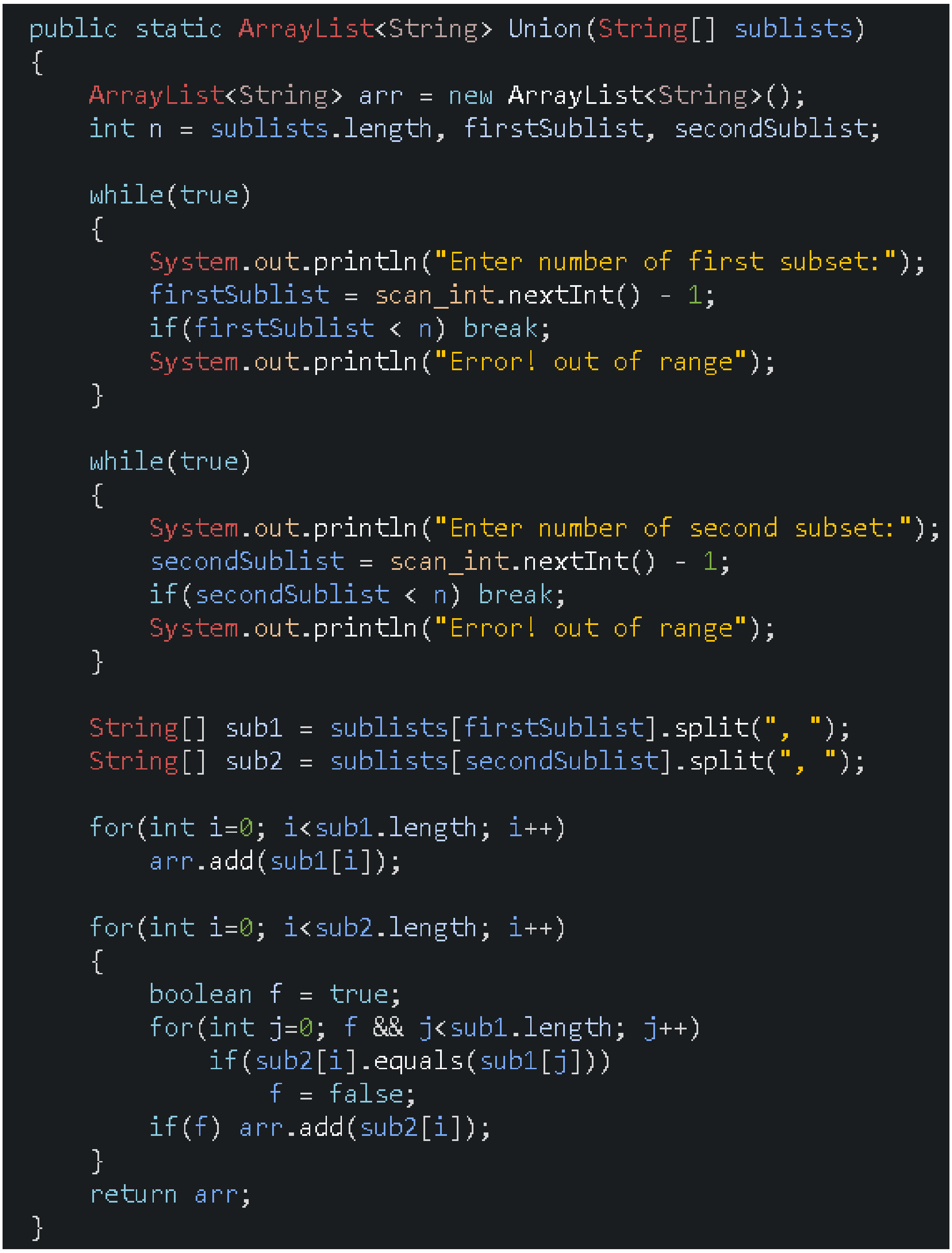
Function isSubsetFromUniversal:

```
Loop1: Iterate over the sublist
F ← false
Loop2: Iterate over the Universal
If(set[j] is the same as set[k]), f ← true
End loop2
If(exist is false), return false
End loop1
Return true
```

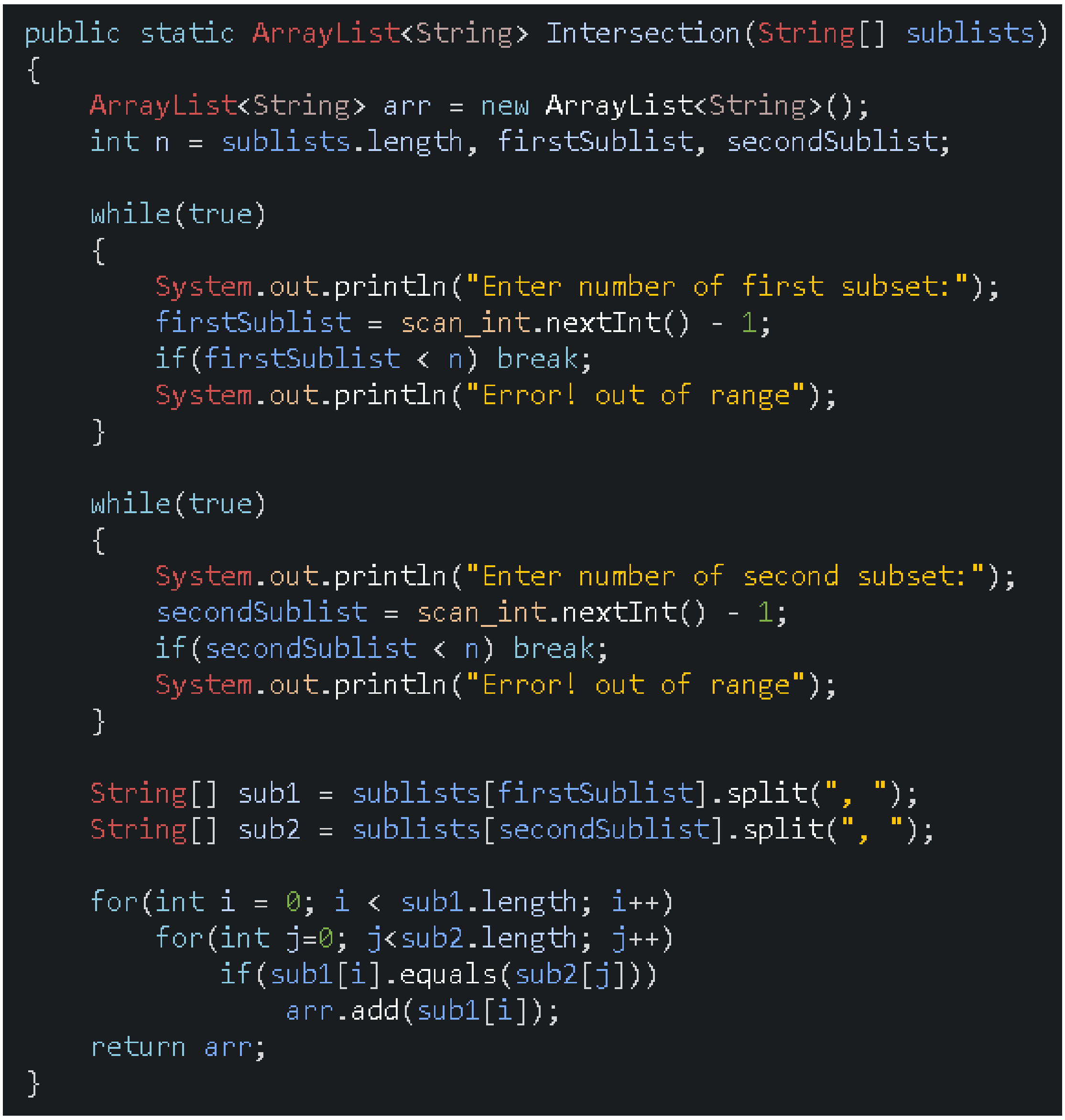
Code Snippets:



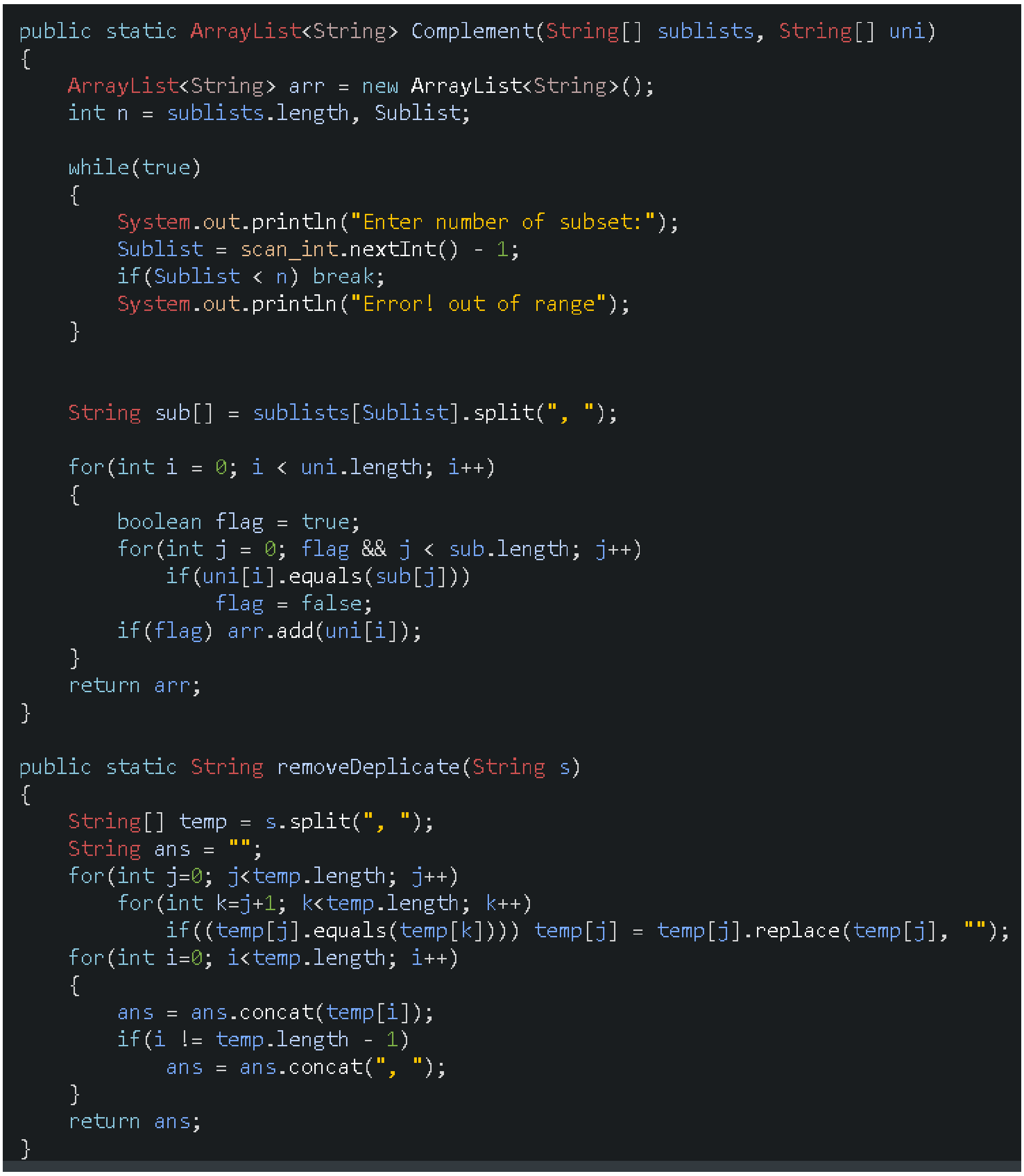
```java
public static void printResults(ArrayList<String> arr)
{
    System.out.print("Result: {");
    if(arr.size() == 0)
        System.out.print("No elements exist.}\n");
    else
        for(int i = 0; i < arr.size(); i++)
        {
                System.out.print(arr.get(i));
                if(i == arr.size() - 1)
                    System.out.print("}\n");
                else
                    System.out.print(", ");
        }
}
public static String[] takeUniverse()
{
    System.out.println("Enter Universal Set:");
    String universe = scan_str.nextLine();
    universe = removeDeplicate(universe.replaceAll("\\{|\\}", ""));
    return universe.split(", ");
}
public static String[] takeSubList(String[] universe)
{
    System.out.println("Enter number of subsets:");
    int n =  scan_int.nextInt(); //number of sublists
    String[] sublists = new String[n];
    for(int i = 0; i < n; i++)
    {
        System.out.printf("Enter subset number %d:\n", i + 1);
        sublists[i] = removeDeplicate(scan_str.nextLine().replaceAll("\\{|\\}", ""));
        boolean exist = isSubsetFromUniversal(sublists[i], universe);

        while(exist == false)
        {
            System.out.println("Error, the subset has values out of the universe\n");
            System.out.printf("Enter subset number %d again:\n", i + 1);
            sublists[i] = removeDeplicate(scan_str.nextLine().replaceAll("\\{|\\}", ""));
            exist = isSubsetFromUniversal(sublists[i], universe);
        }

    }
    return sublists;
}
```



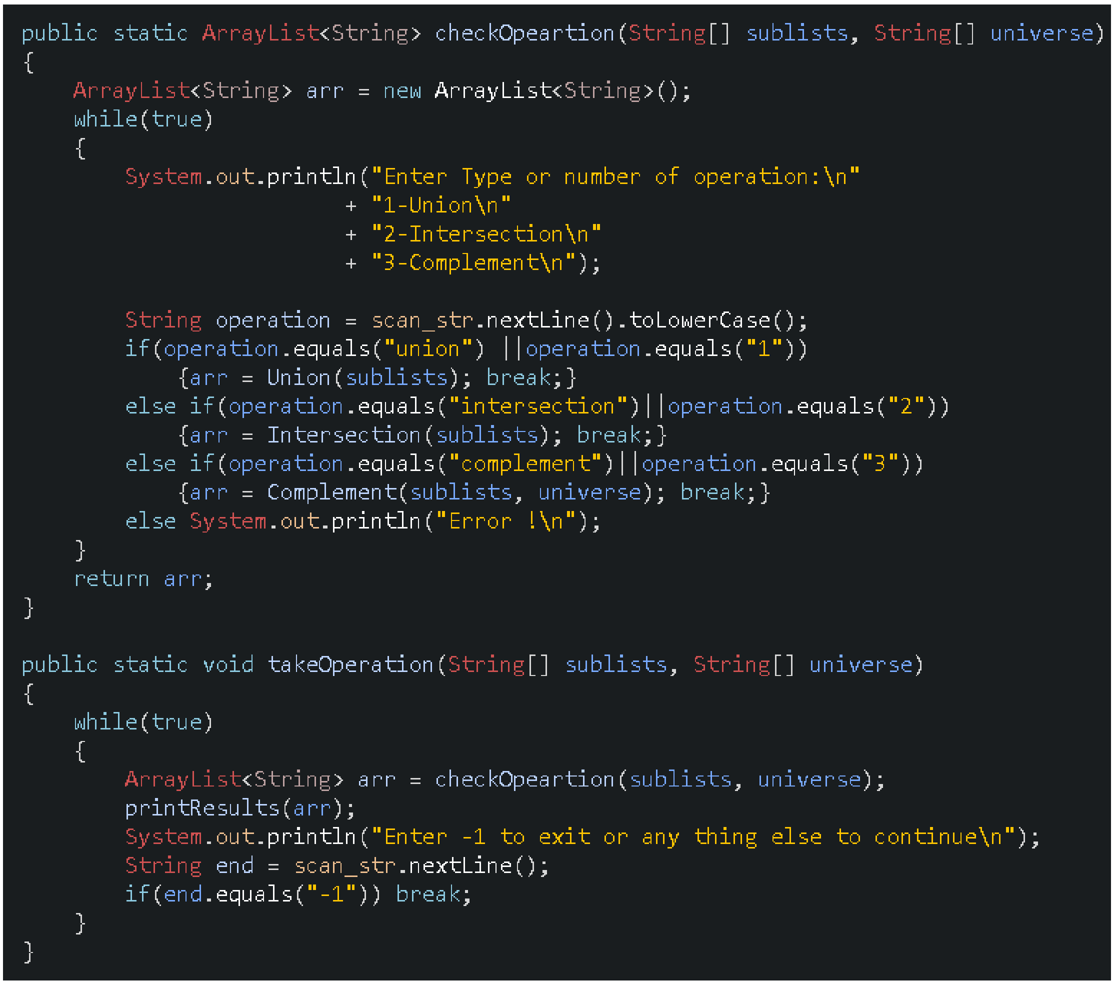
```
public static ArrayList<String> Union(String[] sublists)
{
    ArrayList<String> arr = new ArrayList<String>();
    int n = sublists.length, firstSublist, secondSublist;

    while(true)
    {
        System.out.println("Enter number of first subset:");
        firstSublist = scan_int.nextInt() - 1;
        if(firstSublist < n) break;
        System.out.println("Error! out of range");
    }

    while(true)
    {
        System.out.println("Enter number of second subset:");
        secondSublist = scan_int.nextInt() - 1;
        if(secondSublist < n) break;
        System.out.println("Error! out of range");
    }

    String[] sub1 = sublists[firstSublist].split(", ");
    String[] sub2 = sublists[secondSublist].split(", ");

    for(int i=0; i<sub1.length; i++)
        arr.add(sub1[i]);

    for(int i=0; i<sub2.length; i++)
    {
        boolean f = true;
        for(int j=0; f && j<sub1.length; j++)
            if(sub2[i].equals(sub1[j]))
                f = false;
        if(f) arr.add(sub2[i]);
    }
    return arr;
}
```



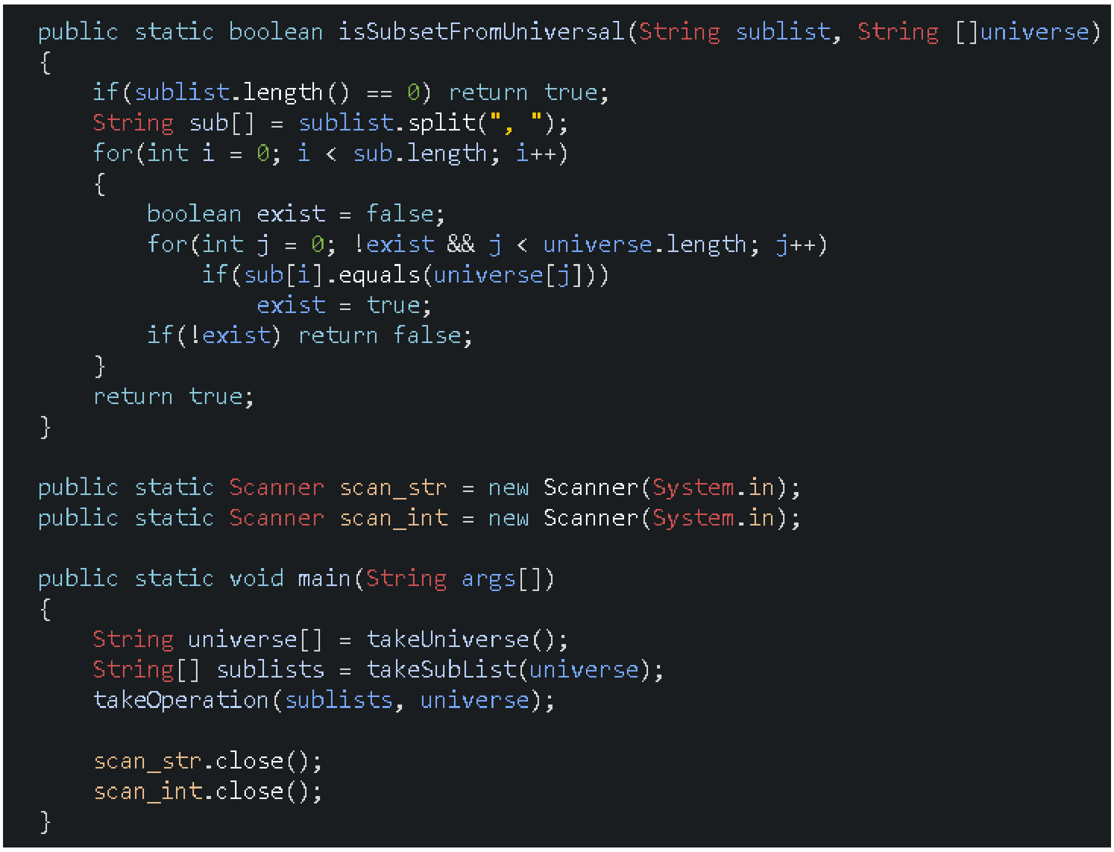
```java
public static ArrayList<String> Intersection(String[] sublists)
{
    ArrayList<String> arr = new ArrayList<String>();
    int n = sublists.length, firstSublist, secondSublist;

    while(true)
    {
        System.out.println("Enter number of first subset:");
        firstSublist = scan_int.nextInt() - 1;
        if(firstSublist < n) break;
        System.out.println("Error! out of range");
    }

    while(true)
    {
        System.out.println("Enter number of second subset:");
        secondSublist = scan_int.nextInt() - 1;
        if(secondSublist < n) break;
        System.out.println("Error! out of range");
    }

    String[] sub1 = sublists[firstSublist].split(", ");
    String[] sub2 = sublists[secondSublist].split(", ");

    for(int i = 0; i < sub1.length; i++)
        for(int j=0; j<sub2.length; j++)
            if(sub1[i].equals(sub2[j]))
                arr.add(sub1[i]);
    return arr;
}
```



```
public static ArrayList<String> Complement(String[] sublists, String[] uni)
{
    ArrayList<String> arr = new ArrayList<String>();
    int n = sublists.length, Sublist;

    while(true)
    {
        System.out.println("Enter number of subset:");
        Sublist = scan_int.nextInt() - 1;
        if(Sublist < n) break;
        System.out.println("Error! out of range");
    }


    String sub[] = sublists[Sublist].split(", ");

    for(int i = 0; i < uni.length; i++)
    {
        boolean flag = true;
        for(int j = 0; flag && j < sub.length; j++)
            if(uni[i].equals(sub[j]))
                flag = false;
        if(flag) arr.add(uni[i]);
    }
    return arr;
}

public static String removeDeplicate(String s)
{
    String[] temp = s.split(", ");
    String ans = "";
    for(int j=0; j<temp.length; j++)
        for(int k=j+1; k<temp.length; k++)
            if((temp[j].equals(temp[k]))) temp[j] = temp[j].replace(temp[j], "");
    for(int i=0; i<temp.length; i++)
    {
        ans = ans.concat(temp[i]);
        if(i != temp.length - 1)
            ans = ans.concat(", ");
    }
    return ans;
}
```

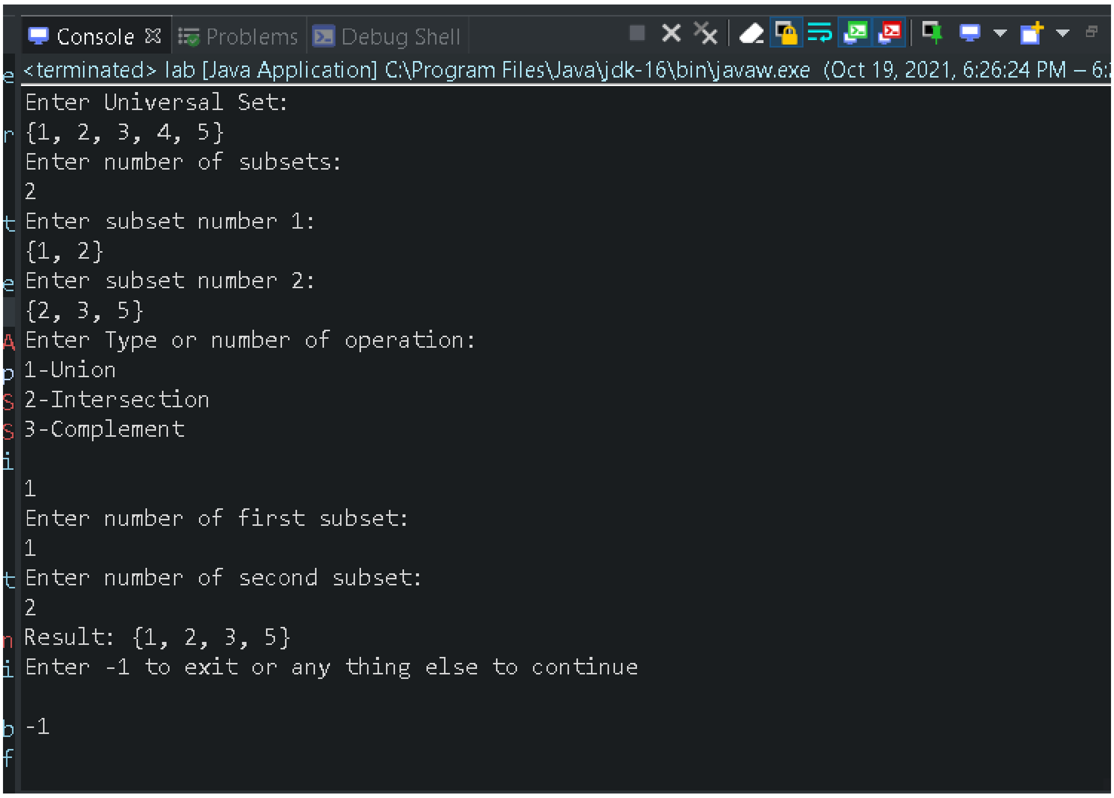


```
public static ArrayList<String> checkOpeartion(String[] sublists, String[] universe)
{
    ArrayList<String> arr = new ArrayList<String>();
    while(true)
    {
        System.out.println("Enter Type or number of operation:\n"
                         + "1-Union\n"
                         + "2-Intersection\n"
                         + "3-Complement\n");

        String operation = scan_str.nextLine().toLowerCase();
        if(operation.equals("union") ||operation.equals("1"))
            {arr = Union(sublists); break;}
        else if(operation.equals("intersection")||operation.equals("2"))
            {arr = Intersection(sublists); break;}
        else if(operation.equals("complement")||operation.equals("3"))
            {arr = Complement(sublists, universe); break;}
        else System.out.println("Error !\n");
    }
    return arr;
}

public static void takeOperation(String[] sublists, String[] universe)
{
    while(true)
    {
        ArrayList<String> arr = checkOpeartion(sublists, universe);
        printResults(arr);
        System.out.println("Enter -1 to exit or any thing else to continue\n");
        String end = scan_str.nextLine();
        if(end.equals("-1")) break;
    }
}
```

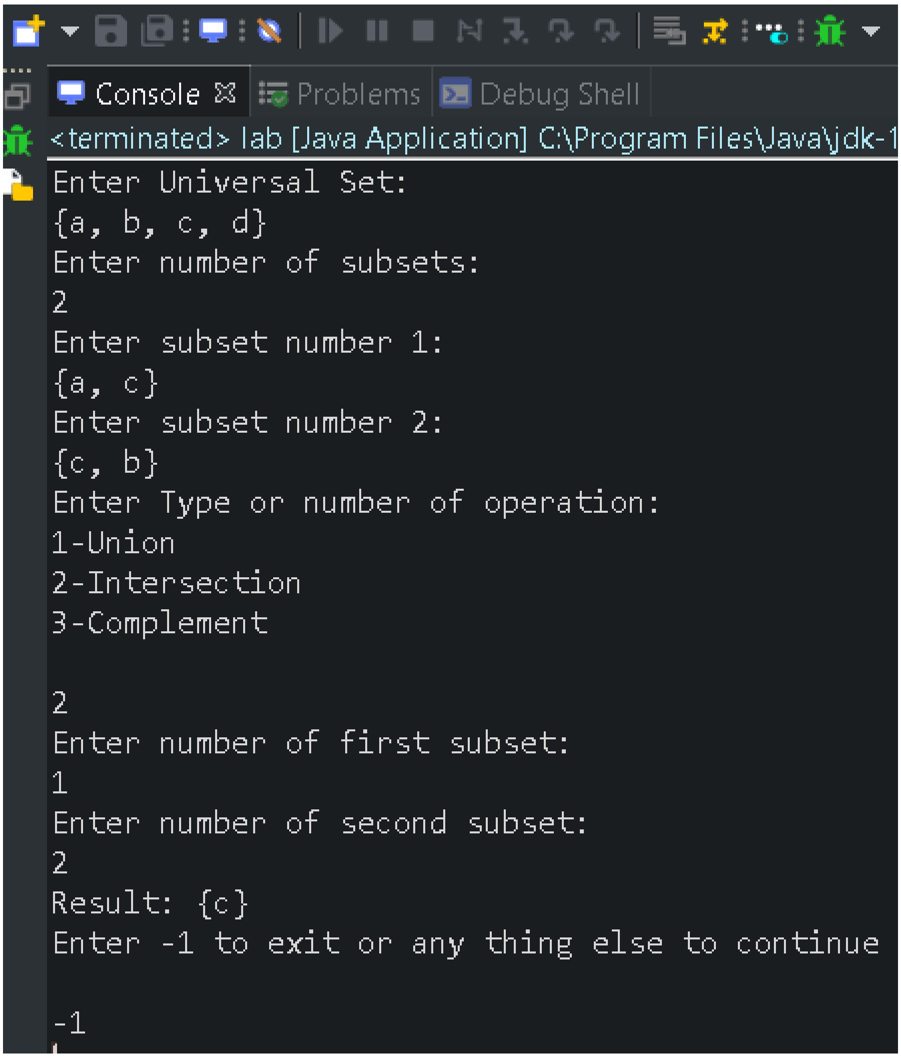


```
public static boolean isSubsetFromUniversal(String sublist, String []universe)
{
    if(sublist.length() == 0) return true;
    String sub[] = sublist.split(", ");
    for(int i = 0; i < sub.length; i++)
    {
        boolean exist = false;
        for(int j = 0; !exist && j < universe.length; j++)
            if(sub[i].equals(universe[j]))
                exist = true;
        if(!exist) return false;
    }
    return true;
}

public static Scanner scan_str = new Scanner(System.in);
public static Scanner scan_int = new Scanner(System.in);

public static void main(String args[])
{
    String universe[] = takeUniverse();
    String[] sublists = takeSubList(universe);
    takeOperation(sublists, universe);

    scan_str.close();
    scan_int.close();
}
```

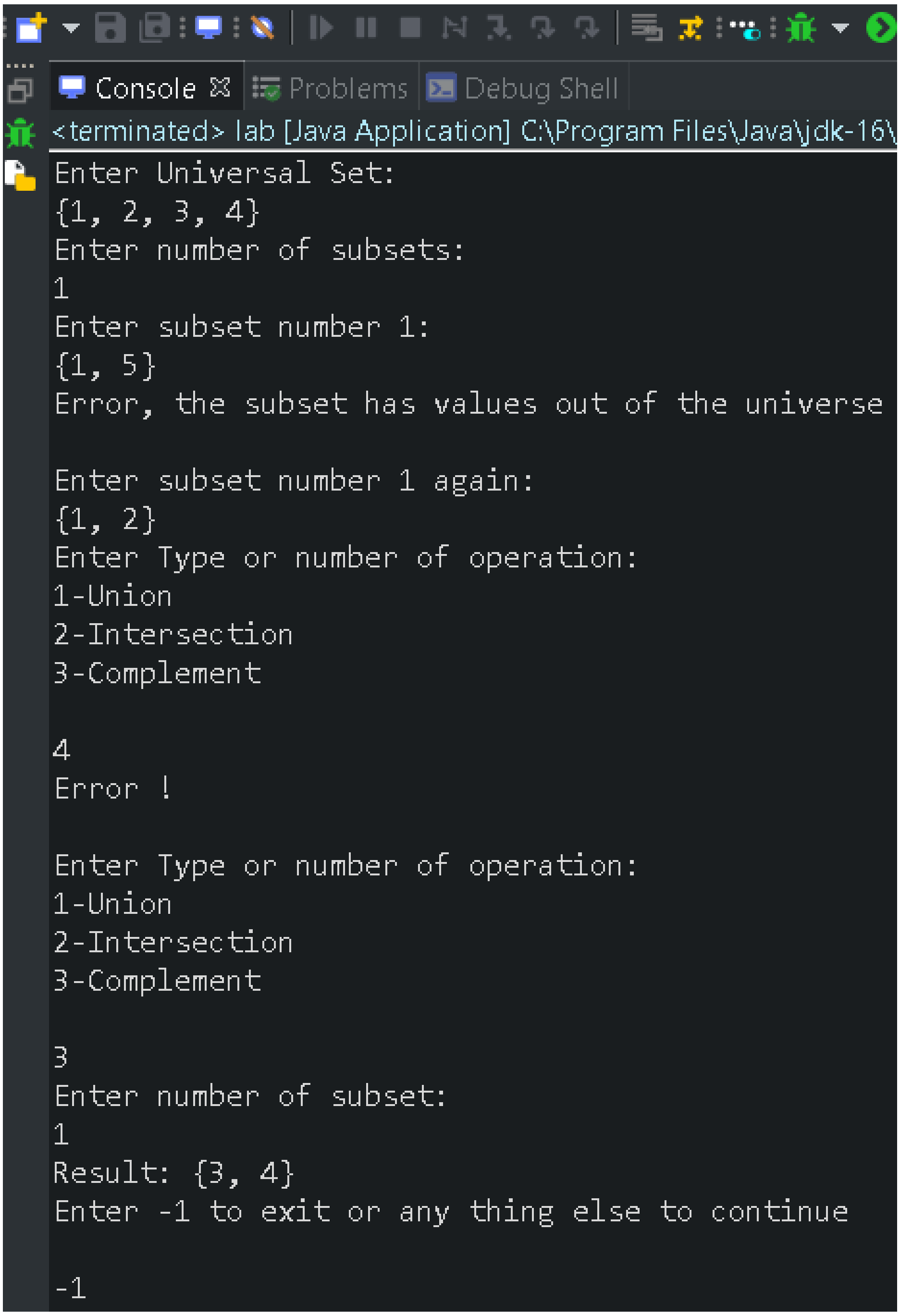
Sample runs:



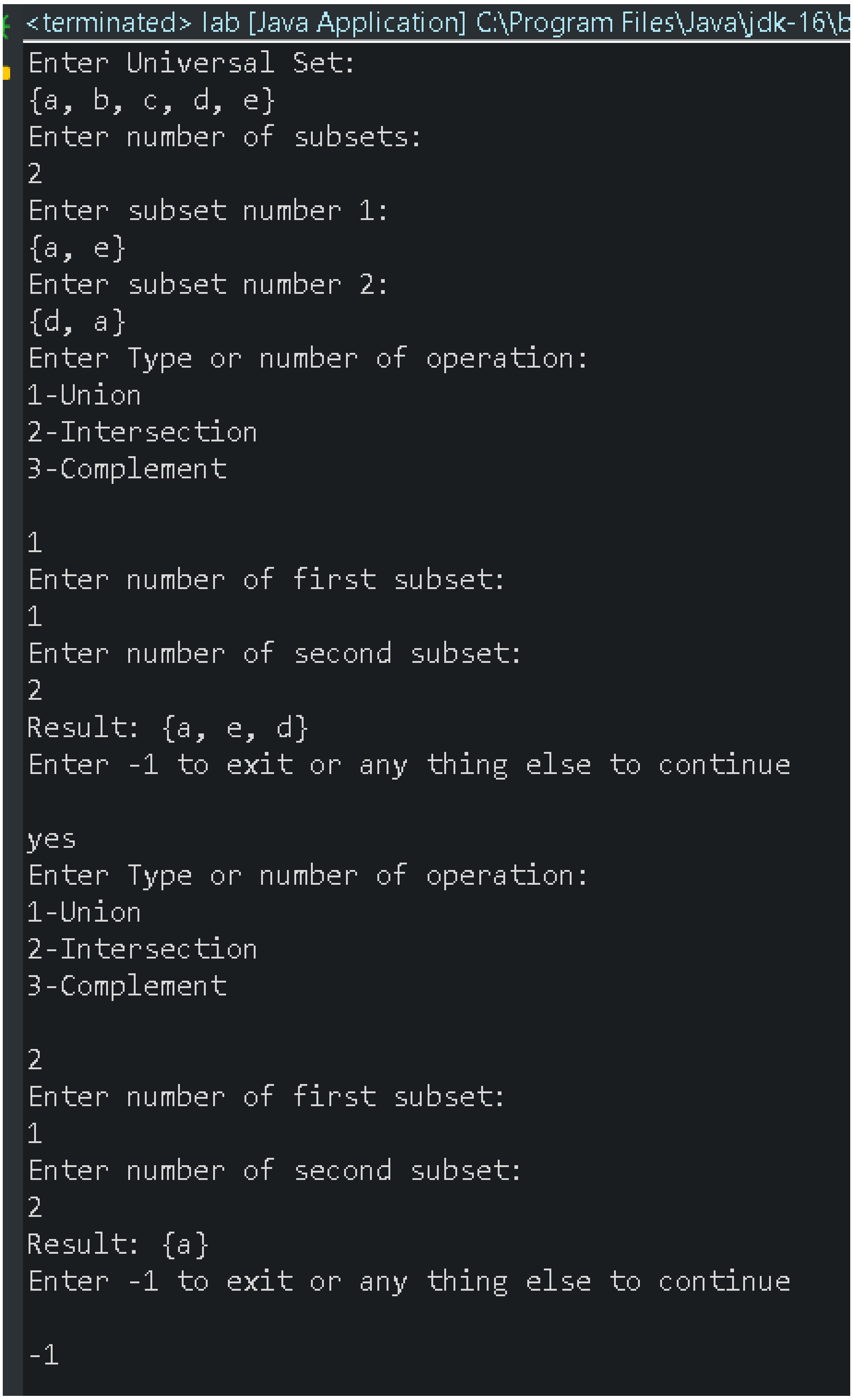
```
<terminated> lab [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (Oct 19, 2021, 6:26:24 PM – 6:
Enter Universal Set:
{1, 2, 3, 4, 5}
Enter number of subsets:
2
Enter subset number 1:
{1, 2}
Enter subset number 2:
{2, 3, 5}
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

1
Enter number of first subset:
1
Enter number of second subset:
2
Result: {1, 2, 3, 5}
Enter -1 to exit or any thing else to continue

-1
```



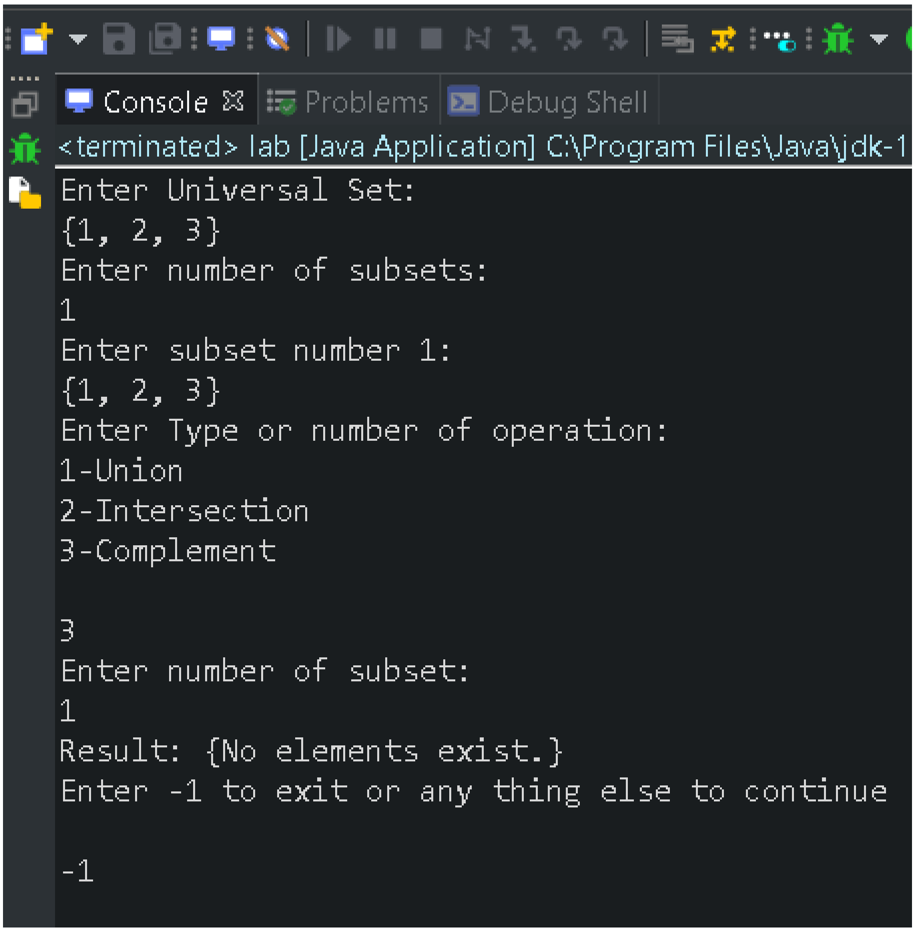
```
<terminated> lab [Java Application] C:\Program Files\Java\jdk-1
Enter Universal Set:
{a, b, c, d}
Enter number of subsets:
2
Enter subset number 1:
{a, c}
Enter subset number 2:
{c, b}
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

2
Enter number of first subset:
1
Enter number of second subset:
2
Result: {c}
Enter -1 to exit or any thing else to continue

-1
```



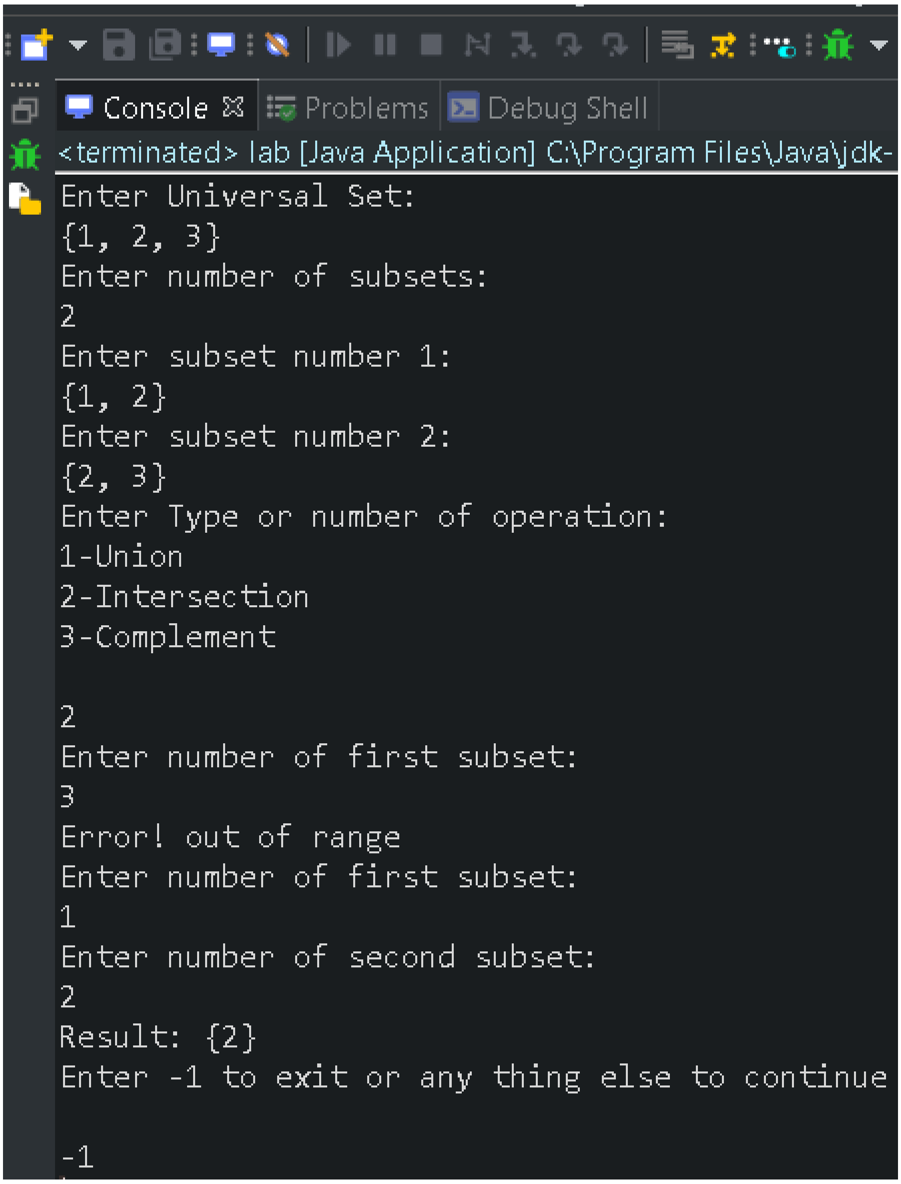
```
<terminated> lab [Java Application] C:\Program Files\Java\jdk-16\
Enter Universal Set:
{1, 2, 3, 4}
Enter number of subsets:
1
Enter subset number 1:
{1, 5}
Error, the subset has values out of the universe

Enter subset number 1 again:
{1, 2}
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

4
Error !

Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

3
Enter number of subset:
1
Result: {3, 4}
Enter -1 to exit or any thing else to continue

-1
```



```
<terminated> lab [Java Application] C:\Program Files\Java\jdk-16\b
Enter Universal Set:
{a, b, c, d, e}
Enter number of subsets:
2
Enter subset number 1:
{a, e}
Enter subset number 2:
{d, a}
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

1
Enter number of first subset:
1
Enter number of second subset:
2
Result: {a, e, d}
Enter -1 to exit or any thing else to continue

yes
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

2
Enter number of first subset:
1
Enter number of second subset:
2
Result: {a}
Enter -1 to exit or any thing else to continue

-1
```



```
<terminated> lab [Java Application] C:\Program Files\Java\jdk-1
Enter Universal Set:
{1, 2, 3}
Enter number of subsets:
1
Enter subset number 1:
{1, 2, 3}
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

3
Enter number of subset:
1
Result: {No elements exist.}
Enter -1 to exit or any thing else to continue

-1
```



<terminated> lab [Java Application] C:\Program Files\Java\jdk-

```
Enter Universal Set:
{1, 2, 3}
Enter number of subsets:
2
Enter subset number 1:
{1, 2}
Enter subset number 2:
{2, 3}
Enter Type or number of operation:
1-Union
2-Intersection
3-Complement

2
Enter number of first subset:
3
Error! out of range
Enter number of first subset:
1
Enter number of second subset:
2
Result: {2}
Enter -1 to exit or any thing else to continue

-1
```

Assumptions and details:

1. Enter the Universal Set between the curly braces{} and separate the elements with(,). for ex:{a, b, c, d}. If you enter the set between any other braces, the whole set will be considered a single element. For ex, [1, 2, 3] is the

same as {[1, 2, 3]} which is a set containing the element [1, 2, 3].

2. Enter the number of subsets you have.
3. Enter the subsets sequentially in the same form of the Universal Set.
4. If you enter a set containing elements out of the universal, it will print error and ask you to enter a correct set instead.
5. Enter the type(letter case doesn’t affect) or the number of the operation:
   1- Union
   2- Intersection
   3- Complement
6. If You enter a wrong name or wrong number, it will ask you again to enter a correct one.
7. Enter the number of the subsets to work on, for ex, to get the union of the first and the third sets, enter 1 then 3.
8. After the Result appears, Enter (-1) to exit or any thing else to continue.