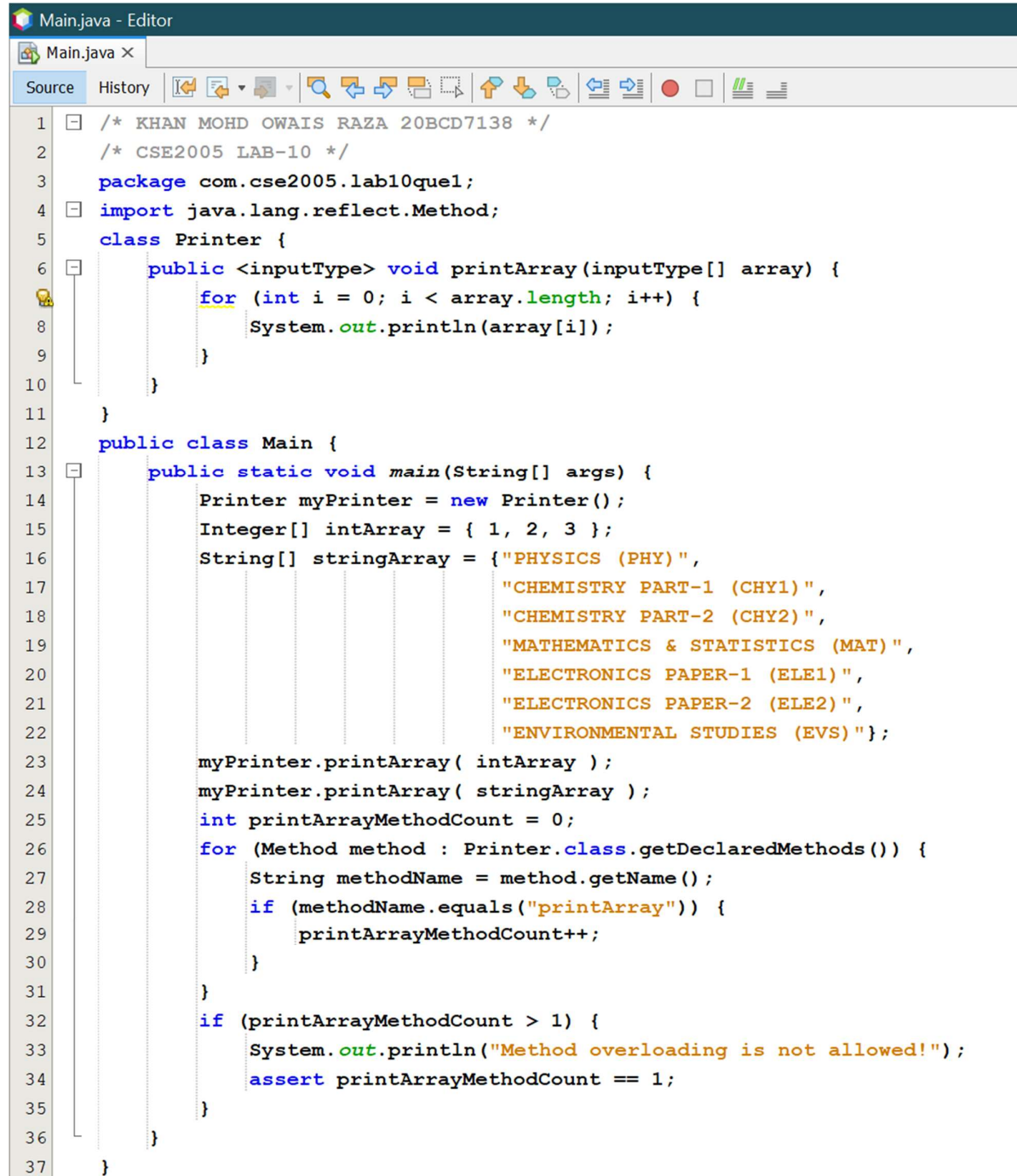


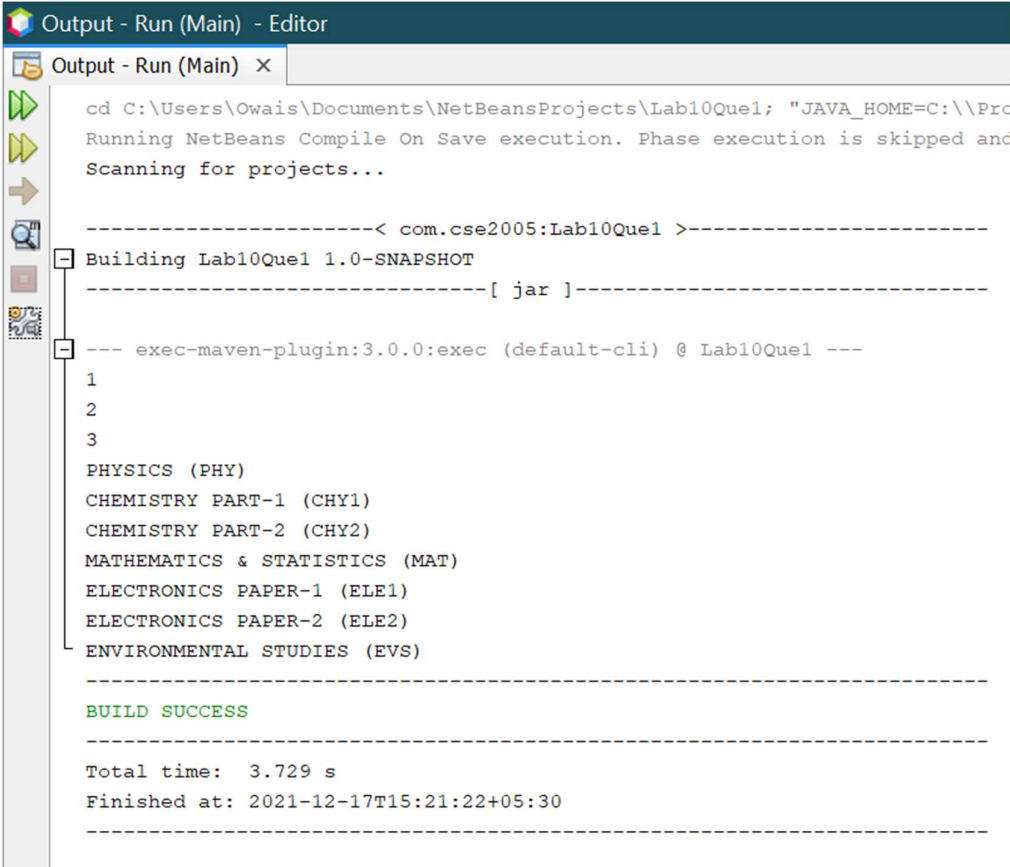
KHAN MOHD OWAIS RAZA
20BCD7138

Q.1]



```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-10 */
3  package com.cse2005.lab10que1;
4  import java.lang.reflect.Method;
5  class Printer {
6      public <inputType> void printArray(inputType[] array) {
7          for (int i = 0; i < array.length; i++) {
8              System.out.println(array[i]);
9          }
10     }
11 }
12 public class Main {
13     public static void main(String[] args) {
14         Printer myPrinter = new Printer();
15         Integer[] intArray = { 1, 2, 3 };
16         String[] stringArray = {"PHYSICS (PHY)",
17                                 "CHEMISTRY PART-1 (CHY1)",
18                                 "CHEMISTRY PART-2 (CHY2)",
19                                 "MATHEMATICS & STATISTICS (MAT)",
20                                 "ELECTRONICS PAPER-1 (ELE1)",
21                                 "ELECTRONICS PAPER-2 (ELE2)",
22                                 "ENVIRONMENTAL STUDIES (EVS)"};
23         myPrinter.printArray( intArray );
24         myPrinter.printArray( stringArray );
25         int printArrayMethodCount = 0;
26         for (Method method : Printer.class.getDeclaredMethods()) {
27             String methodName = method.getName();
28             if (methodName.equals("printArray")) {
29                 printArrayMethodCount++;
30             }
31         }
32         if (printArrayMethodCount > 1) {
33             System.out.println("Method overloading is not allowed!");
34             assert printArrayMethodCount == 1;
35         }
36     }
37 }
```

Output console –



```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab10Que1; "JAVA_HOME=C:\\Pro
Running NetBeans Compile On Save execution. Phase execution is skipped and
Scanning for projects...

-----< com.cse2005:Lab10Que1 >-----
Building Lab10Que1 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab10Que1 ---
1
2
3
PHYSICS (PHY)
CHEMISTRY PART-1 (CHY1)
CHEMISTRY PART-2 (CHY2)
MATHEMATICS & STATISTICS (MAT)
ELECTRONICS PAPER-1 (ELE1)
ELECTRONICS PAPER-2 (ELE2)
ENVIRONMENTAL STUDIES (EVS)

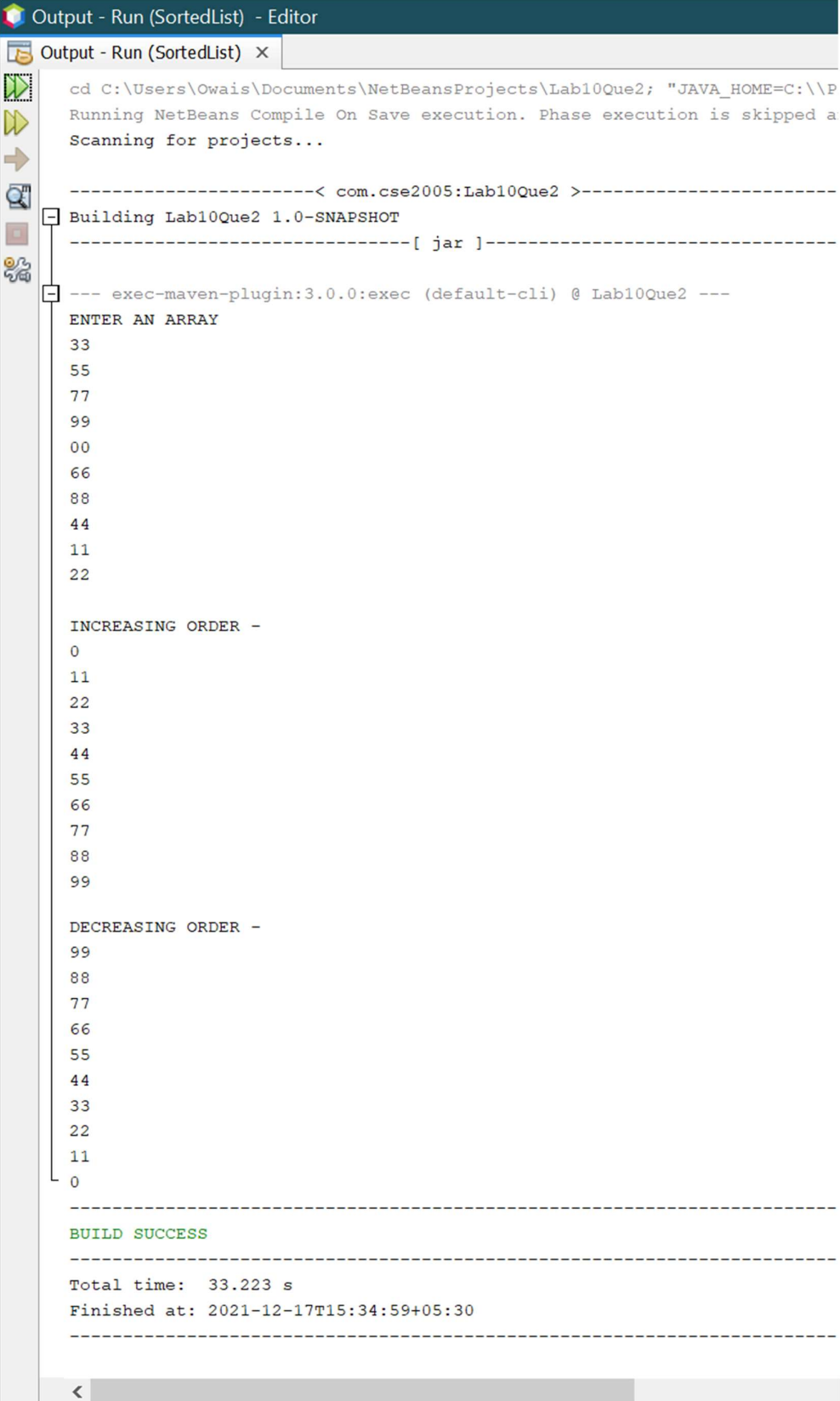
-----
BUILD SUCCESS
-----

Total time: 3.729 s
Finished at: 2021-12-17T15:21:22+05:30
-----
```

Q.2]

```
SortedList.java - Editor
SortedList.java ×
Source History
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-10 */
3  package com.cse2005.lab10que2;
4  import java.util.Scanner;
5  public class SortedList {
6      public static void main(String[] args) {
7          Scanner ed = new Scanner(System.in);
8          int[] a = new int[10];
9          int i, j, temp;
10         System.out.println("ENTER AN ARRAY");
11         for (i = 0; i < 10; i++) {
12             a[i] = ed.nextInt();
13         }
14         for (i = 0; i < 10; i++) {
15             for (j = i + 1; j < 10; j++) {
16                 if (a[i] > a[j]) {
17                     temp = a[i];
18                     a[i] = a[j];
19                     a[j] = temp;
20                 }
21             }
22         }
23         System.out.println("\nINCREASING ORDER -");
24         for (j = 0; j < 10; j++) {
25             System.out.println(a[j]);
26         }
27         for (i = 0; i < 10; i++) {
28             for (j = i + 1; j < 10; j++) {
29                 if (a[i] < a[j]) {
30                     temp = a[i];
31                     a[i] = a[j];
32                     a[j] = temp;
33                 }
34             }
35         }
36         System.out.println("\nDECREASING ORDER -");
37         for (j = 0; j < 10; j++) {
38             System.out.println(a[j]);
39         }
40     }
41 }
```

Output console –



```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab10Que2; "JAVA_HOME=C:\\P
Running NetBeans Compile On Save execution. Phase execution is skipped a
Scanning for projects...

-----< com.cse2005:Lab10Que2 >-----
Building Lab10Que2 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab10Que2 ---
ENTER AN ARRAY
33
55
77
99
00
66
88
44
11
22

INCREASING ORDER -
0
11
22
33
44
55
66
77
88
99

DECREASING ORDER -
99
88
77
66
55
44
33
22
11
0

-----
BUILD SUCCESS
-----

Total time: 33.223 s
Finished at: 2021-12-17T15:34:59+05:30
-----
```

Q.3]

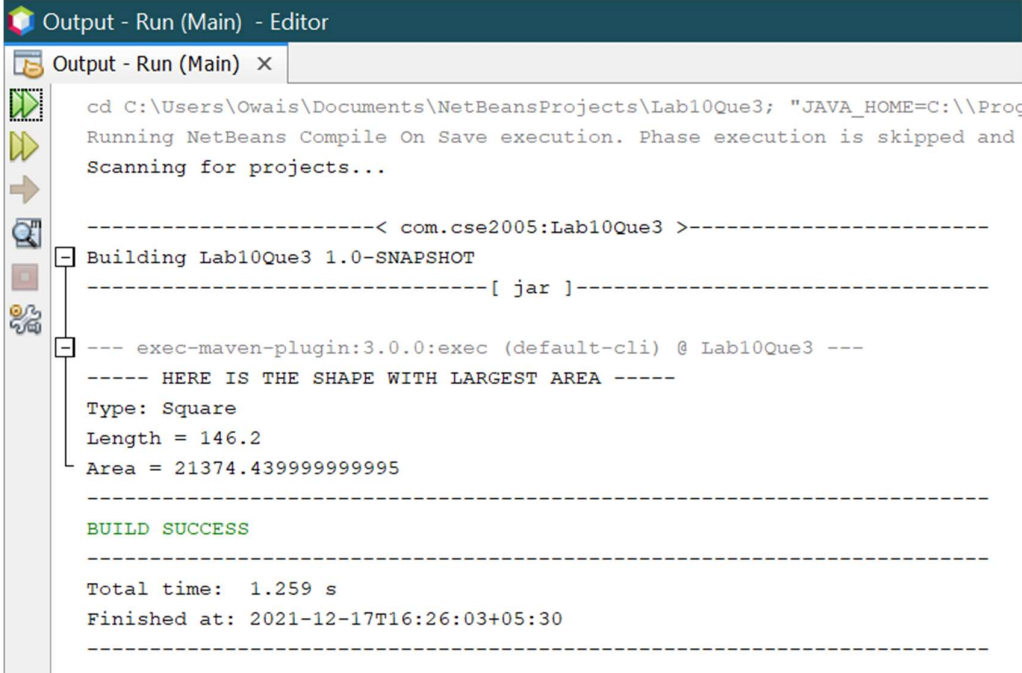
```
Main.java - Editor
Main.java ×
Source History
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-10 */
3  package com.cse2005.lab10que3;
4  import java.util. * ;
5  abstract class Node{
6      private double area;
7      abstract void calcArea();
8      abstract void DisplayDetails();
9      public double getArea(){
10         return area;
11     }
12     public void setArea(double ar){
13         area = ar;
14     }
15     public static void compareArea(List < ?>list){
16         Node largest = (Node)list.get(0);
17         for(int i = 0; i < list.size(); i++){
18             if(((Node)list.get(i)).getArea() > largest.getArea()){
19                 largest = (Node)list.get(i);
20             }
21         }
22         System.out.println("----- HERE IS THE SHAPE WITH LARGEST AREA -----" );
23         largest.DisplayDetails();
24     }
25 }
26 class Circle extends Node{
27     double radius;
28     Circle(double radius)
29     {
30         this.radius = radius;
31         calcArea();
32     }
33     void calcArea()
34     {
35         setArea(3.14 * radius * radius);
36     }
37     void DisplayDetails(){
38         System.out.println("Type: Circle");
39         System.out.println("Radius = " + radius);
40         System.out.println("Area = " + getArea());
41     }
42 }
```

```

43     class Square extends Node{
44         double length;
45         Square(double length){
46             this.length = length; calcArea();
47         }
48         void calcArea(){
49             setArea(length * length);
50         }
51         void DisplayDetails(){
52             System.out.println("Type: Square");
53             System.out.println("Length = " + length);
54             System.out.println("Area = " + getArea());
55         }
56     }
57     class Rectangle extends Node{
58         double l;
59         double b;
60         Rectangle(double l, double b){
61             this.l = l;
62             this.b = b;
63             calcArea();
64         }
65         void calcArea(){
66             setArea(l * b);
67         }
68         void DisplayDetails(){
69             System.out.println("Type: Rectangle");
70             System.out.println("Length = " + l);
71             System.out.println("Breadth = " + b);
72             System.out.println("Area = " + getArea());
73         }
74     }
75     public class Main{
76         public static void main(String[] args){
77             List<Node> ls=new ArrayList<>();
78             ls.add(new Rectangle(30, 40));
79             ls.add(new Circle(4.0));
80             ls.add(new Square(146.2));
81             ls.get(0).compareArea(ls);
82         }
83     }

```


Output console –



```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab10Que3; "JAVA_HOME=C:\\Progr
Running NetBeans Compile On Save execution. Phase execution is skipped and
Scanning for projects...

-----< com.cse2005:Lab10Que3 >-----
Building Lab10Que3 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab10Que3 ---
----- HERE IS THE SHAPE WITH LARGEST AREA -----
Type: Square
Length = 146.2
Area = 21374.439999999995

BUILD SUCCESS

Total time: 1.259 s
Finished at: 2021-12-17T16:26:03+05:30
```

Q.4]

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-10 */
package com.cse2005.lab10q4;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;
class CSE2005_Lab10_Qu4 {
    void printUnion(int arr1[], int arr2[], int m, int n){
        if (m > n) {
            int tempp[] = arr1;
            arr1 = arr2;
            arr2 = tempp;
            int temp = m;
            m = n;
            n = temp;
        }
        Arrays.sort(arr1);
        for (int i = 0; i < m; i++)
            System.out.print(arr1[i] + " ");
        for (int i = 0; i < n; i++) {
            if (binarySearch(arr1, 0, m - 1, arr2[i]) == -1)
                System.out.print(arr2[i] + " ");
        }
    }
    void printIntersection(int arr1[], int arr2[], int m, int n){
        if (m > n) {
            int tempp[] = arr1;
            arr1 = arr2;
            arr2 = tempp;
            int temp = m;
            m = n;
            n = temp;
        }
    }
}
```



```

Arrays.sort(arr1);
for (int i = 0; i < n; i++) {
    if (binarySearch(arr1, 0, m - 1, arr2[i]) != -1)
        System.out.println(arr2[i] + " ");
}
}

int binarySearch(int arr[], int l, int r, int x){
    if (r >= l) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);
        return binarySearch(arr, mid + 1, r, x);
    }
    return -1;
}

```

```

boolean ji = true;
boolean ji1 = true;

```

```

boolean check(int arr1[],int ch)
{
    for(int i =0;i<arr1.length;i++)
    {
        if(ch==arr1[i])
        {

            ji= true;
        }
        else
        {

```

```

        ji= false;
    }
}
return ji;
}
boolean check1(int arr2[],int ch)
{
    for(int i =0;i<arr2.length;i++)
    {
        if(ch==arr2[i])
        {

            ji1= true;
        }
        else
        {

            ji1= false;
        }
    }
    return ji1;
}

}

class Boolean_Operation
{
    public static void main(String[] args){
        CSE2005_Lab10_Qu4 u_i;
        u_i = new CSE2005_Lab10_Qu4();

        int arr1[] = { 7, 1, 5, 2, 3, 6 };
        int arr2[] = { 3, 8, 6, 2, 1, 7 };
        int m = arr1.length;
        int n = arr2.length;

```

```

        System.out.println("Union of two arrays is ");
        u_i.printUnion(arr1, arr2, m, n);
        System.out.println("");
        System.out.println("Intersection of two arrays is ");
        int nw = 7;
        u_i.printIntersection(arr1, arr2, m, n);
        boolean jh;
        jh= u_i.check(arr1, 7);
        boolean jh1;
        jh1= u_i.check(arr1, 7);

        if(jh==true)
        {
            System.out.println("True found in Arr1");
        }
        if(jh1==true)
        {
            System.out.println("True found in Arr2");
        }
    }
}

```

Output –

```

Union of two arrays is
1 2 3 5 6 7 8
Intersection of two arrays is
3
6
2
1
7
True found in Arr1
True found in Arr2

```

