ARRAY AND SEARCHING – LAB 01

EC 4070

DATA STRUCTURES AND ALGORITHMS

NAME	**: WIJAYAWARDHANA W.A.H.A.**
REGISTRATION NO.	**: 2019/E/166**
SEMESTER	**: SEMESTER 04**
DATE ASSIGNED	**: 28 FABRUARY 2022**

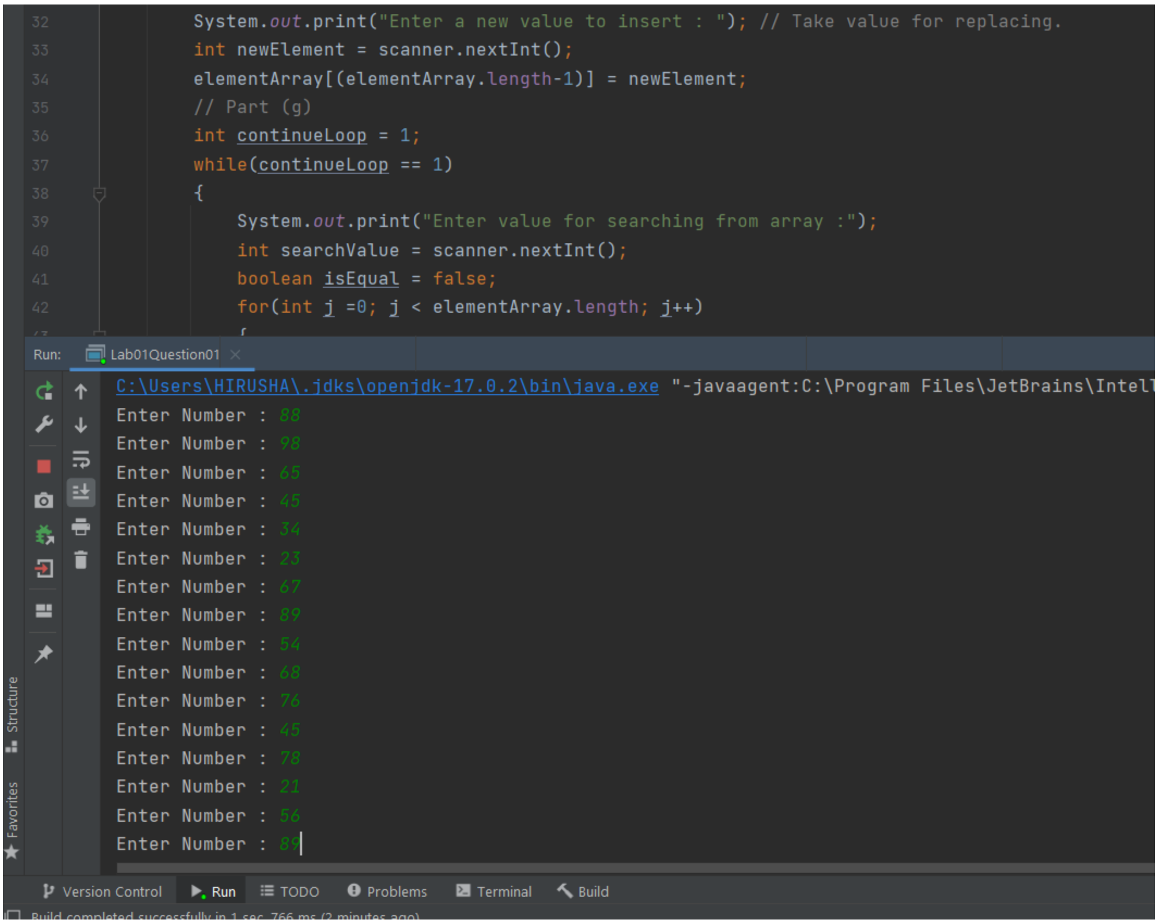
01.

Code:-

```
import java.util.Scanner; // Importing java scanner library.
public class Lab01Question01 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);  // Object of scanner.
      // Part (a)
      int[] elementArray = new int[20];         // New array size is 20
      // Part (b)
      for (int i = 0; i < elementArray.length; i++) // Read the values.
      {
         System.out.print("Enter Number : ");
         elementArray[i] = scanner.nextInt();
      }

      // Part (c)
      System.out.println("Enter two index need to interchange (0-19)");
      System.out.print("First index : "); // Take the index to change from user.
      int indexOne = scanner.nextInt();
      System.out.print("Second index : ");
      int indexTwo = scanner.nextInt();
      System.out.println("Before interchange : "+indexOne+" element "+elementArray[indexOne]
+ " " +indexTwo+" element "+ elementArray[indexTwo]);
      int temp = elementArray[indexOne]; // Keep the replacing value in temporary variable.
      elementArray[indexOne] = elementArray[indexTwo]; // Change the elements' values.
      elementArray[indexTwo] = temp;  // Reassign the value in temporary variable.
      System.out.println("After interchange : "+indexOne+" element "+elementArray[indexOne] +
" " +indexTwo+" element " + elementArray[indexTwo]);
      // Part (d)
      System.out.print("Enter index of array you need to read (0-19) : "); // Take index for reading.
      int readingArrayIndex = scanner.nextInt();
      System.out.println(readingArrayIndex+" element is " + elementArray[readingArrayIndex]);
      // Part (e)
      System.out.print("Enter index of array you need to delete (0-19) : "); // Take index for
deleting an item.
      int deletingArrayIndex = scanner.nextInt();
      elementArray[deletingArrayIndex] = 0; // For delete the relevant index element that index
replace by 0.
      // Part (f)
      System.out.print("Enter a new value to insert : "); // Take value for replacing.
      int newElement = scanner.nextInt();
      elementArray[(elementArray.length-1)] = newElement;
      // Part (g)
      int continueLoop = 1;
      while(continueLoop == 1)
```

```
          {
            System.out.print("Enter value for searching from array :");
            int searchValue = scanner.nextInt();
            boolean isEqual = false;
            for(int j =0; j < elementArray.length; j++)
            {
              if(elementArray[j] == searchValue)
              {
                isEqual = true;
                System.out.println("Index of equal value : "+j);
              }
            }
            if(isEqual == false)
            {
              System.out.println("Your element can not found in array.");
            }
            System.out.println("If you need to compare more press '1' or else press '0'");
            continueLoop = scanner.nextInt();
          }
        }
      }
```

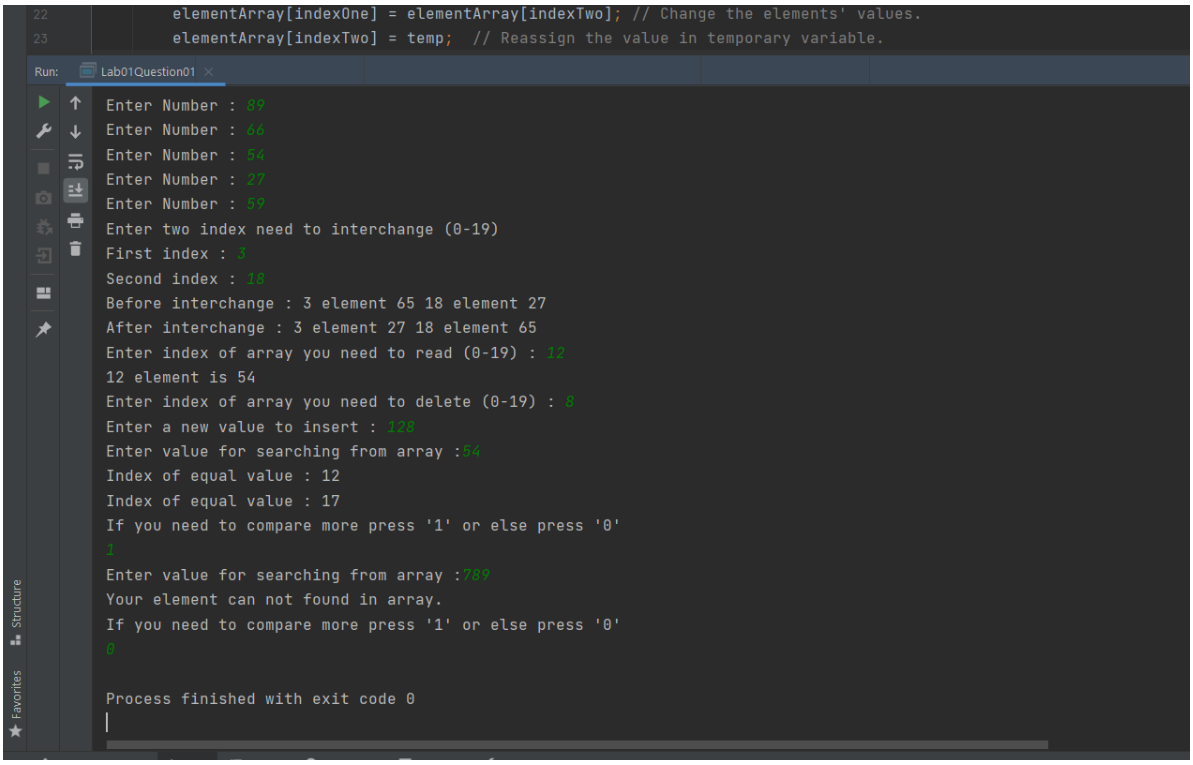
Outputs:-



```
        System.out.print("Enter a new value to insert : "); // Take value for replacing.
        int newElement = scanner.nextInt();
        elementArray[(elementArray.length-1)] = newElement;
        // Part (g)
        int continueLoop = 1;
        while(continueLoop == 1)
        {
            System.out.print("Enter value for searching from array :");
            int searchValue = scanner.nextInt();
            boolean isEqual = false;
            for(int j =0; j < elementArray.length; j++)
```

```
C:\Users\HIRUSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Intel
Enter Number : 88
Enter Number : 98
Enter Number : 65
Enter Number : 45
Enter Number : 34
Enter Number : 23
Enter Number : 67
Enter Number : 89
Enter Number : 54
Enter Number : 68
Enter Number : 76
Enter Number : 45
Enter Number : 78
Enter Number : 21
Enter Number : 56
Enter Number : 89
```

FIGURE 01 – READING ELEMENTS FROM USER.



```
        elementArray[indexOne] = elementArray[indexTwo]; // Change the elements' values.
        elementArray[indexTwo] = temp;  // Reassign the value in temporary variable.
```

```
Enter Number : 89
Enter Number : 66
Enter Number : 54
Enter Number : 27
Enter Number : 59
Enter two index need to interchange (0-19)
First index : 3
Second index : 18
Before interchange : 3 element 65 18 element 27
After interchange : 3 element 27 18 element 65
Enter index of array you need to read (0-19) : 12
12 element is 54
Enter index of array you need to delete (0-19) : 8
Enter a new value to insert : 128
Enter value for searching from array :54
Index of equal value : 12
Index of equal value : 17
If you need to compare more press '1' or else press '0'
1
Enter value for searching from array :789
Your element can not found in array.
If you need to compare more press '1' or else press '0'
0

Process finished with exit code 0
```

FIGURE 02 – OUTPUT OF THE QUESTION 01

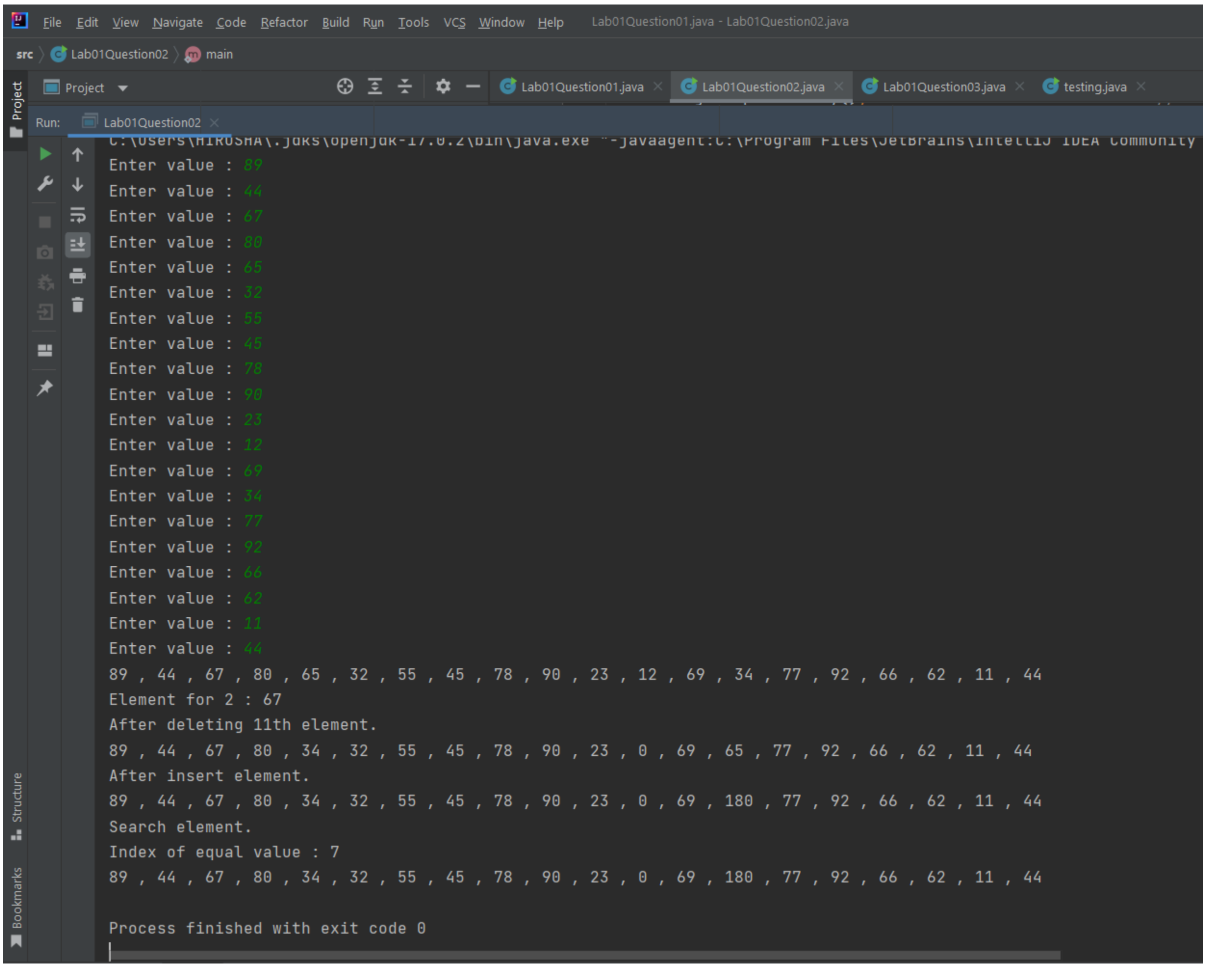
02.

Code:

```
import java.util.Scanner;       // Importing java scanning library.
public class Lab01Question02 {
   int[] elementArray = new int[20]; // Array defining with 20 elements.
   int indexOne;   // For interchangeElements method variable define.
   int indexTwo;   // For interchangeElements method variable define.
   int indexToPrint;  // For printElement method variable define.
   int indexToDelete; // For deleteElements method variable define.
   int elementValue;  // For insertElement method variable define.
   int insertIndex;   // For insertElement method variable define.
   int searchNumber;  // For searchElement method variable define.
   Scanner scanner = new Scanner(System.in);

   // Part 01
   public void readElements() // Reading elements method.
   {
      for(int i = 0; i <elementArray.length; i++)
      {
         System.out.print("Enter value : ");  // Get value for reading elements from user.
```

```
        elementArray[i] = scanner.nextInt(); // Take input value.
      }
    }

    // part 02
    public void printArray()  // Print the array.
    {
      for(int i = 0; i <elementArray.length; i++)
      {
        System.out.print(elementArray[i] );       // Print the element of array.
        if(i < elementArray.length-1)             // Check whether last element or not of array.
        {
          System.out.print(" , ");
        }
      }
      System.out.println();
    }

    // Part 03
    public void interchangeElements(int indexOne,int indexTwo) // Use for inter change values.
    {
      this.indexOne = indexOne;  // Assign value on the class with the value given at method
  calling value.
      this.indexTwo = indexTwo;  // Assign value on the class with the value given at method
  calling value.
      int temp = elementArray[indexOne]; // Keep element value in temporary value before
  replacing.
      elementArray[indexOne] = elementArray[indexTwo]; // Replace value 01 with value 02.
      elementArray[indexTwo] = temp; // Replace value 02 with value 01(temp variable value).
    }

    // Part 04
    public void printElement(int indexToPrint)  // Print the given index's element.
    {
      this.indexToPrint = indexToPrint;  // Assign value on the class with the value given at method
  calling value.
      System.out.println("Element for "+indexToPrint+ " : " + elementArray[indexToPrint]);
    }

    // Part 05
    public void deleteElement(int indexToDelete)  // Delete the given index's element.
    {
      this.indexToDelete = indexToDelete;  // Assign value on the class with the value given at
  method calling value.
      elementArray[indexToDelete] = 0;
    }

    // Part 06
    public void insertElement(int elementValue, int insertIndex)  // Use for
    {
        this.elementValue = elementValue;  // Assign value on the class with the value given at
method calling value.
        this.insertIndex = insertIndex;    // Assign value on the class with the value given at method
calling value.
        elementArray[insertIndex] = elementValue;
    }

    // Part 07
    public void searchElement(int searchNumber)
    {
        this.searchNumber = searchNumber;   // Assign value on the class with the value given at
method calling value.
        boolean isEqual = false;           // Check whether condition is true or false.
        for(int j =0; j < elementArray.length; j++)   // Checking the elements of array one by one
incrementing.
        {
            if(elementArray[j] == searchNumber)        // Check the searching number is equal or not.
            {
                isEqual = true;
                System.out.println("Index of equal value : "+j);    // Print the searching element index.
            }
        }
        if(!isEqual)          // If the searching element is not in array in will output this statement.
        {
            System.out.println("Your element can not found in array.");
        }
    }

    public static void main(String[] args) {
        Lab01Question02 newObject = new Lab01Question02();   // Making object of
Lab01Question02 class.
        newObject.readElements();                        // Calling the readElement method.
        newObject.printArray();                          // Calling printArray method.
        newObject.interchangeElements(4,13);  // Calling interchangeElements method.
        newObject.printElement(2);                // Calling printElement method.
        newObject.deleteElement(11);              // Calling deleteElement method.
        System.out.println("After deleting 11th element.");   // Calling printArray method.
        newObject.printArray();                           // Calling printArray method.
        System.out.println("After insert element.");         // Calling printArray method.
        newObject.insertElement(180,13);   // Calling insertElement method.
        newObject.printArray();                           // Calling printArray method.
        System.out.println("Search element.");              // Calling printArray method.
        newObject.searchElement(45);              // Calling searchElement method.
        newObject.printArray();                           // Calling printArray method.
    }

}
```

Output:-



```
C:\Users\HIROSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Enter value : 89
Enter value : 44
Enter value : 67
Enter value : 80
Enter value : 65
Enter value : 32
Enter value : 55
Enter value : 45
Enter value : 78
Enter value : 90
Enter value : 23
Enter value : 12
Enter value : 69
Enter value : 34
Enter value : 77
Enter value : 92
Enter value : 66
Enter value : 62
Enter value : 11
Enter value : 44
89 , 44 , 67 , 80 , 65 , 32 , 55 , 45 , 78 , 90 , 23 , 12 , 69 , 34 , 77 , 92 , 66 , 62 , 11 , 44
Element for 2 : 67
After deleting 11th element.
89 , 44 , 67 , 80 , 34 , 32 , 55 , 45 , 78 , 90 , 23 , 0 , 69 , 65 , 77 , 92 , 66 , 62 , 11 , 44
After insert element.
89 , 44 , 67 , 80 , 34 , 32 , 55 , 45 , 78 , 90 , 23 , 0 , 69 , 180 , 77 , 92 , 66 , 62 , 11 , 44
Search element.
Index of equal value : 7
89 , 44 , 67 , 80 , 34 , 32 , 55 , 45 , 78 , 90 , 23 , 0 , 69 , 180 , 77 , 92 , 66 , 62 , 11 , 44

Process finished with exit code 0
```

FIGURE 03 – OUTPUTS OF QUESTION 02

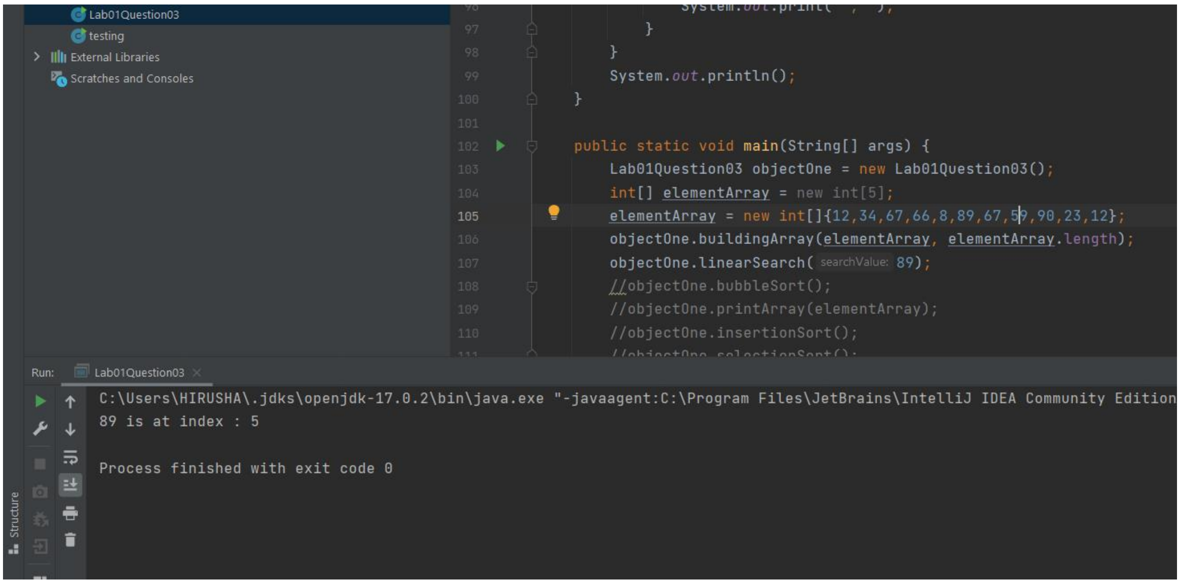
03.

Code:

```
public class Lab01Question03 {
  int arraySize;
  int[] elementArray = new int[arraySize];   // Define an array with 5 elements.
  int searchValue;                 // Define variable for searching value.
  boolean elementAvailable = false;

  public void buildingArray(int[] elementArray, int arraySize)
  {
    this.arraySize = arraySize;
    this.elementArray = elementArray;
  }

  // Liner search.
  public void linearSearch(int searchValue)    // LinearSearch method.
  {
    this.searchValue = searchValue;      // Assign value on the class with the value given at
method calling value.
    for(int i = 0; i < elementArray.length; i++) // Search one by one whether equal with
searchValue or not.
    {
      if(elementArray[i] == searchValue)   // Condition for equalling.
      {
        System.out.println(searchValue + " is at index : " + i);  // Print after element find.
        elementAvailable = true;                   // The variable will true if any element found from
array.
      }
    }
    if(elementAvailable == false)                  // Check the element is in array if not this will print.
    {
      System.out.println("Element is not available in array.");
    }
  }

  // Bubble sort.
  public void bubbleSort()
  {
    for(int i =0; i< elementArray.length;i++)     // For comparing element this will run.
    {
      for(int j = 0; j < elementArray.length-1; j++)  // For comparing with the upper loop element
all the other elements.
      {
        if(elementArray[j] > elementArray[j+1])    // Check the both element what is big and
what is small.
        {
               int temp = elementArray[j];           // Store in temporary variable the changing one
element.
               elementArray[j] = elementArray[j+1];   // Change the elements with other index.
               elementArray[j+1] = temp;            //// Change the elements with other index.
            }
         }
      }

   }

   // Insertion sort.
   public void insertionSort()
   {
      for(int i =1; i < elementArray.length; i++)     // For taking element one by one for comparing.
      {
         int tempValue = elementArray[i];          // Store in temporary variable for comparing
purposes.
         int j = i-1;
         while(j >= 0 && elementArray[j] >tempValue)  // Until it found less value this loop will run.
          {
            elementArray[j+1] = elementArray[j];     // Inter change the elements after the
statement is right.
            j--;          // Decrement the j variable.
         }
         elementArray[j+1] = tempValue;        // Reassign value for temporary variable.
      }
   }

   // Selection sort.
   public void selectionSort()
   {
      for(int i =0; i < elementArray.length-1; i++)    // For taking element one by one for
comparing.
      {
         int minValue = i;             // Assign first element as the minimum value.
         for(int j =i+1; j< elementArray.length; j++) //For search the element.
         {
            if(elementArray[j] < elementArray[minValue])    // Check what is the minimum value.
            {
               minValue = j;      // After statement is true reassign the minimum value.
            }
         }
         if(minValue != i)
         {
            int tempValue = elementArray[i];
            elementArray[i] = elementArray[minValue];
            elementArray[minValue] = tempValue;
         }
      }
```

```
  }

  // Print array.
  public void printArray(int[] elementArray)  // Print the array.
  {
    this.elementArray = elementArray;
    for(int i = 0; i <elementArray.length; i++)  // Print element one by one.
    {
      System.out.print(elementArray[i] );
      if(i < elementArray.length-1)
      {
        System.out.print(" , ");
      }
    }
    System.out.println();
  }

  public static void main(String[] args) {
    Lab01Question03 objectOne = new Lab01Question03();             // Create an object of
Lab01Question03 class.
    int []elementArray = new int[]{12,34,67,66,8,89,67,59,90,23,12}; // Assigning and defining
the array.
    objectOne.buildingArray(elementArray, elementArray.length);      // Calling the buildingArray
method for make the array.
    objectOne.linearSearch(11);                         // Calling linearSearch method for search the
element of 11.
    objectOne.bubbleSort();                                  // Calling bubbleSort method for sorting the
array.
    objectOne.printArray(elementArray);                         // Calling printArray method for
checking purposes.
    objectOne.insertionSort();                               // Calling bubbleSort method for sorting the
array.
    objectOne.printArray(elementArray);                         // Calling printArray method for
checking purposes.
    objectOne.selectionSort();                               // Calling bubbleSort method for sorting the
array.
    objectOne.printArray(elementArray);                         // Calling printArray method for
checking purposes.
  }
}
```

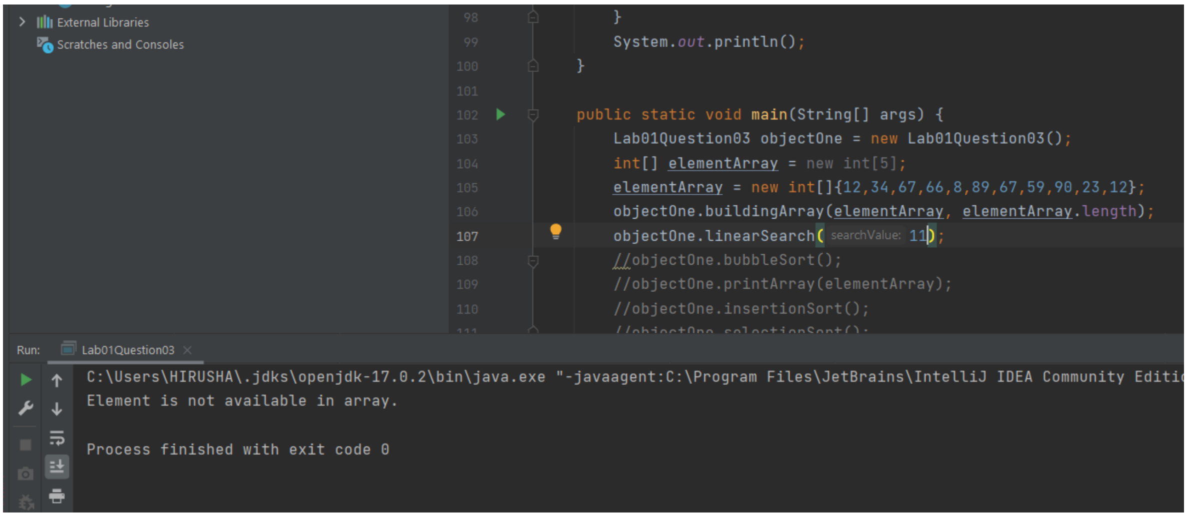
Output:



```
            System.out.print(" ");
        }
    }
    System.out.println();
}

public static void main(String[] args) {
    Lab01Question03 objectOne = new Lab01Question03();
    int[] elementArray = new int[5];
    elementArray = new int[]{12,34,67,66,8,89,67,59,90,23,12};
    objectOne.buildingArray(elementArray, elementArray.length);
    objectOne.linearSearch( searchValue: 89);
    //objectOne.bubbleSort();
    //objectOne.printArray(elementArray);
    //objectOne.insertionSort();
    //objectOne.selectionSort();
```

```
C:\Users\HIRUSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
89 is at index : 5

Process finished with exit code 0
```

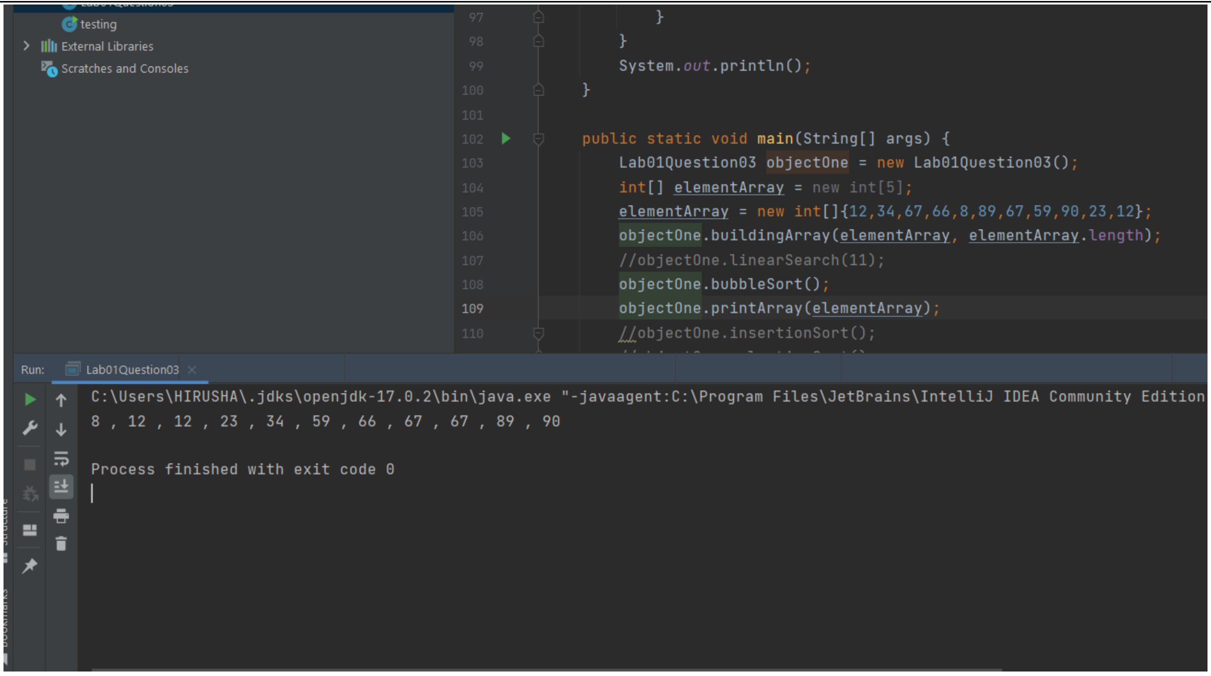
FIGURE 04 – OUTPUT FOR LINEAR SEARCH



```
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Lab01Question03 objectOne = new Lab01Question03();
        int[] elementArray = new int[5];
        elementArray = new int[]{12,34,67,66,8,89,67,59,90,23,12};
        objectOne.buildingArray(elementArray, elementArray.length);
        objectOne.linearSearch( searchValue: 11);
        //objectOne.bubbleSort();
        //objectOne.printArray(elementArray);
        //objectOne.insertionSort();
        //objectOne.selectionSort();
```

```
C:\Users\HIRUSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editi
Element is not available in array.

Process finished with exit code 0
```

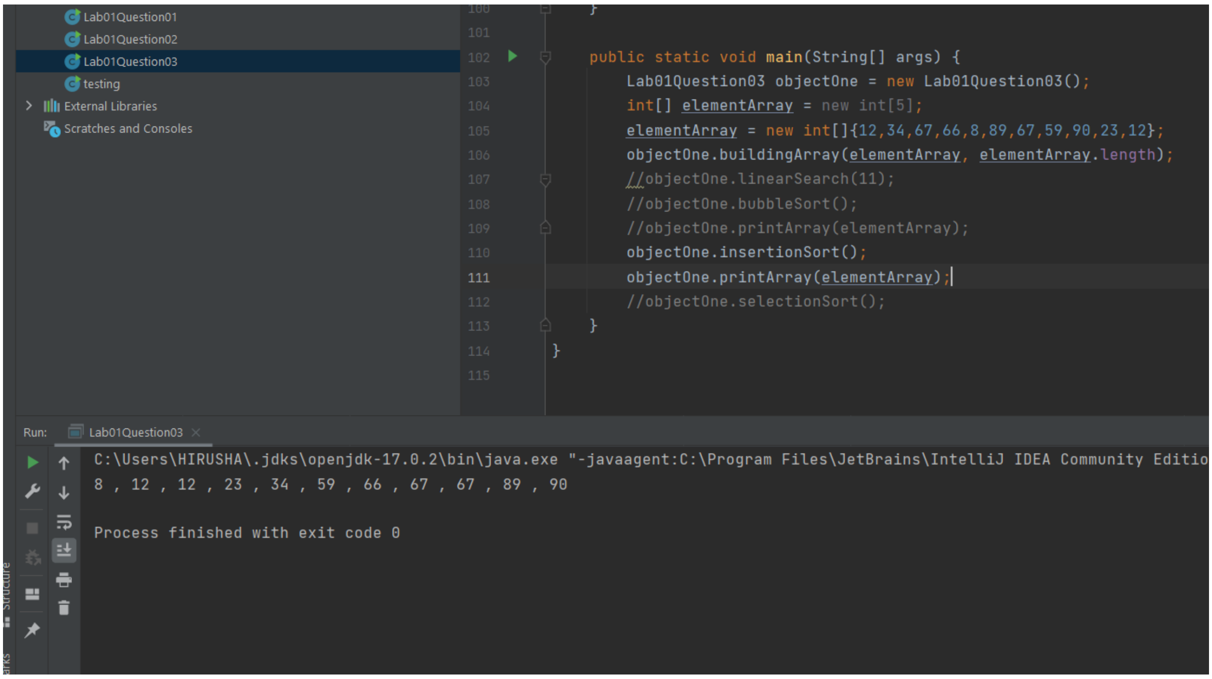
FIGURE 05 – OUTPUT FOR LINEAR SEARCH



```
            }
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Lab01Question03 objectOne = new Lab01Question03();
        int[] elementArray = new int[5];
        elementArray = new int[]{12,34,67,66,8,89,67,59,90,23,12};
        objectOne.buildingArray(elementArray, elementArray.length);
        //objectOne.linearSearch(11);
        objectOne.bubbleSort();
        objectOne.printArray(elementArray);
        //objectOne.insertionSort();
```

```
C:\Users\HIRUSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
8 , 12 , 12 , 23 , 34 , 59 , 66 , 67 , 67 , 89 , 90

Process finished with exit code 0
```

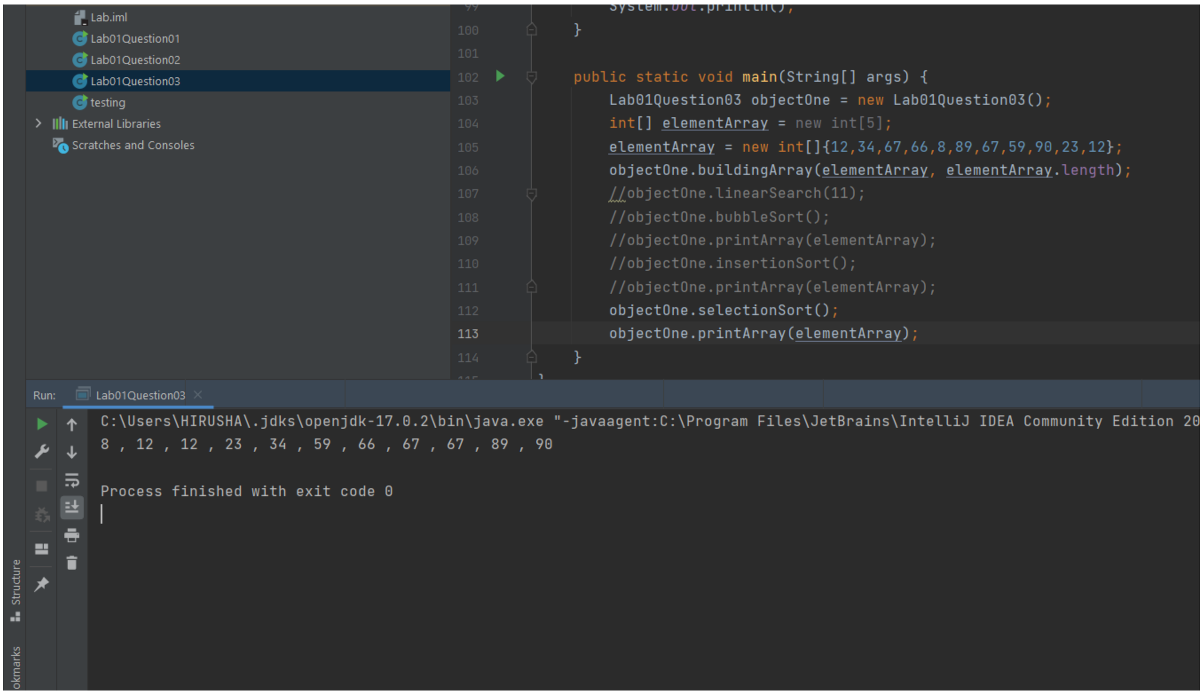
FIGURE 06 – OUTPUT FOR BINARY SORT



```
    }

    public static void main(String[] args) {
        Lab01Question03 objectOne = new Lab01Question03();
        int[] elementArray = new int[5];
        elementArray = new int[]{12,34,67,66,8,89,67,59,90,23,12};
        objectOne.buildingArray(elementArray, elementArray.length);
        //objectOne.linearSearch(11);
        //objectOne.bubbleSort();
        //objectOne.printArray(elementArray);
        objectOne.insertionSort();
        objectOne.printArray(elementArray);
        //objectOne.selectionSort();
    }
}
```

```
C:\Users\HIRUSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editio
8 , 12 , 12 , 23 , 34 , 59 , 66 , 67 , 67 , 89 , 90

Process finished with exit code 0
```

FIGURE 07 – OUTPUT FOR INSERTION SORT



```
        System.out.println();
    }

    public static void main(String[] args) {
        Lab01Question03 objectOne = new Lab01Question03();
        int[] elementArray = new int[5];
        elementArray = new int[]{12,34,67,66,8,89,67,59,90,23,12};
        objectOne.buildingArray(elementArray, elementArray.length);
        //objectOne.linearSearch(11);
        //objectOne.bubbleSort();
        //objectOne.printArray(elementArray);
        //objectOne.insertionSort();
        //objectOne.printArray(elementArray);
        objectOne.selectionSort();
        objectOne.printArray(elementArray);
    }
```

```
C:\Users\HIRUSHA\.jdks\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 20
8 , 12 , 12 , 23 , 34 , 59 , 66 , 67 , 67 , 89 , 90

Process finished with exit code 0
```

FIGURE 08 – OUTPUT FOR SELECTION SORT