

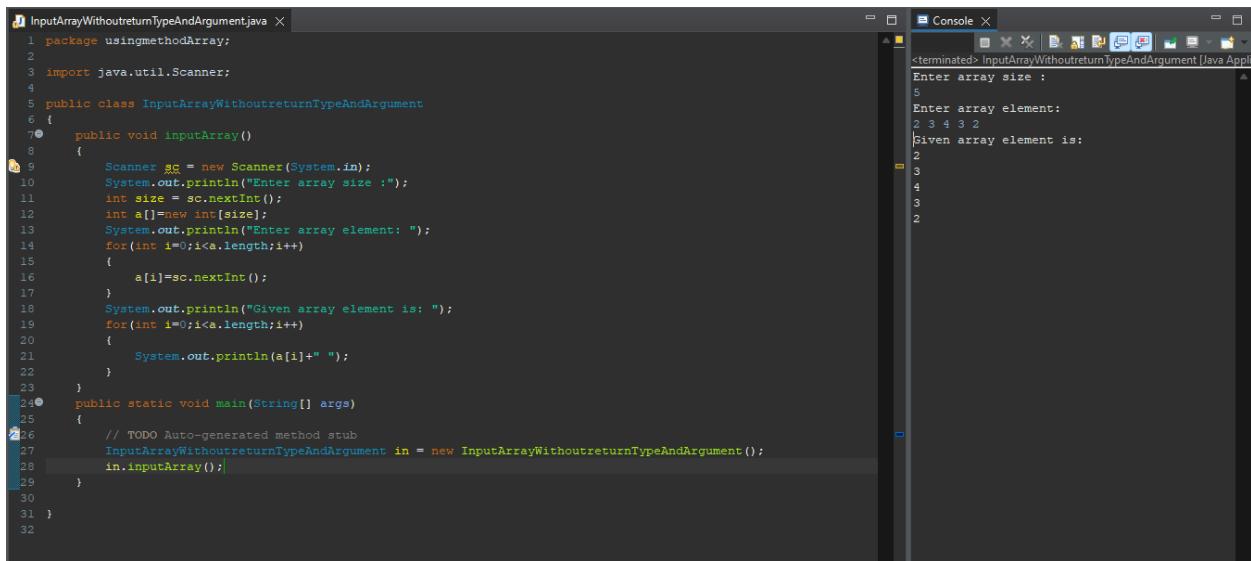
Assignment No:-21

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-3/6/2024

*First day Question.

Q1. Write a program to input an array and print it.



The screenshot shows a Java development environment with two panes. The left pane displays the source code for a Java class named `InputArrayWithoutreturnTypeAndArgument`. The code defines a method `inputArray` that reads an array of integers from standard input and prints its elements. The right pane shows the terminal window where the program is run. It prompts for the array size (5), then for each element (2, 3, 4, 3, 2), and finally prints the given array elements (2, 3, 4, 3, 2).

```
1 package usingmethodFromArray;
2
3 import java.util.Scanner;
4
5 public class InputArrayWithoutreturnTypeAndArgument
6 {
7     public void inputArray()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.println("Given array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            System.out.println(a[i]+" ");
22        }
23    }
24    public static void main(String[] args)
25    {
26        // TODO Auto-generated method stub
27        InputArrayWithoutreturnTypeAndArgument in = new InputArrayWithoutreturnTypeAndArgument();
28        in.inputArray();
29    }
30
31 }
32
```

```
<terminated> InputArrayWithoutreturnTypeAndArgument [Java Application]
Enter array size :
5
Enter array element:
2 3 4 3 2
Given array element is:
2
3
4
3
2
```

Q2. Write a program initialise an array and print it.

The screenshot shows an IDE interface with two tabs: 'InputArrayWithoutreturnTypeAndArgument.java' and 'InitializeArrayWithoutreturnTypeAndWithArgument.java'. The 'InitializeArrayWithoutreturnTypeAndWithArgument.java' tab is active, displaying the following code:

```
1 package usingmethodArray;
2
3 public class InitializeArrayWithoutreturnTypeAndWithArgument
4 {
5     public void ArrayInitializeArgument(int a[])
6     {
7         System.out.println("Given array element with argument");
8         for(int i=0;i<a.length;i++)
9         {
10             System.out.println(a[i]+" ");
11         }
12     }
13     public static void main(String[] args)
14     {
15         // TODO Auto-generated method stub
16         InitializeArrayWithoutreturnTypeAndWithArgument ia = new InitializeArrayWithoutreturnTypeAndWithArgument();
17         int a[]={1,2,3,21,23,11};
18         System.out.println("Array initialized in main Method");
19         for(int i=0;i<a.length;i++)
20         {
21             System.out.println(a[i]+" ");
22         }
23         ia.ArrayInitializeArgument(a);
24     }
25
26 }
```

The 'Console' tab shows the output of the program:

```
<terminated> InitializeArrayWithoutreturnTypeAndWithArgument [1]
Array initialized in main Method
1
2
3
21
23
11
Given array element with argument
1
2
3
21
23
11
```

Q3. Write a program to input an array and print its 1st and last element.

The screenshot shows an IDE interface with two tabs: 'InputArrayWithoutreturnTypeAndArgument.java' and 'InitializeArrayWithreturnTypeAndWithoutArgument.java'. The 'InitializeArrayWithreturnTypeAndWithoutArgument.java' tab is active, displaying the following code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class InitializeArrayWithreturnTypeAndWithoutArgument
6 {
7     public int[] printFirstAndLastArrEle()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size : ");
11        int size = sc.nextInt();
12        int a[]=new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.println("Given first and last array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            if(i==0)
22                System.out.println(a[i]+" ");
23            else if(i==a.length-1)
24                System.out.println(a[i]+" ");
25        }
26        return a;
27    }
28    public static void main(String[] args)
29    {
30        // TODO Auto-generated method stub
31        InitializeArrayWithreturnTypeAndWithoutArgument ia = new InitializeArrayWithreturnTypeAndWithoutArgument ();
32        ia.printFirstAndLastArrEle();
33    }
34 }
```

The 'Console' tab shows the output of the program:

```
<terminated> InitializeArrayWithreturnTypeAndWithoutArgument [1]
Enter array size :
5
Enter array element:
3 4 2 6 4
Given first and last array element is:
3
4
```

Q4. Write a program to input an array and find sum of its 1st and 2nd element.

The screenshot shows an IDE interface with several tabs open. The active tab contains Java code for calculating the sum of the first two elements of an array. The code uses a Scanner to read the array size and elements, then prints the sum. The console window shows the input '9 3 2 6 5' and the output 'Sum of first and second array element is: 12'.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumOfFirstAndSecondEleArrayWithReturnTypeAndArgument
6 {
7     public int[] printFirstAndSecondSumArrEle(int a[])
8     {
9         int sum=0;
10        System.out.println("Given first and last array element is: ");
11        for(int i=0;i<a.length-3;i++)
12        {
13            sum+=a[i];
14            System.out.println(a[i]+" ");
15        }
16        System.out.println("Sum of first and second array element is: \n"+sum);
17        return a;
18    }
19    public static void main(String[] args)
20    {
21        SumOfFirstAndSecondEleArrayWithReturnTypeAndArgument s = new SumOfFirstAndSecondEleArrayWithReturnTypeAndArgument();
22        Scanner sc = new Scanner(System.in);
23        System.out.println("Enter array size :");
24        int size = sc.nextInt();
25        int a[]=new int[size];
26        System.out.println("Enter array element: ");
27        for(int i=0;i<a.length;i++)
28        {
29            a[i]=sc.nextInt();
30        }
31        s.printFirstAndSecondSumArrEle(a);
32    }
33 }
34 }
```

Q5. Write a program to input an array and find sum.

The screenshot shows an IDE interface with several tabs open. The active tab contains Java code for calculating the sum of all elements in an array. The code uses a Scanner to read the array size and elements, then prints the total sum. The console window shows the input '4 2 1 5 6' and the output 'Sum of array element is: 18'.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumOfArrayWithoutreturnTypeAndArgument
6 {
7     public void sum()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[]=new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        int sum=0;
19        System.out.println("Given array element is: ");
20        for(int i=0;i<a.length;i++)
21        {
22            sum+=a[i];
23            System.out.println(a[i]+" ");
24        }
25        System.out.println("Sum of array element is: "+sum);
26    }
27
28    public static void main(String[] args)
29    {
30        // TODO Auto-generated method stub
31        SumOfArrayWithoutreturnTypeAndArgument s = new SumOfArrayWithoutreturnTypeAndArgument();
32        s.sum();
33    }
34 }
35 }
```

Q6. Write a program to input an array and find its average.

The screenshot shows a Java IDE interface with multiple tabs open. The active tab contains the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class AvgOfArrayWithoutreturnTypeAndWithArgument
6 {
7     public void average(int a[])
8     {
9         int sum=0;
10        System.out.println("Given array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            sum+=a[i];
14            System.out.print(a[i]+" ");
15        }
16        System.out.println("Given Average of array element is: "+(sum/a.length));
17    }
18    public static void main(String[] args)
19    {
20        // TODO Auto-generated method stub
21        AvgOfArrayWithoutreturnTypeAndWithArgument a = new AvgOfArrayWithoutreturnTypeAndWithArgument();
22        Scanner sc = new Scanner(System.in);
23        System.out.println("Enter array size :");
24        int size = sc.nextInt();
25        int a[] = new int[size];
26        System.out.println("Enter array element: ");
27        for(int i=0;i<a.length;i++)
28        {
29            a[i]=sc.nextInt();
30        }
31        a.average(a);
32    }
33}
34}
35}
```

The right side of the interface shows the console output:

```
<terminated> AvgOfArrayWithoutreturnTypeAndWithArgument
Enter array size :
5
Enter array element:
1 2 3 4 5
Given array element is:
1
2
3
4
5
Given Average of array element is: 3
```

Q7. Write a program to input an array and print even no

The screenshot shows a Java IDE interface with multiple tabs open. The active tab contains the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class EvenOfArrayWithreturnTypeAndWithoutArgument
6 {
7     public int[] EvenArrEle()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        int sum=0;
19        System.out.println("Given even array element is: ");
20        for(int i=0;i<a.length;i++)
21        {
22            if(a[i]%2==0)
23                System.out.print(a[i]+" ");
24        }
25        return a;
26    }
27    public static void main(String[] args)
28    {
29        // TODO Auto-generated method stub
30        EvenOfArrayWithreturnTypeAndWithoutArgument e = new EvenOfArrayWithreturnTypeAndWithoutArgument();
31        e.EvenArrEle();
32    }
33}
34}
35}
36}
```

The right side of the interface shows the console output:

```
<terminated> EvenOfArrayWithreturnTypeAndWithoutArgument
Enter array size :
5
Enter array element:
3 6 8 2 3
Given even array element is:
6
8
2
```

Q8. Write a program to input an array and print odd no.

```
1 package usingmethodarray;
2
3 import java.util.Scanner;
4
5 public class OddOfArrayWithReturnTypeAndArgument
6 {
7     public int[] oddArrEle(int a[])
8     {
9         int cnt = 0;
10        for(int i=0;i<a.length;i++)
11        {
12            if(a[i]%2!=0)
13            {
14                //System.out.println(a[i]+" ");
15                cnt++;
16            }
17        }
18        int b[] = new int[cnt];
19        int ind=0;
20        System.out.println("Given odd array element is: ");
21        for(int i=0;i<a.length;i++)
22        {
23            if(a[i]%2!=0)
24            {
25                //System.out.println(b[i]+" ");
26                b[ind++]=a[i];
27            }
28        }
29        return b;
30    }
31    public static void main(String[] args)
32    {
33        OddOfArrayWithReturnTypeAndArgument o = new OddOfArrayWithReturnTypeAndArgument();
34        Scanner sc = new Scanner(System.in);
35        System.out.println("Enter array size :");
36        int size = sc.nextInt();
37        int a[] = new int[size];
38        System.out.println("Enter array element: ");
39        for(int i=0;i<a.length;i++)
40        {
41            a[i]=sc.nextInt();
42        }
43        int b[] = o.oddArrEle(a);
44        for(int i=0;i<b.length;i++)
45        {
46            System.out.println(b[i]+" ");
47        }
48    }
49
50 }
51
```

```
<terminated> OddOfArrayWithReturnTypeAndArgument [Java Application]
Enter array size :
5
Enter array element:
1 2 3 4 5
Given odd array element is:
Given odd array element is:
1
3
5
```

```
<terminated> OddOfArrayWithReturnTypeAndArgument [Java Application]
Enter array size :
5
Enter array element:
1 2 3 4 5
Given odd array element is:
Given odd array element is:
1
3
5
```

Q9. Write a program to input an array and find the sum of even no.

The screenshot shows a Java code editor with a file named EvenSumOfArrayWithoutreturnTypeAndArgument.java. The code defines a class with a main method and a SumEvenArrEle method. The SumEvenArrEle method prompts for array size and elements, then iterates through the array to calculate the sum of even numbers. The output window shows the execution of the program with user input and the resulting sum.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class EvenSumOfArrayWithoutreturnTypeAndArgument
6 {
7     public void SumEvenArrEle()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        int sum=0;
19        System.out.println("Given even array element is: ");
20        for(int i=0;i<a.length;i++)
21        {
22            if(a[i]%2==0)
23            {
24                sum+=a[i];
25                System.out.println(a[i]+" ");
26            }
27        }
28        System.out.println("Sum of even array element is :" +sum);
29    }
30    public static void main(String[] args)
31    {
32        // TODO Auto-generated method stub
33        EvenSumOfArrayWithoutreturnTypeAndArgument e = new EvenSumOfArrayWithoutreturnTypeAndArgument();
34        e.SumEvenArrEle();
35    }
36}
37
```

```
<terminated> EvenSumOfArrayWithoutreturnTypeAndArgument
Enter array size :
5
Enter array element:
1 2 3 7 8
Given even array element is:
2
8
Sum of even array element is :10
```

Q10. Write a program to input an array and find the sum of odd no.

The screenshot shows a Java code editor with a file named SumOddOfArrayWithoutreturnTypeAndWithArgument.java. The code defines a class with a main method and a sumOfOdd method. The sumOfOdd method prompts for array size and elements, then iterates through the array to calculate the sum of odd numbers. The output window shows the execution of the program with user input and the resulting sum.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumOddOfArrayWithoutreturnTypeAndWithArgument
6 {
7     public void sumOfOdd(int a[])
8     {
9         int sum=0;
10        System.out.println("Given odd array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            if(a[i]%2!=0)
14            {
15                sum+=a[i];
16                System.out.println(a[i]+" ");
17            }
18        }
19        System.out.println("Given sum of odd array element is: "+sum);
20    }
21    public static void main(String[] args)
22    {
23        // TODO Auto-generated method stub
24        SumOddOfArrayWithoutreturnTypeAndWithArgument s = new SumOddOfArrayWithoutreturnTypeAndWithArgument();
25        Scanner sc = new Scanner(System.in);
26        System.out.println("Enter array size :");
27        int size = sc.nextInt();
28        int a[] = new int[size];
29        System.out.println("Enter array element: ");
30        for(int i=0;i<a.length;i++)
31        {
32            a[i]=sc.nextInt();
33        }
34        s.sumOfOdd(a);
35    }
36}
37
```

```
<terminated> SumOddOfArrayWithoutreturnTypeAndWithArgument
Enter array size :
5
Enter array element:
3 7 2 1 6
Given odd array element is:
3
7
1
Given sum of odd array element is: 11
```

Q11. Write a program to input an array and print its 1st 2nd and 3rd element.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for a class named `SumOfThreeEleWithReturnTypeAndWithoutArgument`. The code uses a scanner to read an array size and elements, then prints the first three elements. The console window shows the output of running the program, where it asks for array size (5), elements (1 2 3 4 5), and then prints the first three elements (1 2 3).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumOfThreeEleWithReturnTypeAndWithoutArgument
6 {
7     public int[] printThreeSumArrEle()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element :");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.println("Given first three array element is: ");
19        for(int i=0;i<a.length-2;i++)
20        {
21            System.out.println(a[i]+" ");
22        }
23        return a;
24    }
25    public static void main(String[] args)
26    {
27        // TODO Auto-generated method stub
28        SumOfThreeEleWithReturnTypeAndWithoutArgument s = new SumOfThreeEleWithReturnTypeAndWithoutArgument();
29        s.printThreeSumArrEle();
30    }
31}
32
33}
34
```

Q12. Write a program to input an array and find sum of its 1st and 2nd and 3rd element.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for a class named `SumOfThreeEleWithReturnTypeAndArgument`. The code uses a scanner to read an array size and elements, then calculates and prints the sum of the first three elements. The console window shows the output of running the program, where it asks for array size (5), elements (1 7 8 4 3), and then prints the sum of the first three elements (16).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumOfThreeEleWithReturnTypeAndArgument
6 {
7     public int[] printThreeSumArrEle(int[] a)
8     {
9         int sum=0;
10        System.out.println("Given first three array element is: ");
11        for(int i=0;i<a.length-2;i++)
12        {
13            sum+=a[i];
14            System.out.println(a[i]+" ");
15        }
16        System.out.println("Given sum of first three array element :" +sum);
17        return a;
18    }
19    public static void main(String[] args)
20    {
21        // TODO Auto-generated method stub
22        SumOfThreeEleWithReturnTypeAndArgument s = new SumOfThreeEleWithReturnTypeAndArgument();
23        Scanner sc = new Scanner(System.in);
24        System.out.println("Enter array size :");
25        int size = sc.nextInt();
26        int a[] = new int[size];
27        System.out.println("Enter array element :");
28        for(int i=0;i<a.length;i++)
29        {
30            a[i]=sc.nextInt();
31        }
32        s.printThreeSumArrEle(a);
33    }
34}
35
36}
37
```

Q13. Write a program to input an array and find no of 0s.no of positive no.no of negative no.

The image shows two screenshots of a Java IDE interface. Both screenshots display a code editor with Java code and a 'Console' window showing the output of running the program.

Code Snippet 1:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class FindZeroPosiAndNegetiveWithoutreturnTypeAndArgument
6 {
7     public void findZeroPosiAndNegetive()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int al[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<al.length;i++)
15        {
16            al[i]=sc.nextInt();
17        }
18        int c=0,c1=0,c2=0;
19        System.out.println("Given array element is: ");
20        for(int i=0;i<al.length;i++)
21        {
22            if(al[i]==0)
23            {
24                c++;
25            }
26            else if(al[i]>0)
27            {
28                c1++;
29            }
30            else if(al[i]<0)
31            {
32                c2++;
33            }
34        }
35        System.out.println("Number of zero present in array element is: "+c);
36        System.out.println("Number of positive present in array element is: "+c1);
37        System.out.println("Number of negative present in array element is: "+c2);
38    }
39
40    public static void main(String[] args)
41    {
42        // TODO Auto-generated method stub
43        FindZeroPosiAndNegetiveWithoutreturnTypeAndArgument f = new FindZeroPosiAndNegetiveWithoutreturnTypeAndArgument();
44        f.findZeroPosiAndNegetive();
45    }
46
47 }
```

Code Snippet 2:

```
12     int al[] = new int[size];
13     System.out.println("Enter array element: ");
14     for(int i=0;i<al.length;i++)
15     {
16         al[i]=sc.nextInt();
17     }
18     int c=0,c1=0,c2=0;
19     System.out.println("Given array element is: ");
20     for(int i=0;i<al.length;i++)
21     {
22         if(al[i]==0)
23         {
24             c++;
25         }
26         else if(al[i]>0)
27         {
28             c1++;
29         }
30         else if(al[i]<0)
31         {
32             c2++;
33         }
34     }
35     System.out.println("Number of zero present in array element is: "+c);
36     System.out.println("Number of positive present in array element is: "+c1);
37     System.out.println("Number of negative present in array element is: "+c2);
38 }
39
40 public static void main(String[] args)
41 {
42     // TODO Auto-generated method stub
43     FindZeroPosiAndNegetiveWithoutreturnTypeAndArgument f = new FindZeroPosiAndNegetiveWithoutreturnTypeAndArgument();
44     f.findZeroPosiAndNegetive();
45 }
46
47 }
```

Console Output:

Both consoles show the same output for both code snippets:

```
<terminated> FindZeroPosiAndNegetiveWithoutreturnTypeAndArgument []
Enter array size :
5
Enter array element:
0 -1 4 5 4
Given array element is:
Number of zero present in array element is: 1
Number of positive present in array element is: 3
Number of negative present in array element is: 1
```

Q14. Write a program to input an array and find its average of first 5 element.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class AvgOFFiveArrayWithoutreturnTypeAndWithArgument
6 {
7     public void averageOFfive(int a[])
8     {
9         int sum=0;
10        System.out.println("Given array element is: ");
11        for(int i=0;i<a.length-3;i++)
12        {
13            sum+=a[i];
14            System.out.println(a[i]+" ");
15        }
16        System.out.println("Given Average of array element is: "+(sum/5));
17    }
18    public static void main(String[] args)
19    {
20        // TODO Auto-generated method stub
21        AvgOFFiveArrayWithoutreturnTypeAndWithArgument a = new AvgOFFiveArrayWithoutreturnTypeAndWithArgument();
22        Scanner sc = new Scanner(System.in);
23        System.out.println("Enter array size :");
24        int size = sc.nextInt();
25        int a1[]=new int[size];
26        System.out.println("Enter array element: ");
27        for(int i=0;i<a1.length;i++)
28        {
29            a1[i]=sc.nextInt();
30        }
31        a.averageOFfive(a1);
32    }
33}
34
35
```

The console window shows the output of the program:

```
<terminated> AvgOFFiveArrayWithoutreturnTypeAndWithArgument
Enter array size :
8
Enter array element:
1 2 3 4 5 6 7 8
Given array element is:
1
2
3
4
5
Given Average of array element is: 3
```

Q15. Write a program to input an array and print the sum and count of even no.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class EvenSumAndCountOFArrayWithReturnType
6 {
7     public int[] EvenSumAndCountArrEle()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a1[]=new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a1.length;i++)
15        {
16            a1[i]=sc.nextInt();
17        }
18        int sum=0,c=0;
19        System.out.println("Given even array element is: ");
20        for(int i=0;i<a1.length;i++)
21        {
22            if(a1[i]%2==0)
23            {
24                sum+=a1[i];
25                System.out.println(a1[i]+" ");
26                c++;
27            }
28        }
29        System.out.println("Given sum of even array element is: "+sum);
30        System.out.println("Given count of even array element is: "+c);
31        return a1;
32    }
33    public static void main(String[] args)
34    {
35        EvenSumAndCountOFArrayWithReturnType e = new EvenSumAndCountOFArrayWithReturnType();
36        e.EvenSumAndCountArrEle();
37    }
38}
```

The console window shows the output of the program:

```
<terminated> EvenSumAndCountOFArrayWithReturnType
Enter array size :
5
Enter array element:
1 2 3 4 5
Given even array element is:
2
4
Given sum of even array element is: 6
Given count of even array element is: 2
```

Q16. Write a program to input an array and print the sum and count of odd no.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for calculating the sum and count of odd numbers in an array. The code uses a Scanner to read input and prints the results to the console. The console window shows the execution of the program, including user input and the resulting output.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class OddSumAndCountOFArrayWithReturnTypeAndArg {
6
7     public int[] OddSumAndCountArrEle(int a[])
8     {
9         int sum=0,c=0;
10        System.out.println("Given odd array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            if(a[i]%2!=0)
14            {
15                sum+=a[i];
16                System.out.println(a[i]+" ");
17                c++;
18            }
19        }
20        System.out.println("Given sum of odd array element is: "+sum);
21        System.out.println("Given count of odd array element is: "+c);
22        return a;
23    }
24
25    public static void main(String[] args)
26    {
27        OddSumAndCountOFArrayWithReturnTypeAndArg e = new OddSumAndCountOFArrayWithReturnTypeAndArg();
28        Scanner sc = new Scanner(System.in);
29        System.out.println("Enter array size :");
30        int size = sc.nextInt();
31        int a[] = new int[size];
32        System.out.println("Enter array element: ");
33        for(int i=0;i<a.length;i++)
34        {
35            a[i]=sc.nextInt();
36        }
37        e.OddSumAndCountArrEle(a);
    }
```

Console Output:

```
terminated> OddSumAndCountOFArrayWithReturnTypeAndArg [Java]
Enter array size :
6
Enter array element:
2 7 5 3 1 2
Given odd array element is:
7
5
3
1
Given sum of odd array element is: 16
Given count of odd array element is: 4
```

Q17. Write a program to enter an array and print element present at odd position and also find count.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for printing elements at odd positions and counting them. The code uses a Scanner to read input and prints the results to the console. The console window shows the execution of the program, including user input and the resulting output.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class OddPositionCountOFArrayWithoutReturnTypeAndArg
6 {
7     public void OddPositionCountArrEle()
8     {
9         int sum=0,c=0;
10        Scanner sc = new Scanner(System.in);
11        System.out.println("Enter array size :");
12        int size = sc.nextInt();
13        int a[] = new int[size];
14        System.out.println("Enter array element: ");
15        for(int i=0;i<a.length;i++)
16        {
17            a[i]=sc.nextInt();
18        }
19        System.out.println("Given odd position array element is: ");
20        for(int i=0;i<a.length;i++)
21        {
22            if((i+1)%2!=0)
23            {
24                sum+=a[i];
25                System.out.println(a[i]+" ");
26                c++;
27            }
28        }
29        System.out.println("Given count of odd position array element is: "+c);
30    }
31
32    public static void main(String[] args) []
33    {
34        // TODO Auto-generated method stub
35        OddPositionCountOFArrayWithoutReturnTypeAndArg o = new OddPositionCountOFArrayWithoutReturnTypeAndArg();
36        o.OddPositionCountArrEle();
    }
```

Console Output:

```
terminated> OddPositionCountOFArrayWithoutReturnTypeAndArg [Java]
Enter array size :
6
Enter array element:
2 7 5 3 2 1
Given odd position array element is:
2
5
2
Given count of odd position array element is: 3
```

Q18. Write a program to enter an array and print element present at even position and also find count.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named EvenPositionCountOFArrayWithArg. The right pane shows the console output of the program's execution.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class EvenPositionCountOFArrayWithArg
6 {
7     public void EvenPositionCountArrEle(int a[])
8     {
9         int sum=0, c=0;
10        System.out.println("Given even position array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            if((i+1)%2==0)
14            {
15                sum+=a[i];
16                System.out.println(a[i]+" ");
17                c++;
18            }
19        }
20        System.out.println("Given count of even position array element is: "+c);
21    }
22    public static void main(String[] args)
23    {
24        EvenPositionCountOFArrayWithArg e = new EvenPositionCountOFArrayWithArg();
25        Scanner sc = new Scanner(System.in);
26        System.out.println("Enter array size : ");
27        int size = sc.nextInt();
28        int a[]=new int[size];
29        System.out.println("Enter array element: ");
30        for(int i=0;i<a.length;i++)
31        {
32            a[i]=sc.nextInt();
33        }
34        e.EvenPositionCountArrEle(a);
35    }
36 }
37
```

The console output shows the following interaction:

```
<terminated> EvenPositionCountOFArrayWithArg [Java Application] C:\Users\S...
Enter array size :
6
Enter array element:
2 7 4 2 1 4
Given even position array element is:
7
2
4
Given count of even position array element is: 3
```

Second Assignment:

Q1. Write a program to enter an array and find the sum and count of element present at even position.

The screenshot shows an IDE interface with several tabs open. The active tab contains Java code for a class named `SumCountOfEvenPosieWithReturnType`. The code prompts the user for an array size and elements, then iterates through the array to calculate the sum and count of elements at even indices. The output window shows the input values and the resulting sum and count.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumCountOfEvenPosieWithReturnType
6 {
7
8     public int[] printCpuntSumEvenPosiArrEle()
9     {
10        Scanner sc = new Scanner(System.in);
11        System.out.println("Enter array size : ");
12        int size = sc.nextInt();
13        int a[] = new int[size];
14        System.out.println("Enter array element: ");
15        for(int i=0;i<a.length;i++)
16        {
17            a[i]=sc.nextInt();
18        }
19        int c=0,sum=0;
20        System.out.println("Given even position array element is: ");
21        for(int i=0;i<a.length;i++)
22        {
23            if((i+1)%2==0)
24            {
25                sum+=a[i];
26                System.out.println(a[i]+" ");
27                c++;
28            }
29        }
30        System.out.println("Sum of even position array element is: "+sum);
31        System.out.println("Count of even position array element is: "+c);
32        return a;
33    }
34    public static void main(String[] args)
35    {
36        // TODO Auto-generated method stub
37        SumCountOfEvenPosieWithReturnType s = new SumCountOfEvenPosieWithReturnType();
38        s.printCpuntSumEvenPosiArrEle();
39    }
40 }
41
42 }
```

```
Enter array size :
5
Enter array element:
2 5 7 2 1
Given even position array element is:
5
2
Sum of even position array element is: 7
Count of even position array element is: 2
```

Q2. Write a program to enter an array and find the sum and count of odd element.

The screenshot shows an IDE interface with several tabs open. The active tab contains Java code for a class named `SumCountOfOddEleWithReturnTypeAndArg`. The code prompts the user for an array size and elements, then iterates through the array to calculate the sum and count of elements at odd indices. The output window shows the input values and the resulting sum and count.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumCountOfOddEleWithReturnTypeAndArg
6 {
7
8     public int[] printCountAndSumOddEle(int a[])
9     {
10        int c=0,sum=0;
11        System.out.println("Given even array element is: ");
12        for(int i=0;i<a.length;i++)
13        {
14            if(a[i]%2!=0)
15            {
16                sum+=a[i];
17                System.out.println(a[i]+" ");
18                c++;
19            }
20        }
21        System.out.println("Sum of odd array element is: "+sum);
22        System.out.println("Count of odd array element is: "+c);
23        return a;
24    }
25    public static void main(String[] args)
26    {
27        // TODO Auto-generated method stub
28        SumCountOfOddEleWithReturnTypeAndArg s = new SumCountOfOddEleWithReturnTypeAndArg();
29        Scanner sc = new Scanner(System.in);
30        System.out.println("Enter array size : ");
31        int size = sc.nextInt();
32        int a[] = new int[size];
33        System.out.println("Enter array element: ");
34        for(int i=0;i<a.length;i++)
35        {
36            a[i]=sc.nextInt();
37        }
38        s.printCountAndSumOddEle(a);
39
40    }
41 }
```

```
Enter array size :
6
Enter array element:
3 7 4 1 2 3
Given even array element is:
3
7
1
3
Sum of odd array element is: 14
Count of odd array element is: 4
```

Q3. Write a program to replace all the 0's with 1's in your array. Your array is [26, 0, 67, 45, 0, 78, 54, 34, 10, 0,34].

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class ReplaceAllZeroWithOneWithoutArgAndReturnType
6 {
7     public void replaceValue()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        for(int i=0;i<a.length;i++)
19        {
20            if(a[i]==0)
21            {
22                a[i]=1;
23            }
24        }
25        System.out.println("After replacing zero with one: ");
26        for(int i=0;i<a.length;i++)
27        {
28            System.out.println(a[i]+" ");
29        }
30    }
31
32    public static void main(String[] args)
33    {
34        // TODO Auto-generated method stub
35        ReplaceAllZeroWithOneWithoutArgAndReturnType r = new ReplaceAllZeroWithOneWithoutArgAndReturnType();
36        r.replaceValue();
37    }
38
39 }
```

The console window shows the output of the program:

```
Enter array size :
11
Enter array element:
26 0 67 45 0 78 54 34 10 0 34
After replacing zero with one:
26
1
67
45
1
78
54
34
10
1
34
```

Q4. Write a program to enter an array and search any particular element and find the count.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SerachArrEleAndCountWithArg
6 {
7     public void searchArray(int a[], int c)
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter any element :");
11        int n = sc.nextInt();
12        for(int i=0;i<a.length;i++)
13        {
14            if(a[i]==n)
15            {
16                System.out.println("Position--> "+(i+1)+" given element is--> "+a[i]+" ");
17            }
18        }
19        System.out.println("Count of array element is: "+c);
20    }
21    public static void main(String[] args)
22    {
23        // TODO Auto-generated method stub
24        SerachArrEleAndCountWithArg s = new SerachArrEleAndCountWithArg();
25        Scanner sc = new Scanner(System.in);
26        System.out.println("Enter array size :");
27        int size = sc.nextInt();
28        int a[] = new int[size];
29        int c=0;
30        System.out.println("Enter array element: ");
31        for(int i=0;i<a.length;i++)
32        {
33            a[i]=sc.nextInt();
34            c++;
35        }
36        s.searchArray(a,c);
37    }
38
39 }
```

The console window shows the output of the program:

```
Enter array size :
5
Enter array element:
32 56 41 78 55
Enter any element :
56
Position--> 2 given element is--> 56
Count of array element is: 5
```

Q5. Write a program to check if an array of integers contains two specified elements.

The image shows a Java development environment with two code snippets and their respective console outputs.

Code Snippet 1:

```
19     int n= sc.nextInt();
20     int nl= sc.nextInt();
21     int c=0,cl=0,c2=0,i=0;
22     for(i=0;i<a.length;i++)
23     {
24         if(a[i]==n)
25         {
26             c++;
27         }
28         if(a[i]==nl)
29         {
30             cl++;
31         }
32         else if(a[i]!=n || a[i]!=nl)
33         {
34             c2++;
35         }
36     }
37     if(c>0)
38     {
39         System.out.println(n+" given element is Present");
40     }
41     if(cl>0)
42     {
43         System.out.println(nl+" given element is Present");
44     }
45     else if(c2==0)
46     {
47         System.out.println("Both are not present");
48     }
49     return a;
50 }
51 public static void main(String[] args)
52 {
    // TODO Auto-generated method stub
    CheckTwoEleWithReturnType s = new CheckTwoEleWithReturnType();
    s.CheckArrayConatainsOrNot();
53
54
55 }
```

Code Snippet 2:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class CheckTwoEleWithReturnType
6 {
7     public int[] CheckArrayConatainsOrNot()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[]=new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.println("Enter two element :");
19        int n= sc.nextInt();
20        int nl= sc.nextInt();
21        int c=0,cl=0,c2=0,i=0;
22        for(i=0;i<a.length;i++)
23        {
24            if(a[i]==n)
25            {
26                c++;
27            }
28            if(a[i]==nl)
29            {
30                cl++;
31            }
32            else if(a[i]!=n || a[i]!=nl)
33            {
34                c2++;
35            }
36        }
37        if(c>0)
38        {
39            System.out.println(n+" given element is Present");
40        }
41        if(cl>0)
42        {
43            System.out.println(nl+" given element is Present");
44        }
45        else if(c2==0)
46        {
47            System.out.println("Both are not present");
48        }
49    }
50 }
```

Console Output 1 (Top):

```
<terminated> CheckTwoEleWithReturnType [Java Application]
Enter array size :
5
Enter array element:
1 2 3 45 6
Enter two element :
6 45
6 given element is Present
45 given element is Present
```

Console Output 2 (Bottom):

```
<terminated> CheckTwoEleWithReturnType [Java Application]
Enter array size :
5
Enter array element:
1 2 3 45 6
Enter two element :
6 45
6 given element is Present
45 given element is Present
```

Q6. Write a program to enter an array and find the duplicate element and also count of that.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class CheckDuplicateWithReturnTypeAndArg
6 {
7     public int[] duplicateArray(int a[])
8     {
9         int c=0;
10        System.out.print("\nGiven duplicate array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            for(int j=i+1;j<a.length;j++)
14            {
15                if(a[i]==a[j])
16                {
17                    System.out.print(a[i]+" ");
18                    c++;
19                }
20            }
21        }
22        System.out.println("\nCount of duplicate array element is :" +c);
23        return a;
24    }
25
26    public static void main(String[] args)
27    {
28        // TODO Auto-generated method stub
29        CheckDuplicateWithReturnTypeAndArg c = new CheckDuplicateWithReturnTypeAndArg();
30        Scanner sc = new Scanner(System.in);
31        System.out.print("Enter array size :");
32        int size = sc.nextInt();
33        int a[] = new int[size];
34        System.out.print("Enter array element: ");
35        for(int i=0;i<a.length;i++)
36        {
37            a[i]=sc.nextInt();
38        }
39        c.duplicateArray(a);
40    }
41 }
```

The console output shows the execution of the program:

```
<terminated> CheckDuplicateWithReturnTypeAndArg [Java Application]
Enter array size :
5
Enter array element:
22 22 33 33 4
Given duplicate array element is: 22 33
Count of duplicate array element is :2
```

Q7. Write a program to enter an array and print the square of the element which is on odd position.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintSquareInOddPosiWithoutReturnTypeAndArg
6 {
7     public void squareOdd()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.print("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.print("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.print("\nGiven square of odd position array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            if((i+1)%2!=0)
22            {
23                a[i]*=a[i];
24                System.out.print(a[i]+" ");
25            }
26        }
27    }
28
29    public static void main(String[] args)
30    {
31        // TODO Auto-generated method stub
32        PrintSquareInOddPosiWithoutReturnTypeAndArg p = new PrintSquareInOddPosiWithoutReturnTypeAndArg();
33        p.squareOdd();
34    }
35
36 }
```

The console output shows the execution of the program:

```
<terminated> PrintSquareInOddPosiWithoutReturnTypeAndArg [Java Application]
Enter array size :
5
Enter array element:
2 7 8 4 1
Given square of odd position array element is: 4 64 1
```

Q8. Write a program to enter an array and print the square of the element which is on even position.

```

1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintSquareInEvenPosiWithArg
6 {
7
8     public void EvenPositionCountArrEle(int a[])
9     {
10         System.out.println("Given square of even position array element is: ");
11         for(int i=0;i<a.length;i++)
12         {
13             if((i+1)%2==0)
14             {
15                 a[i]*=a[i];
16                 System.out.println(a[i]+" ");
17             }
18         }
19     }
20
21     public static void main(String[] args) {
22         // TODO Auto-generated method stub
23         PrintSquareInEvenPosiWithArg o = new PrintSquareInEvenPosiWithArg();
24         Scanner sc = new Scanner(System.in);
25         System.out.println("Enter array size :");
26         int size = sc.nextInt();
27         int a[]=new int[size];
28         System.out.println("Enter array element : ");
29         for(int i=0;i<a.length;i++)
30         {
31             a[i]=sc.nextInt();
32         }
33         o.EvenPositionCountArrEle(a);
34     }
35 }
36

```

Console output:

```

<terminated> PrintSquareInEvenPosiWithArg [Java Application] C:\Users\Shree
Enter array size :
6
Enter array element:
2 7 5 4 3 2
Given square of even position array element is:
49
16
4

```

Q9. Write a program to replace all negative value with its immediate left elements square. Means arr[] = [12, 3, -19, 29, 5, -61, 44, 7, -9] Output array will be [12, 3, 9, 29, 5, 25, 44, 7, 49].//i=3a[i-1]

```

1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class ReplaceNegrativeEleWithReturnType
6 {
7
8     public int[] ReplaceArry()
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter array size :");
12         int size = sc.nextInt();
13         int a[]=new int[size];
14         System.out.println("Enter array element: ");
15         for(int i=0;i<a.length;i++)
16         {
17             a[i]=sc.nextInt();
18         }
19         for(int i=0;i<a.length;i++)
20         {
21             if(a[i]<0)
22             {
23                 a[i]=a[i-1]*a[i-1];
24             }
25         }
26         System.out.print("\nGiven square of negative array element is: ");
27         for(int i=0;i<a.length;i++)
28         {
29             System.out.print(a[i]+" ");
30         }
31         return a;
32     }
33
34     public static void main(String[] args)
35     {
36         // TODO Auto-generated method stub
37         ReplaceNegrativeEleWithReturnType c = new ReplaceNegrativeEleWithReturnType();
38         c.ReplaceArry();
39     }
40 }

```

Console output:

```

<terminated> ReplaceNegrativeEleWithReturnType [Java Application] C:\Users\Shree\p2\pool\plugins\org.
Enter array size :
9
Enter array element:
12 3 -19 29 5 -61 44 7 -9
Given square of negative array element is: 12 3 9 29 5 25 44 7 49

```

Q10. Write a program to enter an array and print it in reverse order.

The screenshot shows a Java IDE interface with two panes. The left pane displays the source code for a class named `PrintReverseArrWithReturnTypeAndArg`. The right pane shows the output of the program's execution in the console.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintReverseArrWithReturnTypeAndArg
6 {
7
8     public int[] reverseArray(int a[])
9     {
10         int g=0;
11         System.out.print("\nGiven reverse array element is: ");
12         for(int i=a.length-1;i>=0;i--)
13         {
14             System.out.print(a[i]+" ");
15         }
16         return a;
17     }
18     public static void main(String[] args)
19     {
20         // TODO Auto-generated method stub
21         PrintReverseArrWithReturnTypeAndArg c = new PrintReverseArrWithReturnTypeAndArg();
22         Scanner sc = new Scanner(System.in);
23         System.out.println("Enter array size :");
24         int size = sc.nextInt();
25         int a[]=new int[size];
26         System.out.println("Enter array element: ");
27         for(int i=0;i<a.length;i++)
28         {
29             a[i]=sc.nextInt();
30         }
31         c.reverseArray(a);
32     }
33 }
34 }
```

Console Output:

```
<terminated> PrintReverseArrWithReturnTypeAndArg [Java Application] C:\Users\Shree\p2\pool\plugins\ox
Enter array size :
6
Enter array element:
4 7 8 9 6 7 4
|
Given reverse array element is: 4 6 7 9 8 7 4
```

Q11. Write a program to enter an array and find the no in array which is equal to the sum of 1st and last element.

The screenshot shows a Java IDE interface with two panes. The left pane displays the source code for a class named `PrintSumOfFirstLastEleWithoutReturnTypeAndArg`. The right pane shows the output of the program's execution in the console.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintSumOfFirstLastEleWithoutReturnTypeAndArg
6 {
7
8     public void sumOfFirstAndLastArr()
9     {
10         int sum=0,g=0;
11         Scanner sc = new Scanner(System.in);
12         System.out.println("Enter array size :");
13         int size = sc.nextInt();
14         int a[]=new int[size];
15         System.out.println("Enter array element: ");
16         for(int i=0;i<a.length;i++)
17         {
18             a[i]=sc.nextInt();
19         }
20         sum=a[0]+a[a.length-1];
21         System.out.println("Given sum of first and last array element is: ");
22         for(int i=0;i<a.length;i++)
23         {
24             if(a[i]==sum)
25             {
26                 System.out.println(a[i]+" ");
27             }
28         }
29     }
30     public static void main(String[] args) {
31         // TODO Auto-generated method stub
32         PrintSumOfFirstLastEleWithoutReturnTypeAndArg o = new PrintSumOfFirstLastEleWithoutReturnTypeAndArg();
33         o.sumOfFirstAndLastArr();
34     }
35 }
36 }
```

Console Output:

```
<terminated> PrintSumOfFirstLastEleWithoutReturnTypeAndArg [Java Application] C:\Us
Enter array size :
7
Enter array element:
1 3 3 6 6 3 5
Given sum of first and last array element is:
6
6
```

3rd Assignment

Q1. Write a program to merge two Array and print the final array.

The screenshot shows a Java development environment with multiple tabs open. The active tab contains the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class MergeArrayWithoutArgReturnType
6 {
7     public void mergeArr()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int size1 = sc.nextInt();
13        int a[] = new int[size];
14        int b[] = new int[size];
15        int c[] = new int[a.length+b.length];
16        int x=0;
17        System.out.println("Enter 1 array element: ");
18        for(int i=0;i<a.length;i++)
19        {
20            a[i]=sc.nextInt();
21            c[i]=a[i];
22            x++;
23        }
24        System.out.println("Enter 2 array element: ");
25        for(int i=0;i<b.length;i++)
26        {
27            b[i]=sc.nextInt();
28            c[x++]=b[i];
29        }
30        System.out.println("Given two merged array element is: ");
31        for(int i=0;i<c.length;i++)
32        {
33            System.out.println(c[i]+" ");
34        }
35    }
36    public static void main(String[] args)
37    {
38        MergeArrayWithoutArgReturnType m = new MergeArrayWithoutArgReturnType();
39        m.mergeArr();
40    }
41 }
```

The console window to the right shows the execution of the program. It prompts for the array size, then for the elements of two arrays, and finally prints the merged array.

```
<terminated> MergeArrayWithoutArgReturnType [Java Application] C:\Users\Shree\p2\po...
Enter array size :
5
6
Enter 1 array element:
1 5 3 7 8
Enter 2 array element:
2 4 6 9 10 11
Given two merged array element is:
1
5
3
7
8
2
4
6
9
10
11
```

Q2. Write a program to enter an array and sort that in ascending order.

The screenshot shows a Java development environment with multiple tabs open. The active tab contains the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class AscendingArrayWithArg
6 {
7     public void AscendingArrEle(int a[])
8     {
9         for(int i=0;i<a.length;i++)
10        {
11            for(int j=i+1;j<a.length;j++)
12            {
13                if(a[i]>a[j])
14                {
15                    int t=a[i];
16                    a[i]=a[j];
17                    a[j]=t;
18                }
19            }
20        }
21        System.out.println("Given ascending array element is: ");
22        for(int i=0;i<a.length;i++)
23        {
24            System.out.println(a[i]+" ");
25        }
26    }
27    public static void main(String[] args)
28    {
29        AscendingArrayWithArg a = new AscendingArrayWithArg();
30        Scanner sc = new Scanner(System.in);
31        System.out.println("Enter array size :");
32        int size = sc.nextInt();
33        int a1[] = new int[size];
34        System.out.println("Enter 1 array element: ");
35        for(int i=0;i<a1.length;i++)
36        {
37            a1[i]=sc.nextInt();
38        }
39        a.AscendingArrEle(a1);
40    }
41 }
```

The console window to the right shows the execution of the program. It prompts for the array size, then for the elements of one array, and finally prints the sorted array.

```
<terminated> AscendingArrayWithArg [Java Application] C:\Users\Shree\p2\pool\plugins\o...
Enter array size :
5
Enter 1 array element:
2 5 6 7 4
Given ascending array element is:
2
4
5
6
7
```

Q3. Write a program to enter an array and sort that in descending order.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named `AscendingArrayWithReturnType`. The right pane shows the console output of the application running. The user enters an array size of 5 and elements 6, 4, 3, 7, 3. The sorted array in descending order is printed as 7, 6, 4, 3, 3.

```
1 package usingsmethodArray;
2
3 import java.util.Scanner;
4
5 public class AscendingArrayWithReturnType
6 {
7     public int[] DescendingArrEle()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        for(int i=0;i<a.length;i++)
19        {
20            for(int j=i+1;j<a.length;j++)
21            {
22                if(a[i]<a[j])
23                {
24                    int t=a[i];
25                    a[i]=a[j];
26                    a[j]=t;
27                }
28            }
29        }
30        System.out.println("Given descending array element is: ");
31        for(int i=0;i<a.length;i++)
32        {
33            System.out.println(a[i]+" ");
34        }
35        return a;
36    }
37    public static void main(String[] args)
38    {
39        AscendingArrayWithReturnType a = new AscendingArrayWithReturnType();
40        a.DescendingArrEle();
41    }
42 }
```

```
Enter array size :
5
Enter array element:
6 4 3 7 3
Given descending array element is:
7
6
4
3
3
```

Q4. Write a program to enter an array and find the second max element.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named `SecMaxArrayWithReturnTypeAndArg`. The right pane shows the console output of the application running. The user enters an array size of 5 and elements 3, 7, 2, 1, 8. The second maximum element is printed as 7.

```
3 import java.util.Scanner;
4
5 public class SecMaxArrayWithReturnTypeAndArg
6 {
7     public int secMax(int a[])
8     {
9         int max=0, smax=0;
10        for(int i=0;i<a.length;i++)
11        {
12            if(a[i]>max)
13            {
14                smax=max;
15                max=a[i];
16            }
17            else if(a[i]>smax)
18            {
19                smax=a[i];
20            }
21        }
22        System.out.println("Second max array element is: "+smax);
23        return smax;
24    }
25    public static void main(String[] args)
26    {
27        SecMaxArrayWithReturnTypeAndArg al = new SecMaxArrayWithReturnTypeAndArg();
28        Scanner sc = new Scanner(System.in);
29        System.out.println("Enter array size :");
30        int size = sc.nextInt();
31        int a[] = new int[size];
32        System.out.println("Enter array element: ");
33        for(int i=0;i<a.length;i++)
34        {
35            a[i]=sc.nextInt();
36        }
37        System.out.println("Given array element is: ");
38        for(int i=0;i<a.length;i++)
39        {
40            System.out.println(a[i]+" ");
41        }
42        al.secMax(a);
43    }
44 }
```

```
Enter array size :
5
Enter array element:
3 7 2 1 8
Given array element is:
3
7
2
1
8
Second max array element is: 7
```

Q5. Write a program to enter an array and find the second min element.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named `SecMinArrayWithReturnTypeAndArg`. The right pane shows the console output of the program's execution.

```
1 package usingmethodArray;
2 import java.util.Scanner;
3 public class SecMinArrayWithReturnTypeAndArg
4 {
5     public int secMin(int a[])
6     {
7         int min=Integer.MAX_VALUE, smin=Integer.MAX_VALUE;
8         for(int i=0;i<a.length;i++)
9         {
10             if(a[i]<min)
11             {
12                 smin=min;
13                 min=a[i];
14             }
15             else if(a[i]<smin)
16             {
17                 smin=a[i];
18             }
19         }
20         System.out.println("Second min array element is: "+smin);
21         return smin;
22     }
23     public static void main(String[] args)
24     {
25         SecMinArrayWithReturnTypeAndArg al = new SecMinArrayWithReturnTypeAndArg();
26         Scanner sc = new Scanner(System.in);
27         System.out.println("Enter array size :");
28         int size = sc.nextInt();
29         int a[]=new int[size];
30         System.out.println("Enter array element: ");
31         for(int i=0;i<a.length;i++)
32         {
33             a[i]=sc.nextInt();
34         }
35         System.out.println("Given array element is: ");
36         for(int i=0;i<a.length;i++)
37         {
38             System.out.println(a[i]+" ");
39         }
40         al.secMin(a);
41     }
42 }
```

The console output shows the user entering an array size of 6, followed by the elements 4, 7, 8, 2, 1, 5. The program then prints the second minimum element as 2.

```
<terminated> SecMinArrayWithReturnTypeAndArg [Java Application] C:\Users\Shree\p2\pool\SecMinArrayWithReturnTypeAndArg.java
Enter array size :
6
Enter array element:
4 7 8 2 1 5
Given array element is:
4
7
8
2
1
5
Second min array element is: 2
```

Q6. Write a program to enter an array and find max and min element.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named `MaxMinArrayWithReturnTypeAndArg`. The right pane shows the console output of the program's execution.

```
1 package usingmethodArray;
2 import java.util.Scanner;
3 public class MaxMinArrayWithReturnTypeAndArg
4 {
5     public int[] minMax(int a[])
6     {
7         int min=Integer.MAX_VALUE,max=0;
8         for(int i=0;i<a.length;i++)
9         {
10             if(a[i]<min)
11             {
12                 min=a[i];
13             }
14             else if(a[i]>max)
15             {
16                 max=a[i];
17             }
18         }
19         System.out.println("Max array element is: "+max);
20         System.out.println("Min array element is: "+min);
21         return a;
22     }
23     public static void main(String[] args)
24     {
25         MaxMinArrayWithReturnTypeAndArg al = new MaxMinArrayWithReturnTypeAndArg();
26         Scanner sc = new Scanner(System.in);
27         System.out.println("Enter array size :");
28         int size = sc.nextInt();
29         int a[]=new int[size];
30         System.out.println("Enter array element: ");
31         for(int i=0;i<a.length;i++)
32         {
33             a[i]=sc.nextInt();
34         }
35         System.out.println("Given array element is: ");
36         for(int i=0;i<a.length;i++)
37         {
38             System.out.println(a[i]+" ");
39         }
40         al.minMax(a);
41     }
42 }
```

The console output shows the user entering an array size of 5, followed by the elements 3, 7, 8, 2, 1. The program then prints the maximum element as 8 and the minimum element as 1.

```
<terminated> MaxMinArrayWithReturnTypeAndArg [Java Application] C:\Users\Shree\p2\pool\MaxMinArrayWithReturnTypeAndArg.java
Enter array size :
5
Enter array element:
3 7 8 2 1
Given array element is:
3
7
8
2
1
Max array element is: 8
Min array element is: 1
```

Q7. Write a program to input 10 elements in float and print it.

The screenshot shows an IDE interface with multiple tabs at the top. The active tab is 'FloatArrayWithoutReturnTypeAndArg'. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class FloatArrayWithoutReturnTypeAndArg
6 {
7     public void printFloat()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        float a[] = new float[size];
13        System.out.println("Enter float array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextFloat();
17        }
18        System.out.println("Given float array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            System.out.println(a[i]+" ");
22        }
23    }
24    public static void main(String[] args)
25    {
26        // TODO Auto-generated method stub
27        FloatArrayWithoutReturnTypeAndArg f = new FloatArrayWithoutReturnTypeAndArg();
28        f.printFloat();
29
30    }
31
32 }
33
```

The right side of the interface shows the 'Console' window with the output of the program. It prompts for the array size (5), then prints the elements 2.2, 3.0, 3.1, 2, 3.1, followed by 2.2, 3.0, 3.1, 2.0, and 3.1.

Q8. Write a program to input 10 name and print with index, name.

The screenshot shows an IDE interface with multiple tabs at the top. The active tab is 'PrintIndAndNameWithoutReturnTypeAndArg'. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintIndAndNameWithoutReturnTypeAndArg
6 {
7
8    public void printName()
9    {
10        Scanner sc = new Scanner(System.in);
11        System.out.println("Enter array size :");
12        int size = sc.nextInt();
13        String a[] = new String[size];
14        System.out.println("Enter name: ");
15        for(int i=0;i<a.length;i++)
16        {
17            a[i]=sc.next();
18        }
19        System.out.println("Given name with index: ");
20        for(int i=0;i<a.length;i++)
21        {
22            System.out.println(i+" <-Index Name--> "+a[i]+" ");
23        }
24    }
25    public static void main(String[] args)
26    {
27        // TODO Auto-generated method stub
28        PrintIndAndNameWithoutReturnTypeAndArg f = new PrintIndAndNameWithoutReturnTypeAndArg();
29        f.printName();
30
31    }
32
33 }
```

The right side of the interface shows the 'Console' window with the output of the program. It prompts for the array size (5), then asks for names: sangram, harshad, omkar, aditya, kartik. It then prints the names with their indices: 0 <-Index Name--> sangram, 1 <-Index Name--> harshad, 2 <-Index Name--> omkar, 3 <-Index Name--> aditya, 4 <-Index Name--> kartik.

Q9. Write a program to input an array and print it.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named `InputArrayWithoutreturnTypeAndArgument`. The code uses a `Scanner` object to read an array size and elements from standard input, then prints the array elements to standard output. The right pane shows the `Console` window with the application's output. It prompts for the array size (5), then for each element (2, 3, 4, 3, 2), and finally prints the given array elements (2, 3, 4, 3, 2).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class InputArrayWithoutreturnTypeAndArgument
6 {
7     public void inputArray()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i] = sc.nextInt();
17        }
18        System.out.println("Given array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            System.out.println(a[i] + " ");
22        }
23    }
24    public static void main(String[] args)
25    {
26        // TODO Auto-generated method stub
27        InputArrayWithoutreturnTypeAndArgument in = new InputArrayWithoutreturnTypeAndArgument();
28        in.inputArray();
29    }
30}
31
32
```

```
<terminated> InputArrayWithoutreturnTypeAndArgument [Java Application]
Enter array size :
5
Enter array element:
2 3 4 3 2
Given array element is:
2
3
4
3
2
```

Q10. Write a program initialize an array and print it.

The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named `InitializeArrayWithoutreturnTypeAndWithArgument`. It contains a method `ArrayInitializeArgument` that prints the elements of an array passed as an argument. The `main` method initializes an array with values 1, 2, 3, 21, 23, 11, and calls the `ArrayInitializeArgument` method. The right pane shows the `Console` window with the application's output. It first prints "Array initialized in main Method", then lists the array elements (1, 2, 3, 21, 23, 11), followed by "Given array element with argument", and then lists the array elements again (1, 2, 3, 21, 23, 11).

```
1 package usingmethodArray;
2
3 public class InitializeArrayWithoutreturnTypeAndWithArgument
4 {
5     public void ArrayInitializeArgument(int a[])
6     {
7         System.out.println("Given array element with argument");
8         for(int i=0;i<a.length;i++)
9         {
10             System.out.println(a[i] + " ");
11         }
12     }
13     public static void main(String[] args)
14     {
15         // TODO Auto-generated method stub
16         InitializeArrayWithoutreturnTypeAndWithArgument ia = new InitializeArrayWithoutreturnTypeAndWithArgument();
17         int a[] = {1, 2, 3, 21, 23, 11};
18         System.out.println("Array initialized in main Method");
19         for(int i=0;i<a.length;i++)
20         {
21             System.out.println(a[i] + " ");
22         }
23         ia.ArrayInitializeArgument(a);
24     }
25
26 }
```

```
<terminated> InitializeArrayWithoutreturnTypeAndWithArgument [Java Application]
Array initialized in main Method
1
2
3
21
23
11
Given array element with argument
1
2
3
21
23
11
```

Q11. Write a program to input 10 elements and multiply 5th elements by 25.

The screenshot shows a Java code editor and a terminal window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class FifthEleMulWithArg
6 {
7     public void printMulEle(int a[])
8     {
9         for(int i=0;i<a.length;i++)
10        {
11            if((i+1)==5)
12            {
13                a[i]=a[i]*25;
14            }
15        }
16        System.out.println("Given changed array element is: ");
17        for(int i=0;i<a.length;i++)
18        {
19            System.out.println(a[i]+" ");
20        }
21    }
22    public static void main(String[] args)
23    {
24        // TODO Auto-generated method stub
25        FifthEleMulWithArg al = new FifthEleMulWithArg();
26        Scanner sc = new Scanner(System.in);
27        System.out.println("Enter array size :");
28        int size = sc.nextInt();
29        int a[]=new int[size];
30        System.out.println("Enter array element: ");
31        for(int i=0;i<a.length;i++)
32        {
33            a[i]=sc.nextInt();
34        }
35        System.out.println("Given array element is: ");
36        for(int i=0;i<a.length;i++)
37        {
38            System.out.println(a[i]+" ");
39        }
40        al.printMulEle(a);
41    }
42 }
```

The terminal window shows the execution of the program. It prompts for the array size (5), then for each of the 5 elements (3, 6, 2, 7, 2). It then prints the original array elements, followed by the modified array where the 5th element is multiplied by 25 (3, 6, 2, 7, 50).

Q12. Write a program to input an array and find its average of first 5 element.

The screenshot shows a Java code editor and a terminal window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class AvgOFFiveArrayWithoutreturnTypeAndWithArgument
6 {
7     public void averageOFFive(int a[])
8     {
9         int sum=0;
10        System.out.println("Given array element is: ");
11        for(int i=0;i<a.length-3;i++)
12        {
13            sum+=a[i];
14            System.out.println(a[i]+" ");
15        }
16        System.out.println("Given Average of array element is: "+(sum/5));
17    }
18    public static void main(String[] args)
19    {
20        // TODO Auto-generated method stub
21        AvgOFFiveArrayWithoutreturnTypeAndWithArgument a = new AvgOFFiveArrayWithoutreturnTypeAndWithArgument();
22        Scanner sc = new Scanner(System.in);
23        System.out.println("Enter array size :");
24        int size = sc.nextInt();
25        int a1[]=new int[size];
26        System.out.println("Enter array element: ");
27        for(int i=0;i<a1.length;i++)
28        {
29            a1[i]=sc.nextInt();
30        }
31        a.averageOFFive(a1);
32    }
33 }
34 }
```

The terminal window shows the execution of the program. It prompts for the array size (8), then for each of the 8 elements (1, 2, 3, 4, 5, 6, 7, 8). It then prints the original array elements, followed by the average of the first 5 elements (3).

Q16. Write a program enter an array and print the square of the element which is present at odd position.

The screenshot shows a Java IDE interface with two tabs open: 'PrintSquareInOddPosiWithoutReturnTypeAndArg' and 'Console'. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintSquareInOddPosiWithoutReturnTypeAndArg
6 {
7     public void squareOdd()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element : ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.print("\nGiven square of odd position array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            if((i+1)%2!=0)
22            {
23                a[i]*=a[i];
24                System.out.print(a[i]+" ");
25            }
26        }
27    }
28    public static void main(String[] args)
29    {
30        // TODO Auto-generated method stub
31        PrintSquareInOddPosiWithoutReturnTypeAndArg p = new PrintSquareInOddPosiWithoutReturnTypeAndArg();
32        p.squareOdd();
33    }
34
35 }
36
```

The console output shows the user entering an array size of 5 and elements 2, 7, 8, 4, 1. The program then prints the squares of the odd-positioned elements (2, 8, 4) as 4, 64, 16 respectively.

Q17. Write a program to enter an array and print the square of the element which is present at even position.

The screenshot shows a Java IDE interface with two tabs open: 'PrintSquareInEvenPosiWithArg' and 'Console'. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintSquareInEvenPosiWithArg
6 {
7
8     public void EvenPositionCountArrEle(int a[])
9     {
10        System.out.println("Given square of even position array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            if((i+1)%2==0)
14            {
15                a[i]*=a[i];
16                System.out.println(a[i]+" ");
17            }
18        }
19    }
20    public static void main(String[] args) {
21        // TODO Auto-generated method stub
22        PrintSquareInEvenPosiWithArg o = new PrintSquareInEvenPosiWithArg();
23        Scanner sc = new Scanner(System.in);
24        System.out.println("Enter array size :");
25        int size = sc.nextInt();
26        int a[] = new int[size];
27        System.out.println("Enter array element : ");
28        for(int i=0;i<a.length;i++)
29        {
30            a[i]=sc.nextInt();
31        }
32        o.EvenPositionCountArrEle(a);
33    }
34
35 }
36
```

The console output shows the user entering an array size of 6 and elements 2, 7, 5, 4, 3, 2. The program then prints the squares of the even-positioned elements (7, 5, 4, 2) as 49, 25, 16, 4 respectively.

Q18. Write a program to input two array and merge in third array.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class MergeArrayWithoutArgReturnType
6 {
7     public void mergeArr()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int size1 = sc.nextInt();
13        int a[] = new int[size];
14        int b[] = new int[size1];
15        int c[] = new int[a.length+b.length];
16        int x=0;
17        System.out.println("Enter 1 array element: ");
18        for(int i=0;i<a.length;i++)
19        {
20            a[i]=sc.nextInt();
21            c[i]=a[i];
22            x++;
23        }
24        System.out.println("Enter 2 array element: ");
25        for(int i=0;i<b.length;i++)
26        {
27            b[i]=sc.nextInt();
28            c[x++]=b[i];
29        }
30        System.out.println("Given two merged array element is: ");
31        for(int i=0;i<c.length;i++)
32        {
33            System.out.println(c[i]+" ");
34        }
35    }
36    public static void main(String[] args)
37    {
38        MergeArrayWithoutArgReturnType m = new MergeArrayWithoutArgReturnType();
39        m.mergeArr();
40    }
41 }
```

The console window shows the output of the program. It asks for the size of the first array (5), then the elements of the first array (1 5 3 7 8). It then asks for the size of the second array (6), then the elements of the second array (2 4 6 9 10 11). Finally, it prints the merged array (1 5 3 7 8 2 4 6 9 10 11).

Q19. a[]={10,20,30,40,50}, b[]={1,2,3,4,5}

output array=c[]={10,5,20,4,30,3,40,2,50,1}

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class MergeTwoArrayWithoutArgReturnType
6 {
7     public void mergeArr()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int size1 = sc.nextInt();
13        int a[] = new int[size];
14        int b[] = new int[size1];
15        int c[] = new int[a.length+b.length];
16        int x=0,y=0;
17        System.out.println("Enter 1 array element: ");
18        for(int i=0;i<a.length;i++)
19        {
20            a[i]=sc.nextInt();
21        }
22        System.out.println("Enter 2 array element: ");
23        for(int i=b.length-1;i>=0;i--)
24        {
25            b[i]=sc.nextInt();
26        }
27        for(int i=0;i<c.length;i++)
28        {
29            if(i%2==0)
30            {
31                c[i]=a[x];
32                x++;
33            }
34            else
35            {
36                c[i]=b[y];
37                y++;
38            }
39        }
40    }
41 }
```

The console window shows the output of the program. It asks for the size of the first array (5) and the second array (5). It then asks for the elements of the first array (10 20 30 40 50) and the second array (1 2 3 4 5). Finally, it prints the merged array (10 5 20 4 30 3 40 2 50 1).

```

1 int size = sc.nextInt();
2 int size1 = sc.nextInt();
3 int a[] = new int[size];
4 int b[] = new int[size1];
5 int c[] = new int[a.length+b.length];
6 int x=0,y=0;
7 System.out.println("Enter 1 array element: ");
8 for(int i=0;i<a.length;i++)
9 {
10     a[i]=sc.nextInt();
11 }
12 System.out.println("Enter 2 array element: ");
13 for(int i=b.length-1;i>=0;i--)
14 {
15     b[i]=sc.nextInt();
16 }
17 for(int i=0;i<c.length;i++)
18 {
19     if(i%2==0)
20     {
21         c[i]=a[x];
22         x++;
23     }
24     else
25     {
26         c[i]=b[y];
27         y++;
28     }
29 }
30 System.out.println("Given two merged array element is: ");
31 for(int i=0;i<c.length;i++)
32 {
33     System.out.println(c[i]+" ");
34 }
35 }
36 public static void main(String[] args)
37 {
38     MergeTwoArrayWithoutArgReturnType m = new MergeTwoArrayWithoutArgReturnType();
39     m.mergeArr();
40 }
41 }

```

Console Output:

```

<terminated> MergeTwoArrayWithoutArgReturnType [Java Application]
Enter array size :
5 5
Enter 1 array element:
10 20 30 40 50
Enter 2 array element:
1 2 3 4 5
Given two merged array element is:
10
5
20
4
30
3
40
2
50
1

```

Q20. Write a program sort half array in ascending and half in descending order

input= int [] a={9,1,3,5,6,11,22,66,10,19}. output={1,3,5,6,9,10,66,22,19,11,10}

```

1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class HalfAsceAndDesceWithoutArgReturnType
6 {
7     public void halfAscendinAndDescending()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size : ");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter 1 array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        for(int i=0;i<a.length/2;i++)
19        {
20            for(int j=i+1;j<a.length;j++)
21            {
22                if(a[i]>a[j])
23                {
24                    int t=a[i];
25                    a[i]=a[j];
26                    a[j]=t;
27                }
28            }
29        }
30        for(int i=a.length/2;i<a.length;i++)
31        {
32            for(int j=i+1;j<a.length;j++)
33            {
34                if(a[i]<a[j])
35                {
36                    int t=a[i];
37                    a[i]=a[j];
38                    a[j]=t;
39                }
40            }
41        }
42        System.out.println("Given half ascending and descending array element is: ");
43    }
44 }

```

Console Output:

```

<terminated> HalfAsceAndDesceWithoutArgReturnType [Java Application] C:\Users\DELL\OneDrive\Desktop\Java\Assignment\Assignment 1\src
Enter array size :
10
Enter 1 array element:
9 1 3 5 6 11 22 66 10 19
Given half ascending and descending array element is:
1
3
5
6
9
10
66
22
19
11
10

```

The screenshot shows a Java development environment with multiple tabs open. The active tab contains Java code for sorting an array. The code uses two nested loops to compare elements and swap them if they are in the wrong order. It then prints the sorted array to the console.

```
16     a[i]=sc.nextInt();
17   }
18   for(int i=0;i<a.length/2;i++)
19   {
20       for(int j=i+1;j<a.length;j++)
21       {
22           if(a[i]>a[j])
23           {
24               int t=a[i];
25               a[i]=a[j];
26               a[j]=t;
27           }
28       }
29   }
30   for(int i=a.length/2;i<a.length;i++)
31   {
32       for(int j=i+1;j<a.length;j++)
33       {
34           if(a[i]<a[j])
35           {
36               int t=a[i];
37               a[i]=a[j];
38               a[j]=t;
39           }
40       }
41   }
42   System.out.println("Given half ascending and descending array element is: ");
43   for(int i=0;i<a.length;i++)
44   {
45       System.out.println(a[i]+" ");
46   }
47 }
48
49 public static void main(String[] args)
50 {
51     // TODO Auto-generated method stub
52     HalfAscAndDesceWithoutArgReturnType h = new HalfAscAndDesceWithoutArgReturnType();
53     h.halfAscendinAndDescending();
54 }
55
56 }
57 }
```

The console window displays the input and output of the program. It asks for the array size (10), then prompts for 10 integer values: 9, 1, 3, 5, 6, 11, 22, 66, 10, 19. The output shows the sorted array: 1, 3, 5, 6, 9, 10, 11, 19, 22, 66.

```
<terminated> HalfAscAndDesceWithoutArgReturnType [Java Application] C:\U
Enter array size :
10
Enter 10 array element:
9 1 3 5 6 11 22 66 10 19
Given half ascending and descending array element is
1
3
5
6
9
10
11
19
22
66
```

4th Assignment

Q1. Write a program to check if an array of integers without 0 and -1.

The image displays two side-by-side Java code editors. Both editors show the same Java file, `CheckZeroOneWithArg.java`, with line numbers and syntax highlighting. The code is designed to check if an array contains only 0s and -1s. It uses a scanner to read the array size and elements, then iterates through the array to count zeros and ones. If the count of zeros is greater than one, it prints an error message. If the count of ones is greater than zero, it prints another error message. Otherwise, it prints a success message.

Top Editor (Original Code):

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class CheckZeroOneWithArg
6 {
7
8     public void checkArr(int a[])
9     {
10         int c=0,x=0;
11         for(int i=0;i<a.length;i++)
12         {
13             if(a[i]==0)
14             {
15                 c++;
16             }
17             if(a[i]==-1)
18             {
19                 x++;
20             }
21         }
22         if(c>0)
23         {
24             System.out.println("\ngiven array contain "+c+" <-: 0");
25         }
26         if(x>0)
27         {
28             System.out.print("\ngiven array contain "+x+" <-: -1");
29         }
30         else
31         {
32             System.out.println("\ngiven array not contain "+x+" <-: -1 And "+x+" <-: 0");
33         }
34     }
35     public static void main(String[] args)
36     {
37         // TODO Auto-generated method stub
38         CheckZeroOneWithArg al = new CheckZeroOneWithArg();
39         Scanner sc = new Scanner(System.in);
40         System.out.println("Enter array size : ");
41         int size = sc.nextInt();
42         int a[] = new int[size];
43
44         c++;
45         if(a[i]==-1)
46         {
47             x++;
48         }
49         if(c>0)
50         {
51             System.out.println("\ngiven array contain "+c+" <-: 0");
52         }
53         if(x>0)
54         {
55             System.out.print("\ngiven array contain "+x+" <-: -1");
56         }
57         else
58         {
59             System.out.println("\ngiven array not contain "+x+" <-: -1 And "+x+" <-: 0");
60         }
61     }
62 }
```

Bottom Editor (Corrected Code):

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class CheckZeroOneWithArg
6 {
7
8     public void checkArr(int a[])
9     {
10         int c=0,x=0;
11         for(int i=0;i<a.length;i++)
12         {
13             if(a[i]==0)
14             {
15                 x++;
16             }
17             if(a[i]==-1)
18             {
19                 c++;
20             }
21         }
22         if(c>0)
23         {
24             System.out.println("\ngiven array contain "+c+" <-: 0");
25         }
26         if(x>0)
27         {
28             System.out.print("\ngiven array contain "+x+" <-: -1");
29         }
30         else
31         {
32             System.out.println("\ngiven array not contain "+x+" <-: -1 And "+x+" <-: 0");
33         }
34     }
35     public static void main(String[] args)
36     {
37         // TODO Auto-generated method stub
38         CheckZeroOneWithArg al = new CheckZeroOneWithArg();
39         Scanner sc = new Scanner(System.in);
40         System.out.println("Enter array size : ");
41         int size = sc.nextInt();
42         int a[] = new int[size];
43         System.out.println("Enter array element: ");
44         for(int i=0;i<a.length;i++)
45         {
46             a[i]=sc.nextInt();
47         }
48         System.out.print("Given array element is: ");
49         for(int i=0;i<a.length;i++)
50         {
51             System.out.print(a[i]+" ");
52         }
53         al.checkArr(a);
54     }
55 }
```

The right side of each editor shows the terminal output. In both cases, the user enters an array size of 5 and an array containing -1, 0, 0, -1, 4. The top editor's output shows errors for both zero and one counts, while the bottom editor's output correctly identifies the array as containing only -1 and 0.

Q2. Write a Java program to remove the duplicate elements of a given array and print the new length of the array. Sample array: [20, 20, 30, 40, 50, 50, 50]. After removing the duplicate elements the program should return 4 as the new length of the array.

```

1 package usingmethodarray;
2 import java.util.Scanner;
3 public class RemoveDuplicateWithArg
4 {
5     public void removeDuplicate(int a[])
6     {
7         for(int i=0;i<a.length;i++)
8         {
9             for(int j=i+1;j<a.length;j++)
10            {
11                if(a[i]==a[j])
12                {
13                    a[j]=-1;
14                }
15            }
16        }
17        System.out.print("\nNew lwngh array element is: ");
18        for(int i=0;i<a.length;i++)
19        {
20            if(a[i]!=-1)
21            {
22                System.out.print(a[i]+" ");
23            }
24        }
25    }
26    public static void main(String[] args)
27    {
28        RemoveDuplicateWithArg al = new RemoveDuplicateWithArg();
29        Scanner sc = new Scanner(System.in);
30        System.out.println("Enter array size :");
31        int size = sc.nextInt();
32        int a[] = new int[size];
33        System.out.println("Enter array element: ");
34        for(int i=0;i<a.length;i++)
35        {
36            a[i]=sc.nextInt();
37        }
38        System.out.print("Given array element is: ");
39        for(int i=0;i<a.length;i++)
40        {
41            System.out.print(a[i]+" ");
42        }
43        al.removeDuplicate(a);
44    }
45 }

```

The console output shows the program's execution:

```

<terminated> RemoveDuplicateWithArg [Java Application] C:\Users\Shree
Enter array size :
7
Enter array element:
20 20 30 40 50 50 50
Given array element is: 20 20 30 40 50 50 50
New lwngh array element is: 20 30 40 50

```

Q3. Write a Java program to find the sum of the two elements of a given array which is equal to a given integer. Sample array: [1, 2, 4, 5, 6] Target value: 6.

```

1 package usingmethodarray;
2
3 import java.util.Scanner;
4
5 public class SumOfInteWithArg
6 {
7
8     public void sumOfInt(int a[])
9     {
10        Scanner sc = new Scanner(System.in);
11        System.out.println("\nEnter k number :");
12        int k=sc.nextInt();
13        int sum=0;
14        System.out.println("\nGiven addition of targeted value is :");
15        for(int i=0;i<a.length;i++)
16        {
17            for(int j=i+1;j<a.length;j++)
18            {
19                sum=a[i]+a[j];
20                if(sum==k)
21                {
22                    System.out.println(a[i]+"<"+a[j]+"< "+k);
23                }
24            }
25        }
26    }
27    public static void main(String[] args)
28    {
29        SumOfInteWithArg al = new SumOfInteWithArg();
30        Scanner sc = new Scanner(System.in);
31        System.out.println("Enter array size :");
32        int size = sc.nextInt();
33        int a[] = new int[size];
34        System.out.println("Enter array element: ");
35        for(int i=0;i<a.length;i++)
36        {
37            a[i]=sc.nextInt();
38        }
39        System.out.print("Given array element is: ");
40        for(int i=0;i<a.length;i++)
41        {
42            System.out.print(a[i]+" ");
43        }
44        al.sumOfInt(a);
45    }
46 }

```

The console output shows the program's execution:

```

<terminated> SumOfInteWithArg [Java Application] C:\Users\Shree
Enter array size :
5
Enter array element:
1 2 4 5 6
Given array element is: 1 2 4 5 6
Enter k number :
6
Given addition of targeted value is :
1+5 = 6
2+4 = 6

```

Q4. Write a Java program to print all the LEADERS in the array.

Note: An element is leader if it is greater than all the elements to its right side.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for finding leaders in an array. The code uses a Scanner to read the array size and elements, then iterates through the array to find elements that are greater than all elements to their right. The output window shows the input size (6), elements (10 17 3 4 5 2), and the resulting leader array (17 5 2).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class LeaderWithReturnType
6 {
7     public int[] leader()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element : ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.print("Given array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            System.out.print(a[i]+" ");
22        }
23        for(int i=0;i<a.length;i++)
24        {
25            for(int j=i+1;j<a.length;j++)
26            {
27                if(a[i]< a[j])
28                {
29                    a[i]=-1;
30                }
31            }
32        }
33        System.out.println("\nGiven leader array element is: ");
34        for(int i=0;i<a.length;i++)
35        {
36            if(a[i]!=-1)
37            {
38                System.out.println(a[i]);
39            }
40        }
41        return a;
42    }
43    public static void main(String[] args)
44    {
45        // TODO Auto-generated method stub
46        LeaderWithReturnType l = new LeaderWithReturnType();
47        l.leader();
48    }
49}
```

Q5. Write a Java program to check if an array of integers contains two specified elements 65 and 77.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for checking if an array contains two specific elements (65 and 77). The code uses a Scanner to read the array size and elements, then counts occurrences of each element. The output window shows the input size (5), elements (65 77 5 53 3), and the counts for each element (65: 2, 77: 1).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class CheckTwoEleWithReturnType
6 {
7     public int[] CheckArrayContainsOrNot()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.println("Enter two element :");
19        int n= sc.nextInt();
20        int nl= sc.nextInt();
21        int c=0, cl=0, c2=0, i=0;
22        for(i=0;i<a.length;i++)
23        {
24            if(a[i]==n)
25            {
26                c++;
27            }
28            if(a[i]==nl)
29            {
30                cl++;
31            }
32            else if(a[i]!=n || a[i]!=nl)
33            {
34                c2++;
35            }
36        }
37        if(c>0)
38        {
39            System.out.println(n+" given element is Present");
40        }
41        if(cl>0)
```

```

19     int n= sc.nextInt();
20     int nl= sc.nextInt();
21     int c=0,cl=0,c2=0,i=0;
22     for(i=0;i<a.length;i++)
23     {
24         if(a[i]==n)
25         {
26             c++;
27         }
28         if(a[i]==nl)
29         {
30             cl++;
31         }
32         else if(a[i]!=n || a[i]!=nl)
33         {
34             c2++;
35         }
36     }
37     if(c>0)
38     {
39         System.out.println(n+" given element is Present");
40     }
41     if(cl>0)
42     {
43         System.out.println(nl+" given element is Present");
44     }
45     else if(c2==0)
46     {
47         System.out.println("Both are not present");
48     }
49     return a;
50 }
51 public static void main(String[] args)
52 {
53     // TODO Auto-generated method stub
54     CheckTwoEleWithReturnType s = new CheckTwoEleWithReturnType();
55     s.CheckArrayConatainsOrNot();
56 }
57 }
58 }

```

The console output shows the program's execution:

```

<terminated> CheckTwoEleWithReturnType [Java Application]
Enter array size :
5
Enter array element:
65 77 5 53 3
Enter two element :
65 77
65 given element is Present
77 given element is Present

```

Q6. Write a Java program to separate even and odd numbers of a given array of integers. Put all even numbers first, and then odd numbers.

```

1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PutEvenFirstThenOddWithReturnType
6 {
7     public int[] checkEvenOdd()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        for(int i=0;i<a.length;i++)
19        {
20            for(int j=i+1;j<a.length;j++)
21            {
22                if(a[i]%2!=0)
23                {
24                    int t=a[i];
25                    a[i]=a[j];
26                    a[j]=t;
27                }
28            }
29        }
30        System.out.println("Given array element in first even order then odd: ");
31        for(int i=0;i<a.length;i++)
32        {
33            System.out.println(a[i]+" ");
34        }
35        return a;
36    }
37    public static void main(String[] args)
38    {
39        PutEvenFirstThenOddWithReturnType s = new PutEvenFirstThenOddWithReturnType();
40        s.checkEvenOdd();
41    }
}

```

The console output shows the program's execution:

```

<terminated> PutEvenFirstThenOddWithReturnType [Java Application] C:\Users
Enter array size :
5
Enter array element:
2 7 5 8 6
Given array element in first even order then odd:
2
8
6
5
7

```

Q7. Write a java program to find prime number between an arrays of element.

```
import java.util.Scanner;

public class FindPrimeWithReturnType
{
    public int[] checkPrime()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter array size :");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.println("Enter array element: ");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("Given prime array element is: ");
        for(int i=0;i<a.length;i++)
        {
            int c=0;
            int n=a[i];
            for(int j=1;j<=n;j++)
            {
                if(n%j==0)
                {
                    c++;
                }
            }
            if(c==2)
            {
                System.out.println(n+" ");
            }
        }
        return a;
    }
    public static void main(String[] args)
    {
        FindPrimeWithReturnType s = new FindPrimeWithReturnType();
        s.checkPrime();
    }
}
```

The screenshot shows a terminal window titled 'terminated> FindPrimeWithReturnType.java Application'. It displays the following interaction:

```
Enter array size :  
5  
Enter array element:  
1 2 7 9 5  
Given prime array element is:  
2  
7  
5
```

Q1.Given an array and a number K where K is smaller than the size of the array. Find the K'th smallest element in the given array. Given that all array elements are distinct.

Examples: Input: arr[] = {7, 10, 4, 3, 20, 15}, K = 3 Output: 7

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class KthUnsortedArrayFindWithArg
6 {
7
8     public void kUnsorted(int a[])
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter array size: ");
12         int k = sc.nextInt();
13         for(int i=0;i<a.length;i++)
14         {
15             for(int j=i+1;j<a.length;j++)
16             {
17                 if(a[i]>a[j])
18                 {
19                     int t=a[i];
20                     a[i]=a[j];
21                     a[j]=t;
22                 }
23             }
24         }
25         System.out.println("Kth element is: ");
26         for(int i=0;i<a.length;i++)
27         {
28             if((i+1)==k)
29             {
30                 System.out.println(a[i]);
31             }
32         }
33     }
34     public static void main(String[] args)
35     {
36         // TODO Auto-generated method stub
37         KthUnsortedArrayFindWithArg m = new KthUnsortedArrayFindWithArg();
38         Scanner sc = new Scanner(System.in);
39         System.out.println("Enter array size: ");
40         int size = sc.nextInt();
41         int a[]=new int[size];
42         System.out.println("Enter array element: ");
43         for(int i=0;i<a.length;i++)
44         {
45             a[i]=sc.nextInt();
46         }
47         m.kUnsorted(a);
48     }
49
50 }
```

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class KthUnsortedArrayFindWithArg
6 {
7
8     public void kUnsorted(int a[])
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter kth element: ");
12         int k = sc.nextInt();
13         for(int i=0;i<a.length;i++)
14         {
15             for(int j=i+1;j<a.length;j++)
16             {
17                 if(a[i]>a[j])
18                 {
19                     int t=a[i];
20                     a[i]=a[j];
21                     a[j]=t;
22                 }
23             }
24         }
25         System.out.println("Kth element is: ");
26         for(int i=0;i<a.length;i++)
27         {
28             if((i+1)==k)
29             {
30                 System.out.println(a[i]);
31             }
32         }
33     }
34     public static void main(String[] args)
35     {
36         // TODO Auto-generated method stub
37         KthUnsortedArrayFindWithArg m = new KthUnsortedArrayFindWithArg();
38         Scanner sc = new Scanner(System.in);
39         System.out.println("Enter array size: ");
40         int size = sc.nextInt();
41         int a[]=new int[size];
42         System.out.println("Enter array element: ");
43         for(int i=0;i<a.length;i++)
44         {
45             a[i]=sc.nextInt();
46         }
47         m.kUnsorted(a);
48     }
49
50 }
```

Console Output:

```
<terminated> KthUnsortedArrayFindWithArg [Java Application] C:\Users\...
Enter array size: 6
Enter array element: 7 10 4 3 20 15
Enter kth element: 3
Kth element is: 7
```

```
<terminated> KthUnsortedArrayFindWithArg [Java Application] C:\...
Enter array size: 6
Enter array element: 7 10 4 3 20 15
Enter kth element: 3
Kth element is: 7
```

Q2. Given an array of integers arr[], The task is to find the index of first repeating element in it i.e. the element that occurs more than once and whose index of the first occurrence is the smallest.

```

1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class RepeatingEleIndWithArg
6 {
7
8     public void repeatingEle(int a[])
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter array element: ");
12         int k = sc.nextInt();
13         for(int i=0;i<a.length;i++)
14         {
15             for(int j=i+1;j<a.length;j++)
16             {
17                 if(a[i]==a[j])
18                 {
19                     a[j]=-1;
20                 }
21             }
22         }
23         System.out.println("The index of first repeating array element is : ");
24         for(int i=0;i<a.length;i++)
25         {
26             if(a[i]!=-1)
27             {
28                 if(a[i]==k)
29                 System.out.println(i);
30             }
31         }
32     }
33     public static void main(String[] args)
34     {
35         // TODO Auto-generated method stub
36         RepeatingEleIndWithArg m = new RepeatingEleIndWithArg();
37         Scanner sc = new Scanner(System.in);
38         System.out.println("Enter array size: ");
39         int size = sc.nextInt();
40         int a[] = new int[size];
41         System.out.println("Enter array element: ");
42         for(int i=0;i<a.length;i++)
43         {
44             Scanner sc = new Scanner(System.in);
45             int k = sc.nextInt();
46             m.repeatingEle(a);
47         }
48     }
49 }

```

```

<terminated> RepeatingEleIndWithArg [Java Application] C:\Users\Shree\p2\pool\RepeatingEleIndWithArg.java
Enter array size:
6
Enter array element:
1 1 2 2 3 3
Enter kth element:
3
The index of first repeating array element is :
4

<terminated> RepeatingEleIndWithArg [Java Application] C:\Users\Shree\p2\pool\RepeatingEleIndWithArg.java
Enter array size:
6
Enter array element:
1 1 2 2 3 3
Enter kth element:
3
The index of first repeating array element is :
4

```

Q3. Find the majority element in the array. A majority element in an array $A[]$ of size n is an element that appears more than $n/2$ times (and hence there is at most one such element).

Examples : Input : {3, 3, 4, 2, 4, 4, 2, 4, 4}, **Output :** 4 **Explanation:** The frequency of 4 is 5 which is greater than the half of the size of the array size.

The screenshot shows a Java development environment with two tabs open: 'FindMajorityEleWithArg.java' and 'Console'. The code in 'FindMajorityEleWithArg.java' is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class FindMajorityEleWithArg
6 {
7
8     public void majorityEle(int a[],int size)
9     {
10         System.out.println("Given majority array element is: ");
11         for(int i=0;i<a.length;i++)
12         {
13             if(a[i]==(size/2))
14             {
15                 System.out.println(a[i]+" ");
16                 break;
17             }
18         }
19     }
20
21     public static void main(String[] args)
22     {
23         // TODO Auto-generated method stub
24         FindMajorityEleWithArg m = new FindMajorityEleWithArg();
25         Scanner sc = new Scanner(System.in);
26         System.out.println("Enter array size: ");
27         int size = sc.nextInt();
28         int a[] = new int[size];
29         System.out.println("Enter array element: ");
30         for(int i=0;i<a.length;i++)
31         {
32             a[i]=sc.nextInt();
33         }
34         m.majorityEle(a,size);
35     }
36 }
```

The 'Console' tab shows the output of running the program:

```
<terminated> FindMajorityEleWithArg [Java Application] C:\Users\Shree\p2\pool
Enter array size:
9
Enter array element:
3 3 4 2 4 4 2 4 4
Given majority array element is:
4
```

Q4. Given an array of N integers, and a number sum, the task is to find the number of pairs of integers in the array whose sum is equal to sum. Examples: Input: arr[] = {1, 5, 7, -1}, sum = 6, Output: 2

The screenshot shows a Java development environment with two tabs open: 'SumOfPairEleWithArg.java' and 'Console'. The code in 'SumOfPairEleWithArg.java' is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class SumOfPairEleWithArg
6 {
7
8     public void pairSum(int a[])
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter sum: ");
12         int k = sc.nextInt();
13         int sum=0,c=0;
14         System.out.println("Given sum pair array element is: ");
15         for(int i=0;i<a.length;i++)
16         {
17             for(int j=i+1;j<a.length;j++)
18             {
19                 sum=a[i]+a[j];
20                 if(sum==k)
21                 {
22                     System.out.println(a[i]+" "+a[j]+" ");
23                     c++;
24                 }
25             }
26         }
27         System.out.println("Count of sum pair is: "+c);
28     }
29
30     public static void main(String[] args)
31     {
32         SumOfPairEleWithArg m = new SumOfPairEleWithArg();
33         Scanner sc = new Scanner(System.in);
34         System.out.println("Enter array size: ");
35         int size = sc.nextInt();
36         int a[] = new int[size];
37         System.out.println("Enter array element: ");
38         for(int i=0;i<a.length;i++)
39         {
40             a[i]=sc.nextInt();
41         }
42         m.pairSum(a);
43     }
44 }
```

The 'Console' tab shows the output of running the program:

```
<terminated> SumOfPairEleWithArg [Java Application] C:\Users\Shree\p2\pool
Enter array size:
4
Enter array element:
1 5 7 -1
Enter sum:
6
Given sum pair array element is:
1 5
7 -1
Count of sum pair is: 2
```

Q5. Given an array and a value, find if there is a triplet in array whose sum is equal to the given value. If there is such a triplet present in array, then print the triplet and return true. Else return false. Examples: Input: array = {12, 3, 4, 1, 6, 9}, sum = 24, Output: 12, 3, 9

The screenshot shows an IDE interface with multiple tabs open. The active tab is "SumOfThreePairEleWithArg.java". The code implements a solution to find a triplet in an array whose sum equals a given value. It uses three nested loops to iterate through all possible triplets and prints them if their sum matches the target value. The main method creates an instance of the class and calls the triplet summing method. The console window shows the input size (6), the array elements (12, 3, 4, 1, 6, 9), the target sum (24), and the output (12 3 9) which is identified as a sum pair.

```
1 package usingmethodArray;
2 import java.util.Scanner;
3 public class SumOfThreePairEleWithArg
4 {
5     public void threePairSum(int a[])
6     {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Enter sum: ");
9         int k1 = sc.nextInt();
10        int sum=0,c=0;
11        System.out.println("Given sum pair array element is: ");
12        for(int i=0;i<a.length;i++)
13        {
14            for(int j=i+1;j<a.length;j++)
15            {
16                for(int k=j+1;k<a.length;k++)
17                {
18                    sum=a[i]+a[j]+a[k];
19                    if(sum==k1)
20                    {
21                        System.out.println(a[i]+" "+a[j]+" "+a[k]+" ");
22                        c++;
23                    }
24                }
25            }
26        }
27        System.out.println("Count of sum pair is: "+c);
28    }
29    public static void main(String[] args)
30    {
31        SumOfThreePairEleWithArg m = new SumOfThreePairEleWithArg();
32        Scanner sc = new Scanner(System.in);
33        System.out.println("Enter array size: ");
34        int size = sc.nextInt();
35        int a[] = new int[size];
36        System.out.println("Enter array element: ");
37        for(int i=0;i<a.length;i++)
38        {
39            a[i]=sc.nextInt();
40        }
41        m.threePairSum(a);
42    }
}
```

Q3.a[]={10,20,30,40,50}, b[]={1,2,3,4,5}

output array=c[]={10,5,20,4,30,3,40,2,50,1}

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class MergeTwoArrWithArg
6 {
7
8     public void mergeArr(int a[],int b[])
9     {
10
11         int c[]=new int[a.length+b.length];
12         int x=0,y=0;
13         for(int i=0;i<c.length;i++)
14         {
15             if(i%2==0)
16             {
17                 c[i]=a[x];
18                 x++;
19                 []
20             }
21             else
22             {
23                 c[i]=b[y];
24                 y++;
25             }
26         }
27         System.out.println("Given merged array element is:");
28         for(int i=0;i<c.length;i++)
29         {
30             System.out.println(c[i]+" ");
31         }
32     }
33     public static void main(String[] args)
34     {
35         MergeTwoArrWithArg m = new MergeTwoArrWithArg();
36         Scanner sc = new Scanner(System.in);
37         System.out.println("Enter array size: ");
38         int size = sc.nextInt();
39         int a[]=new int[size];
40         int b[]=new int[size];
41         System.out.println("Enter 1 array element: ");
42         for(int i=0;i<a.length;i++)
43         {
44
45         }
46         System.out.println("Enter 2 array element: ");
47         for(int i=b.length-1;i>=0;i--)
48         {
49             b[i]=sc.nextInt();
50         }
51         m.mergeArr(a,b);
52     }
}

```

Console X <terminated> MergeTwoArrWithArg [Java Application] C:\

Enter array size:
5
Enter 1 array element:
10 20 30 40 50
Enter 2 array element:
1 2 3 4 5
Given merged array element is:
10
5
20
4
30
3
40
2
50
1

Console X <terminated> MergeTwoArrWithArg [Java Application] C:\Users\Shree\p2\p

Enter array size:
5
Enter 1 array element:
10 20 30 40 50
Enter 2 array element:
1 2 3 4 5
Given merged array element is:
10
5
20
4
30
3
40
2
50
1

Q4. Write a program to input an array now delete element from array, element is taken from user.

The screenshot shows an IDE interface with three tabs: "SumOfThreePairEleWithArg.java", "MergeTwoArrWithArg.java", and "DeleteArrWithArg.java". The "DeleteArrWithArg.java" tab is active, displaying the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class DeleteArrWithArg
6 {
7     public void deleteArrEle(int a[])
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter an element: ");
11        int k1 = sc.nextInt();
12        int sum=0,g=0;
13        System.out.println("Deleted array element is: ");
14        for(int i=0;i<a.length;i++)
15        {
16            if(a[i]==k1)
17            {
18                a[i]=-1;
19            }
20            for(int i=0;i<a.length;i++)
21            {
22                if(a[i]!=-1)
23                {
24                    System.out.println(a[i]+" ");
25                }
26            }
27        }
28        public static void main(String[] args)
29        {
30            DeleteArrWithArg m = new DeleteArrWithArg();
31            Scanner sc = new Scanner(System.in);
32            System.out.println("Enter array size: ");
33            int size = sc.nextInt();
34            int a[] = new int[size];
35            System.out.println("Enter array element: ");
36            for(int i=0;i<a.length;i++)
37            {
38                a[i]=sc.nextInt();
39            }
40            m.deleteArrEle(a);
41        }
42    }
```

To the right of the code is a "Console" window showing the output of the program. It prompts the user for the array size (6), then for an element to delete (4). The program then prints the deleted array element (4) and the remaining elements of the array: 1, 1, 2, 4, 35, 5.

Q5. Write a program to input an array now delete element from array, position is taken from user.

The screenshot shows an IDE interface with three tabs: "SumOfThreePairEleWithArg.java", "MergeTwoArrWithArg.java", and "DeleteArrWithArg.java". The "DeleteArrWithArg.java" tab is active, displaying the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class DeleteArrWithArg
6 {
7
8    public void deleteArrEle(int a[])
9    {
10        Scanner sc = new Scanner(System.in);
11        System.out.println("Enter an element: ");
12        int k1 = sc.nextInt();
13        int sum=0,g=0;
14        System.out.println("Deleted array element is: ");
15        for(int i=k1-1;i<a.length-1;i++)
16        {
17            a[k1]=a[i+1];
18        }
19        for(int i=0;i<a.length-1;i++)
20        {
21            System.out.println(a[i]+" ");
22        }
23    }
24    public static void main(String[] args)
25    {
26        DeleteArrWithArg m = new DeleteArrWithArg();
27        Scanner sc = new Scanner(System.in);
28        System.out.println("Enter array size: ");
29        int size = sc.nextInt();
30        int a[] = new int[size];
31        System.out.println("Enter array element: ");
32        for(int i=0;i<a.length;i++)
33        {
34            a[i]=sc.nextInt();
35        }
36        m.deleteArrEle(a);
37    }
38 }
```

To the right of the code is a "Console" window showing the output of the program. It prompts the user for the array size (7), then for an element to delete (2). The program then prints the deleted array element (2) and the remaining elements of the array: 3, 6, 3, 6, 2, 5, 3.

Q5. Write a program to input an array and rotate it in anti-clock wise by any no given by user.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for rotating an array anti-clockwise. The code uses a Scanner to read the array size, elements, and the number of rotations. It then performs the rotation and prints the result. The right side of the interface shows the console output where the user enters the array size (5), elements (1 2 3 4 5), and the number of rotations (1). The output shows the anti-clockwise rotated array (5 1 2 3 4).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class RotateAntiClockWiseArrWithArg
6 {
7     public void antiClockWise(int a[])
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter n number: ");
11        int k1 = sc.nextInt();
12        int sum=0;
13        System.out.println("Anti-Clock wise array element is: ");
14        for(int i1=1;i1<=k1;i1++)
15        {
16            int temp=a[0];
17            for(int i=0;i<a.length-1;i++)
18            {
19                a[i]=a[i+1];
20            }
21            a[a.length-1]=temp;
22        }
23        for(int i=0;i<a.length;i++)
24        {
25            System.out.println(a[i]+" ");
26        }
27    }
28    public static void main(String[] args)
29    {
30        RotateAntiClockWiseArrWithArg m = new RotateAntiClockWiseArrWithArg();
31        Scanner sc = new Scanner(System.in);
32        System.out.println("Enter array size: ");
33        int size = sc.nextInt();
34        int a[]=new int[size];
35        System.out.println("Enter array element: ");
36        for(int i=0;i<a.length;i++)
37        {
38            a[i]=sc.nextInt();
39        }
40        m.antiClockWise(a);
41    }
42 }
```

Q6. Write a program to input an array and rotate it in clock wise by any no given by user.

The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for rotating an array clockwise. The code uses a Scanner to read the array size, elements, and the number of rotations. It then performs the rotation and prints the result. The right side of the interface shows the console output where the user enters the array size (5), elements (1 2 3 4 5), and the number of rotations (2). The output shows the clockwise rotated array (4 5 1 2 3).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class RotateClockWiseArrWithArg
6 {
7     public void clockWise(int a[])
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter n number: ");
11        int k1 = sc.nextInt();
12        int sum=0;
13        System.out.println("Clock wise array element is: ");
14        for(int i1=1;i1<=k1;i1++)
15        {
16            int temp=a[a.length-1];
17            for(int i=a.length-1;i>0;i--)
18            {
19                a[i]=a[i-1];
20            }
21            a[0]=temp;
22        }
23        for(int i=0;i<a.length;i++)
24        {
25            System.out.println(a[i]+" ");
26        }
27    }
28    public static void main(String[] args)
29    {
30        RotateClockWiseArrWithArg m = new RotateClockWiseArrWithArg();
31        Scanner sc = new Scanner(System.in);
32        System.out.println("Enter array size: ");
33        int size = sc.nextInt();
34        int a[]=new int[size];
35        System.out.println("Enter array element: ");
36        for(int i=0;i<a.length;i++)
37        {
38            a[i]=sc.nextInt();
39        }
40        m.clockWise(a);
41    }
42 }
```

Q7. Write a program to input an array and delete all duplicate element from array.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class DeleteDuplicateArrWithArg
6 {
7     public void deleteDuplicateArrEle(int a[])
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Deleted all duplicate array element is: ");
11        for(int i=0;i<a.length;i++)
12        {
13            for(int j=i+1;j<a.length;j++)
14            {
15                if(a[i]==a[j])
16                {
17                    a[j]=-1;
18                }
19            }
20            for(int i=0;i<a.length;i++)
21            {
22                if(a[i]==-1)
23                {
24                    System.out.println(a[i] + " ");
25                }
26            }
27        }
28    }
29    public static void main(String[] args)
30    {
31        DeleteDuplicateArrWithArg m = new DeleteDuplicateArrWithArg();
32        Scanner sc = new Scanner(System.in);
33        System.out.print("Enter array size: ");
34        int size = sc.nextInt();
35        int a[] = new int[size];
36        System.out.print("Enter array element: ");
37        for(int i=0;i<a.length;i++)
38        {
39            a[i]=sc.nextInt();
40        }
41        m.deleteDuplicateArrEle(a);
42    }
43 }
```

The console window shows the output of the program. It prompts for the array size (10), then for array elements (1 1 2 2 2 3 3 3 4 4), and then prints the deleted array elements (1 2 3 4).

Q8. Write a Java program to find max number in an array.

The screenshot shows an IDE interface with a code editor and a console window. The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class MaxArrayWithReturnTypeAndArg
6 {
7     public int[] max(int a[])
8     {
9         int max=0;
10        for(int i=0;i<a.length;i++)
11        {
12            if(a[i]>max)
13            {
14                max=a[i];
15            }
16        }
17        System.out.println("Max array element is: "+max);
18        return a;
19    }
20    public static void main(String[] args)
21    {
22        MaxArrayWithReturnTypeAndArg al = new MaxArrayWithReturnTypeAndArg();
23        Scanner sc = new Scanner(System.in);
24        System.out.print("Enter array size : ");
25        int size = sc.nextInt();
26        int a[] = new int[size];
27        System.out.print("Enter array element: ");
28        for(int i=0;i<a.length;i++)
29        {
30            a[i]=sc.nextInt();
31        }
32        System.out.print("Given array element is: ");
33        for(int i=0;i<a.length;i++)
34        {
35            System.out.println(a[i] + " ");
36        }
37        al.max(a);
38    }
39 }
```

The console window shows the output of the program. It prompts for the array size (5), then for array elements (11 23 5 45 4), then prints the given array elements (11 23 5 45 4), and finally prints the maximum array element (45).

Q9. Write a program to input an array now insert any element at any position, element and position is taken by user.

The screenshot shows an IDE interface with several tabs open. The active tab is titled "InsertEleAtAnyPosi...". The code in the editor is as follows:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class InsertEleAtAnyPosiWithReturnTypeAndArg
6 {
7     public int[] insertEleAtGivenPosition(int a[])
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter elelement and position: ");
11        int ele = sc.nextInt();
12        int max = sc.nextInt();
13        for(int i=0;i<a.length;i++)
14        {
15            if((i+1)==max)
16            {
17                a[i]=ele;
18            }
19        }
20        System.out.println("given changed array element is: ");
21        for(int i=0;i<a.length;i++)
22        {
23            System.out.println(a[i]+" ");
24        }
25        return a;
26    }
27    public static void main(String[] args)
28    {
29        InsertEleAtAnyPosiWithReturnTypeAndArg al = new InsertEleAtAnyPosiWithReturnTypeAndArg();
30        Scanner sc = new Scanner(System.in);
31        System.out.println("Enter array size :");
32        int size = sc.nextInt();
33        int a[]=new int[size];
34        System.out.println("Enter array element: ");
35        for(int i=0;i<a.length;i++)
36        {
37            a[i]=sc.nextInt();
38        }
39        System.out.println("Given array element is: ");
40        for(int i=0;i<a.length;i++)
41        {
42            System.out.println(a[i]+" ");
43        }
44        al.insertEleAtGivenPosition(a);
45    }
46 }
```

The "Console" tab shows the output of the program. It prompts for the array size (6), then for elements (23 45 67 89 22 34), then for the element and position to insert (23 4). The output shows the original array followed by the inserted element (23 45 67 89 22 34).

Q1.Given an array of N integers. Find the first element that occurs at least K number of times. Example 1: Input :N = 7, K = 2, A[] = {1, 7, 4, 3, 4, 8, 7} Output :7

```

1 package usingmethodarray;
2
3 import java.util.Scanner;
4
5 public class PrintKthRepeatingEleWithArg
6 {
7
8     public void FindKthEle(int a[],int k)
9     {
10         System.out.println("The first element that occurs at least K number of times is: ");
11         for(int i=0;i<a.length;i++)
12         {
13             int c=0;
14             for(int j=i+1;j<a.length;j++)
15             {
16                 if(a[i]==a[j])
17                 {
18                     c++;
19                     break;
20                 }
21             }
22             if(c==k)
23             {
24                 System.out.println(a[i]+ " ");
25                 break;
26             }
27         }
28     }
29
30     public static void main(String[] args)
31     {
32         // TODO Auto-generated method stub
33         PrintKthRepeatingEleWithArg o = new PrintKthRepeatingEleWithArg();
34         Scanner sc = new Scanner(System.in);
35         System.out.println("Enter array size :");
36         int size = sc.nextInt();
37         int a[]=new int[size];
38         System.out.println("Enter array element: ");
39         for(int i=0;i<a.length;i++)
40         {
41             a[i]=sc.nextInt();
42         }
43         System.out.println("Enter K value: ");
44         int k = sc.nextInt();
45         o.FindKthEle(a,k);
46     }
47 }

```

Q2.Given an array A of positive integers. Your task is to find the leaders in the array. An element of array is leader if it is greater than or equal to all the elements to its right side. The rightmost element is always a leader. Example 1: Input: n = 6A[] = {16,17,4,3,5,2}
Output: 17 5 2

```

1 package usingmethodarray;
2
3 import java.util.Scanner;
4
5 public class LeaderEleWithArg
6 {
7
8     public void FindLeaderEle(int a[])
9     {
10         System.out.println("The leader array element is: ");
11         for(int i=0;i<a.length;i++)
12         {
13             int g=0;
14             for(int j=i+1;j<a.length;j++)
15             {
16                 if(a[i]<=a[j])
17                 {
18                     a[i]=-1;
19                 }
20             }
21         }
22         for(int i=0;i<a.length;i++)
23         {
24             if(a[i]!=-1)
25             {
26                 System.out.println(a[i]+ " ");
27             }
28         }
29     }
30
31     public static void main(String[] args)
32     {
33         // TODO Auto-generated method stub
34         LeaderEleWithArg o = new LeaderEleWithArg();
35         Scanner sc = new Scanner(System.in);
36         System.out.println("Enter array size :");
37         int size = sc.nextInt();
38         int a[]=new int[size];
39         System.out.println("Enter array element: ");
40         for(int i=0;i<a.length;i++)
41         {
42             a[i]=sc.nextInt();
43         }
44         o.FindLeaderEle(a);
45     }
46 }

```

Q3. Given an array A of n positive numbers. The task is to find the first Equilibrium Point in an array. Equilibrium Point in an array is a position such that the sum of elements before it is equal to the sum of elements after it. Note: Retun the index of Equilibrium point. (1-based index)

Example 1: Input: n = 5

A[] = {1,3,5,2,2} Output: 3

The screenshot shows a Java development environment with multiple tabs open. The active tab is 'EquilibriumPointWithArg.java'. The code implements a solution to find the first equilibrium point in an array. It uses two nested loops to calculate the sum of elements before and after the current index. When these sums are equal, it prints the index (1-based). The execution window shows the input size (5), the array elements (1 3 5 2 2), and the output (The first Equilibrium Point in an array is(1-based index): 3).

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class EquilibriumPointWithArg
6 {
7
8     public void equilibriumPoint(int a[])
9     {
10         System.out.println("The first Equilibrium Point in an array is(1-based index): ");
11         for(int i=0;i<a.length;i++)
12         {
13             int g=0,lsum=0,rsum=0;
14             for(int j=0;j<i;j++)
15             {
16                 lsum+=a[j];
17             }
18             for(int k=i+1;k<a.length;k++)
19             {
20                 rsum+=a[k];
21             }
22             if(rsum==lsum)
23             {
24                 System.out.println(a[i-1]+ " ");
25             }
26         }
27     }
28
29     public static void main(String[] args)
30     {
31         // TODO Auto-generated method stub
32         EquilibriumPointWithArg o = new EquilibriumPointWithArg();
33         Scanner sc = new Scanner(System.in);
34         System.out.print("Enter array size : ");
35         int size = sc.nextInt();
36         int a[] = new int[size];
37         System.out.print("Enter array element: ");
38         for(int i=0;i<a.length;i++)
39         {
40             a[i]=sc.nextInt();
41         }
42         o.equilibriumPoint(a);
43     }
44 }
```

Q4.You are given two arrays, A and B, of equal size N. The task is to find the minimum value of $A[0] * B[0] + A[1] * B[1] + \dots + A[N-1] * B[N-1]$, where shuffling of elements of arrays A and B is allowed.

Example 1:

Input: $N = 3$ $A[] = \{3, 1, 1\}$

$B[] = \{6, 5, 4\}$

Output:23

The screenshot shows a Java development environment with multiple tabs open. The active tab contains the following Java code:

```
1 package usingmethodArray;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class FindMinValueOfGivenEquationWithArg
7 {
8     public void minInEqu(int a[],int b[])
9     {
10         int g[] = new int[a.length+b.length];
11         System.out.println("");
12         int x=0,y=0,sum=0,g1=1,g2=0;
13         for(int i=0;i<a.length;i++)
14         {
15             int temp=a[i]*b[i];
16             sum+=temp;
17         }
18         System.out.println("Sum of given equation is: "+sum);
19     }
20     public static void main(String[] args)
21     {
22         FindMinValueOfGivenEquationWithArg o = new FindMinValueOfGivenEquationWithArg();
23         Scanner sc = new Scanner(System.in);
24         System.out.println("Enter array size :");
25         int size = sc.nextInt();
26         int a[] = new int[size];
27         int b[] = new int[size];
28         System.out.println("Enter 1 array element: ");
29         for(int i=0;i<a.length;i++)
30         {
31             a[i]=sc.nextInt();
32         }
33         Arrays.sort(a);
34         System.out.println("Enter 2 array element: ");
35         for(int i=0;i<b.length;i++)
36         {
37             b[i]=sc.nextInt();
38         }
39         o.minInEqu(a,b);
40     }
41 }
```

To the right of the code, the Java console window displays the program's output:

```
<terminated> FindMinValueOfGivenEquationWithArg [Java Application] C:\Users
Enter array size :
3
Enter 1 array element:
3 1 1
Enter 2 array element:
6 5 4
Sum of given equation is: 23
```

Q5. Given two sorted arrays arr1 and arr2 of size N and M respectively and an element K. The task is to find the element that would be at the k'th position of the final sorted array.

Example 1:

Input: arr1[] = {2, 3, 6, 7, 9}, arr2[] = {1, 4, 8, 10}, k = 5, **Output:**6

```

1 package usingmethodArray;
2
3 import java.util.Arrays;
4
5 import java.util.Scanner;
6
7 public class FindKthValueOfSortedArrWithArg
8 {
9     public void sortedArr(int a[],int b[])
10    {
11        int c[] = new int[a.length+b.length];
12        System.out.println("");
13        int x=0,k=5;
14        for(int i=0;i<a.length;i++)
15        {
16            c[i]=a[i];
17            x++;
18        }
19        for(int i=0;i<b.length;i++)
20        {
21            c[x]=b[i];
22            x++;
23        }
24        Arrays.sort(c);
25        for(int i=0;i<c.length;i++)
26        {
27            if((i+1)==k)
28            {
29                System.out.println(c[i]+" ");
30            }
31        }
32    }
33    public static void main(String[] args)
34    {
35        FindKthValueOfSortedArrWithArg o = new FindKthValueOfSortedArrWithArg();
36        Scanner sc = new Scanner(System.in);
37        System.out.println("Enter array size :");
38        int size = sc.nextInt();
39        int a[] = new int[size];
40        int b[] = new int[4];
41        System.out.println("Enter 1 array element: ");
42        for(int i=0;i<a.length;i++)
43        {
44            a[i]=sc.nextInt();
45        }
46        Arrays.sort(a);
47        System.out.println("Enter 2 array element: ");
48        for(int i=0;i<b.length;i++)
49        {
50            b[i]=sc.nextInt();
51        }
52        Arrays.sort(b);
53        o.sortedArr(a,b);
54    }
55}
56
57

```



```

17
18    }
19    for(int i=0;i<b.length;i++)
20    {
21        c[x]=b[i];
22        x++;
23    }
24    Arrays.sort(c);
25    for(int i=0;i<c.length;i++)
26    {
27        if((i+1)==k)
28        {
29            System.out.println(c[i]+" ");
30        }
31    }
32}
33    public static void main(String[] args)
34    {
35        FindKthValueOfSortedArrWithArg o = new FindKthValueOfSortedArrWithArg();
36        Scanner sc = new Scanner(System.in);
37        System.out.println("Enter array size :");
38        int size = sc.nextInt();
39        int a[] = new int[size];
40        int b[] = new int[4];
41        System.out.println("Enter 1 array element: ");
42        for(int i=0;i<a.length;i++)
43        {
44            a[i]=sc.nextInt();
45        }
46        Arrays.sort(a);
47        System.out.println("Enter 2 array element: ");
48        for(int i=0;i<b.length;i++)
49        {
50            b[i]=sc.nextInt();
51        }
52        Arrays.sort(b);
53        o.sortedArr(a,b);
54    }
55}
56
57

```

Q6.Given an array of size N filled with numbers from 1 to N-1 in random order. The array has only one repetitive element. The task is to find the repetitive element.

Examples:

Input: a[] = {1, 3, 2, 3, 4}

Output: 3

The screenshot shows a Java development environment with two code editors and two corresponding terminal windows (Console X).

Code Editor 1 (Top):

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class PrintKthRepeatingEleWithArg
6 {
7
8     public void FindKthEle(int a[],int k)
9     {
10         System.out.println("The first element that occurs at least K number of times is: ");
11         for(int i=0;i<a.length;i++)
12         {
13             int c=1;
14             for(int j=i+1;j<a.length;j++)
15             {
16                 if(a[i]==a[j])
17                 {
18                     c++;
19                 }
20             }
21             if(c==k)
22             {
23                 System.out.println(a[i]+" ");
24                 break;
25             }
26         }
27     }
28
29     public static void main(String[] args)
30     {
31         // TODO Auto-generated method stub
32         PrintKthRepeatingEleWithArg o = new PrintKthRepeatingEleWithArg();
33         Scanner sc = new Scanner(System.in);
34         System.out.println("Enter array size : ");
35         int size = sc.nextInt();
36         int a[] = new int[size];
37         System.out.println("Enter array element: ");
38         for(int i=0;i<a.length;i++)
39         {
40             a[i]=sc.nextInt();
41         }
42         System.out.println("Enter K value: ");
43         int k = sc.nextInt();
44     }
45 }
```

Code Editor 2 (Bottom):

```
1
2
3     public void FindKthEle(int a[],int k)
4     {
5         System.out.println("The first element that occurs at least K number of times is: ");
6         for(int i=0;i<a.length;i++)
7         {
8             int c=1;
9             for(int j=i+1;j<a.length;j++)
10            {
11                if(a[i]==a[j])
12                {
13                    c++;
14                }
15            }
16            if(c==k)
17            {
18                System.out.println(a[i]+" ");
19                break;
20            }
21        }
22    }
23
24    public static void main(String[] args)
25    {
26        // TODO Auto-generated method stub
27        PrintKthRepeatingEleWithArg o = new PrintKthRepeatingEleWithArg();
28        Scanner sc = new Scanner(System.in);
29        System.out.println("Enter array size : ");
30        int size = sc.nextInt();
31        int a[] = new int[size];
32        System.out.println("Enter array element: ");
33        for(int i=0;i<a.length;i++)
34        {
35            a[i]=sc.nextInt();
36        }
37        System.out.println("Enter K value: ");
38        int k = sc.nextInt();
39        o.FindKthEle(a,k);
40    }
41 }
```

Console Output 1 (Top):

```
<terminated> PrintKthRepeatingEleWithArg [Java Application] C:\Users\Shree\p2\pool\pl
Enter array size :
5
Enter array element:
1 3 2 3 4
Enter K value:
2
The first element that occurs at least K number of times is:
3
```

Console Output 2 (Bottom):

```
<terminated> PrintKthRepeatingEleWithArg [Java Application] C:\Users\Shree\p2\pool\pl
Enter array size :
5
Enter array element:
1 3 2 3 4
Enter K value:
2
The first element that occurs at least K number of times is:
3
```

Q7. Given two unsorted arrays that represent two sets (elements in every array are distinct), find the union and intersection of two arrays.

Example: arr1[] = {7, 1, 5, 2, 3, 6} ,arr2[] = {3, 8, 6, 20, 7}

Then your program should print Union as {1, 2, 3, 5, 6, 7, 8, 20} and Intersection as {3, 6, 7}. Note that the elements of union and intersection can be printed in any order.

The image shows two screenshots of a Java IDE interface. Both screenshots display a code editor and a terminal window (Console).

Top Screenshot:

- Code:** A Java class named `UnionAndIntersectionInArrWithArg` containing a method `unionIntersection`. This method concatenates two arrays into a temporary array `c`, sorts it, and then iterates through it to print common elements (intersection) and unique elements (union). It also prints the arrays themselves.
- Console Output:**

```
<terminated> UnionAndIntersectionInArrWithArg [Java Application] C:\Users\Shree\p2\p1
Enter array size :
6
Enter 1 array element:
7 1 5 2 3 6
Enter 2 array element:
3 8 6 20 7

Intersection array element is: 3 6 7
Union of array element is: 1 2 3 5 6 7 8 20
```

Bottom Screenshot:

- Code:** The same Java class `UnionAndIntersectionInArrWithArg`, but with a different implementation. It uses a temporary array `c` to mark elements as found in the first array. It then iterates through the second array to print elements that were marked as found in the first array (intersection) and all remaining elements of the second array (union).
- Console Output:**

```
<terminated> UnionAndIntersectionInArrWithArg [Java Application] C:\Users\Shree\p2\p1
Enter array size :
6
Enter 1 array element:
7 1 5 2 3 6
Enter 2 array element:
3 8 6 20 7

Intersection array element is: 3 6 7
Union of array element is: 1 2 3 5 6 7 8 20
```

Q8.Given three arrays sorted in non-decreasing order, print all common elements in these arrays.

Examples: Input: ar1[] = {1, 5, 10, 20, 40, 80} ,ar2[] = {6, 7, 20, 80, 100} ,ar3[] = {3, 4, 15, 20, 30, 70, 80, 120} ,**Output:** 20, 80

The screenshot shows two code snippets in a Java IDE. Both snippets implement the same logic to find common elements in three sorted arrays (ar1, ar2, ar3) and print them. The first snippet uses nested loops to iterate through all elements of ar1 and ar2, comparing each element with every element in ar3. The second snippet optimizes this by using a single loop for ar1 and ar2, and a binary search for ar3. Both snippets include imports for java.util.Arrays and java.util.Scanner, and use System.out.println statements to interact with the user and output results. The console output for both runs shows the same input and output: entering array sizes and elements, and then printing the common elements 20 and 80.

```
1 package usingmethodArray;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class PrintCommonArrEleWithArg
7 {
8
9     public void printCoomonEle(int a[],int b[],int c[])
10    {
11        Arrays.sort(c);
12        System.out.print("\nCommon array element is: ");
13        for(int i=0;i<c.length;i++)
14        {
15            for(int j=0;j<a.length;j++)
16            {
17                for(int k=0;k<b.length;k++)
18                {
19                    if(c[i]==a[j] && c[i]==b[k])
20                    {
21                        System.out.print(c[i]+" ");
22                        break;
23                    }
24                }
25            }
26        }
27    }
28
29    public static void main(String[] args)
30    {
31        PrintCommonArrEleWithArg o = new PrintCommonArrEleWithArg();
32        Scanner sc = new Scanner(System.in);
33        System.out.println("Enter 3 array size: ");
34        int size = sc.nextInt();
35        int size2 = sc.nextInt();
36        int a[] = new int[size];
37        int b[] = new int[size];
38        int c[] = new int[size2];
39        System.out.println("Enter 1 array element: ");
40        for(int i=0;i<a.length;i++)
41        {
42            a[i]=sc.nextInt();
43        }
44        System.out.println("Enter 2 array element: ");
45        for(int i=0;i<b.length;i++)
46        {
47            b[i]=sc.nextInt();
48        }
49        System.out.println("Enter 3 array element: ");
50        for(int i=0;i<c.length;i++)
51        {
52            c[i]=sc.nextInt();
53        }
54        o.printCoomonEle(a,b,c);
55    }
56 }
```

```
1 package usingmethodArray;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class PrintCommonArrEleWithArg
7 {
8
9     public void printCoomonEle(int a[],int b[],int c[])
10    {
11        if(c[1]==a[0] && c[1]==b[0])
12        {
13            System.out.print(c[1]+" ");
14            break;
15        }
16    }
17
18    public static void main(String[] args)
19    {
20        PrintCommonArrEleWithArg o = new PrintCommonArrEleWithArg();
21        Scanner sc = new Scanner(System.in);
22        System.out.println("Enter 3 array size: ");
23        int size = sc.nextInt();
24        int size2 = sc.nextInt();
25        int size1 = sc.nextInt();
26        int a[] = new int[size];
27        int b[] = new int[size1];
28        int c[] = new int[size2];
29        System.out.println("Enter 1 array element: ");
30        for(int i=0;i<a.length;i++)
31        {
32            a[i]=sc.nextInt();
33        }
34        System.out.println("Enter 2 array element: ");
35        for(int i=0;i<b.length;i++)
36        {
37            b[i]=sc.nextInt();
38        }
39        System.out.println("Enter 3 array element: ");
40        for(int i=0;i<c.length;i++)
41        {
42            c[i]=sc.nextInt();
43        }
44        o.printCoomonEle(a,b,c);
45    }
46 }
```

Q10. Given an array arr[] of integers, segregate even and odd numbers in the array. Such that all the even numbers should be present first, and then the odd numbers.

Examples:

Input: arr[] = 1 9 5 3 2 6 7 11

Output: 2 6 5 3 1 9 7 11

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 class EvenOdd
6 {
7     public int[] segregateEvenOdd(int a[])
8     {
9         for(int i=0;i<a.length;i++)
10        {
11            for(int j=i+1;j<a.length;j++)
12            {
13                if(a[j]%2==0)
14                {
15                    int t=a[i];
16                    a[i]=a[j];
17                    a[j]=t;
18                }
19            }
20        }
21        int te=a[0];
22        int ti=a[1];
23        a[0]=ti;
24        a[1]=te;
25        return a;
26    }
27 }
28 public class SegregateEvenAndOddWithReturnTypeAndArg
29 {
30
31     public static void main(String[] args)
32     {
33         // TODO Auto-generated method stub
34         Scanner sc = new Scanner(System.in);
35         System.out.println("Enter array size: ");
36         int size = sc.nextInt();
37         int a[]=new int[size];
38
39         for(int i=0;i<size;i++)
40         {
41             a[i]=sc.nextInt();
42         }
43         EvenOdd e= new EvenOdd();
44         int c[]=e.segregateEvenOdd(a);
45         System.out.println("The even numbers present first and then the odd numbers:");
46         for(int i=0;i<a.length;i++)
47         {
48             System.out.println(c[i]+" ");
49         }
50     }
51
52 }
```

```
1 Enter array size:
2 8
3 Enter array element:
4 1 9 5 3 2 6 7 11
5 The even numbers present first and then the odd numbers:
6 2
7 6
8 5
9 3
10 1
11 9
12 7
13 11
```

```
1 Enter array size:
2 8
3 Enter array element:
4 1 9 5 3 2 6 7 11
5 The even numbers present first and then the odd numbers:
6 2
7 6
8 5
9 3
10 1
11 9
12 7
13 11
```

Q11. Given a sorted array of n distinct integers where each integer is in the range from 0 to m-1 and m > n. Find the smallest number that is missing from the array.

Examples:

Input: {0, 1, 2, 6, 9}, n = 5, m = 10

Output: 3

The screenshot shows an IDE interface with two tabs: 'PrimeReturnType.java' and 'FindMissingEleWithReturnTypeAndArg.java'. The 'FindMissingEleWithReturnTypeAndArg.java' tab is active, displaying the following code:

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 class MissingEle
6 {
7     public int findMissingEle(int a[], int size)
8     {
9         int sum=0;
10        System.out.println("Given array element is:");
11        for(int i=0;i<a.length;i++)
12        {
13            System.out.println(a[i]+" ");
14            sum+=a[i];
15        }
16        int mis = size*(size+1)/2;
17        return sum-mis;
18    }
19 }
20 public class FindMissingEleWithReturnTypeAndArg
21 {
22     public static void main(String[] args)
23     {
24         Scanner sc = new Scanner(System.in);
25         System.out.println("Enter array size: ");
26         int size = sc.nextInt();
27         int a[] = new int[size];
28         System.out.println("Enter array element: ");
29         for(int i=0;i<a.length;i++)
30         {
31             a[i] = sc.nextInt();
32         }
33         MissingEle ee = new MissingEle();
34         int cee = ee.findMissingEle(a, size);
35         System.out.println("The given missing array element is: "+ce
36     }
37 }
```

To the right of the code, the 'Console' tab shows the output of running the program:

```
<terminated> FindMissingEleWithReturnTypeAndArg [Java Application] C:\Users\Shree\p2\pool\plu
Enter array size:
5
Enter array element:
0 1 2 6 9
Given array element is:
0
1
2
6
9
The given missing array element is: 3
```

Q12. Given a sorted array with possibly duplicate elements. The task is to find indexes of first and last occurrences of an element X in the given array.

Note: If the element is not present in the array return {-1,-1} as pair.

Example 1:

Input: N = 9

v[] = {1, 3, 5, 5, 5, 5, 67, 123, 125}

X = 5

Output: 2 5

Explanation:

Index of first occurrence of 5 is 2

and index of last occurrence of 5 is 5.

```

1 package usingmethodFromArray;
2
3 import java.util.Scanner;
4
5 public class RepeatingEleIndWithReturnType
6 {
7
8     public int[] repatDulArrEle()
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter array size: ");
12         int size = sc.nextInt();
13         int a[] = new int[size];
14         System.out.println("Enter array element: ");
15         for (int i = 0; i < a.length; i++)
16         {
17             a[i] = sc.nextInt();
18         }
19         System.out.println("Enter repeating element: ");
20         int k = sc.nextInt();
21         int firstIndex = -1;
22         int lastIndex = -1;
23         for (int i = 0; i < a.length; i++)
24         {
25             if (a[i] == k)
26             {
27                 if (firstIndex == -1)
28                 {
29                     firstIndex = i;
30                 }
31                 lastIndex = i;
32             }
33         }
34         if (firstIndex == -1)
35         {
36             System.out.println("Element not found in the array.");
37         }
38     }
39
40     public int[] EnterRepeatingEleInd()
41     {
42         int k = sc.nextInt();
43         int firstIndex = -1;
44         int lastIndex = -1;
45         for (int i = 0; i < a.length; i++)
46         {
47             if (a[i] == k)
48             {
49                 if (firstIndex == -1)
50                 {
51                     firstIndex = i;
52                 }
53                 lastIndex = i;
54             }
55         }
56         if (firstIndex == -1)
57         {
58             System.out.println("Element not found in the array.");
59         }
59         else
60         {
61             System.out.println("The first occurrence of the element is at index: " + firstIndex);
62             System.out.println("The last occurrence of the element is at index: " + lastIndex);
63         }
64
65         return new int[] {firstIndex, lastIndex};
66     }
67
68     public static void main(String[] args)
69     {
70         RepeatingEleIndWithReturnType m = new RepeatingEleIndWithReturnType();
71         int[] result = m.repatDulArrEle();
72         System.out.println("First Index: " + result[0]);
73         System.out.println("Last Index: " + result[1]);
74     }
75 }

```

Console output:

```

<terminated> RepeatingEleIndWithReturnType [Java Application] C:\Users\Shree\p2\pool\plugins\RepeatingEleIndWithReturnType.jar
Enter array size:
9
Enter array element:
1 3 5 5 5 67 123 125
Enter repeating element:
5
The first occurrence of the element is at index: 2
The last occurrence of the element is at index: 5
First Index: 2
Last Index: 5

```

Console output (second run):

```

<terminated> RepeatingEleIndWithReturnType [Java Application] C:\Users\Shree\p2\pool\plugins\RepeatingEleIndWithReturnType.jar
Enter array size:
9
Enter array element:
1 3 5 5 5 67 123 125
Enter repeating element:
5
The first occurrence of the element is at index: 2
The last occurrence of the element is at index: 5
First Index: 2
Last Index: 5

```

1. Write a Java program to find maximum product of two integers in a given array of integers.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class FindMaxProductArrEleWithReturnTypeAndArg
6 {
7     public int maxProduct(int a[])
8     {
9
10         int max =0,smax =0;
11         for(int i=0;i<a.length;i++)
12         {
13             if(a[i]>max)
14             {
15                 smax=max;
16                 max=a[i];
17             }
18             else if(a[i]>smax && smax!=max)
19             {
20                 smax=a[i];
21             }
22         }
23         int product=max*smax;
24         return product;
25     }
26
27     public static void main(String[] args)
28     {
29         Scanner sc = new Scanner(System.in);
30         System.out.println("Enter array size: ");
31         int size = sc.nextInt();
32         int a[]=new int[size];
33         System.out.println("Enter array element: ");
34         for(int i=0;i<a.length;i++)
35         {
36             a[i]=sc.nextInt();
37         }
38     }
39 }

```



```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class RepeatingEleIndWithReturnType
6 {
7     public int maxProduct(int a[])
8     {
9
10         int max =0,smax =0;
11         for(int i=0;i<a.length;i++)
12         {
13             if(a[i]>max)
14             {
15                 smax=max;
16                 max=a[i];
17             }
18             else if(a[i]>smax && smax!=max)
19             {
20                 smax=a[i];
21             }
22         }
23         int product=max*smax;
24         return product;
25     }
26
27     public static void main(String[] args)
28     {
29         Scanner sc = new Scanner(System.in);
30         System.out.println("Enter array size: ");
31         int size = sc.nextInt();
32         int a[]=new int[size];
33         System.out.println("Enter array element: ");
34         for(int i=0;i<a.length;i++)
35         {
36             a[i]=sc.nextInt();
37         }
38         FindMaxProductArrEleWithReturnTypeAndArg f = new FindMaxProductArrEleWithReturnTypeAndArg();
39         int c=f.maxProduct(a);
40         System.out.println("Maximum product of two array element: "+c);
41     }
42 }
43 
```

2. Write a Java program to shuffle a given array of integers.

Example:

Input: nums = { 1, 2, 3, 4, 5, 6 }

Output: Shuffle Array: [4, 2, 6, 5, 1, 3]

The screenshot shows an IDE interface with two tabs open: "RandomShuffleArrEleWithReturnTypeAndArg.java" and "Console".

Code (RandomShuffleArrEleWithReturnTypeAndArg.java):

```
1 package usingmethodArray;
2 import java.util.Arrays;
3 import java.util.Random;
4 import java.util.Scanner;
5
6 public class RandomShuffleArrEleWithReturnTypeAndArg
7 {
8     public int[] randomShuffle(int a[])
9     {
10         Random r = new Random();
11         for(int i=0;i<a.length;i++)
12         {
13             int ind = r.nextInt(a.length);
14             int t=a[i];
15             a[i]=a[ind];
16             a[ind]=t;
17         }
18         return a;
19     }
20     public static void main(String[] args)
21     {
22         Scanner sc =new Scanner(System.in);
23         Random g = new Random();
24         System.out.println("Enter array size: ");
25         int size = sc.nextInt();
26         int a[] =new int [size];
27         System.out.println("Enter array element: ");
28         for(int i=0;i<a.length;i++)
29         {
30             a[i]=sc.nextInt();
31         }
32         RandomShuffleArrEleWithReturnTypeAndArg r1 = new RandomShuffleArrEleWithRet
33         int b[] =r1.randomShuffle(a);
34         System.out.println("Shuffle Array element is: ");
35         System.out.println(Arrays.toString(b));
36     }
37 }
```

Console Output:

```
terminated> RandomShuffleArrEleWithReturnTypeAndArg [Java Application]
Enter array size:
6
Enter array element:
1 2 3 4 5 6
Shuffle Array element is:
[4, 2, 1, 5, 3, 6]
```

3. Write a Java program to rearrange a given array of unique elements such that every second element of the array is greater than its left and right elements.

Example:

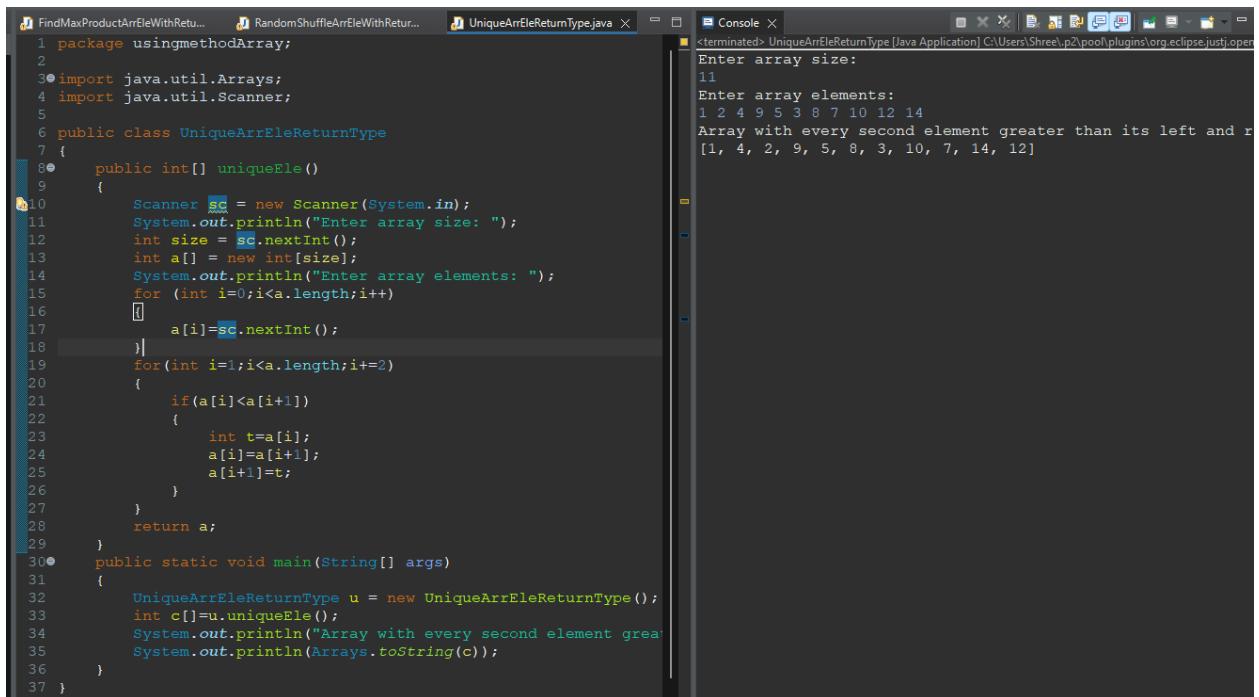
Input :

nums= { 1, 2, 4, 9, 5, 3, 8, 7, 10, 12, 14 }

Output:

Array with every second element is greater than its left and right elements:

[1, 4, 2, 9, 3, 8, 5, 10, 7, 14, 12]



The screenshot shows the Eclipse IDE interface with two tabs open: 'UniqueArrEleReturnType.java' and 'Console'. The code in 'UniqueArrEleReturnType.java' defines a class 'UniqueArrEleReturnType' with a static method 'uniqueEle()' that takes an array of integers and returns it rearranged according to the problem statement. The 'main()' method creates an instance of the class and prints the resulting array. The 'Console' tab shows the user inputting an array size of 11 and elements [1, 2, 4, 9, 5, 3, 8, 7, 10, 12, 14], followed by the output message 'Array with every second element greater than its left and right elements' and the rearranged array [1, 4, 2, 9, 3, 8, 5, 10, 7, 14, 12].

```
1 package usingmethodArray;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class UniqueArrEleReturnType
7 {
8     public int[] uniqueEle()
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter array size: ");
12         int size = sc.nextInt();
13         int a[] = new int[size];
14         System.out.println("Enter array elements: ");
15         for (int i=0;i<a.length;i++)
16         {
17             a[i]=sc.nextInt();
18         }
19         for(int i=1;i<a.length;i+=2)
20         {
21             if(a[i]<a[i+1])
22             {
23                 int t=a[i];
24                 a[i]=a[i+1];
25                 a[i+1]=t;
26             }
27         }
28         return a;
29     }
30     public static void main(String[] args)
31     {
32         UniqueArrEleReturnType u = new UniqueArrEleReturnType();
33         int c[] = u.uniqueEle();
34         System.out.println("Array with every second element greater than its left and right elements: ");
35         System.out.println(Arrays.toString(c));
36     }
37 }
```

```
Console X
<terminated> UniqueArrEleReturnType [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.jdt.core\open
Enter array size:
11
Enter array elements:
1 2 4 9 5 3 8 7 10 12 14
Array with every second element greater than its left and right elements:
[1, 4, 2, 9, 3, 8, 5, 10, 7, 14, 12]
```

4. Write a Java program to replace each element of the array with product of every other element in a given array of integers.

Example:

Input : nums1 = { 1, 2, 3, 4, 5, 6, 7}, nums2 = {0, 1, 2, 3, 4, 5, 6, 7}

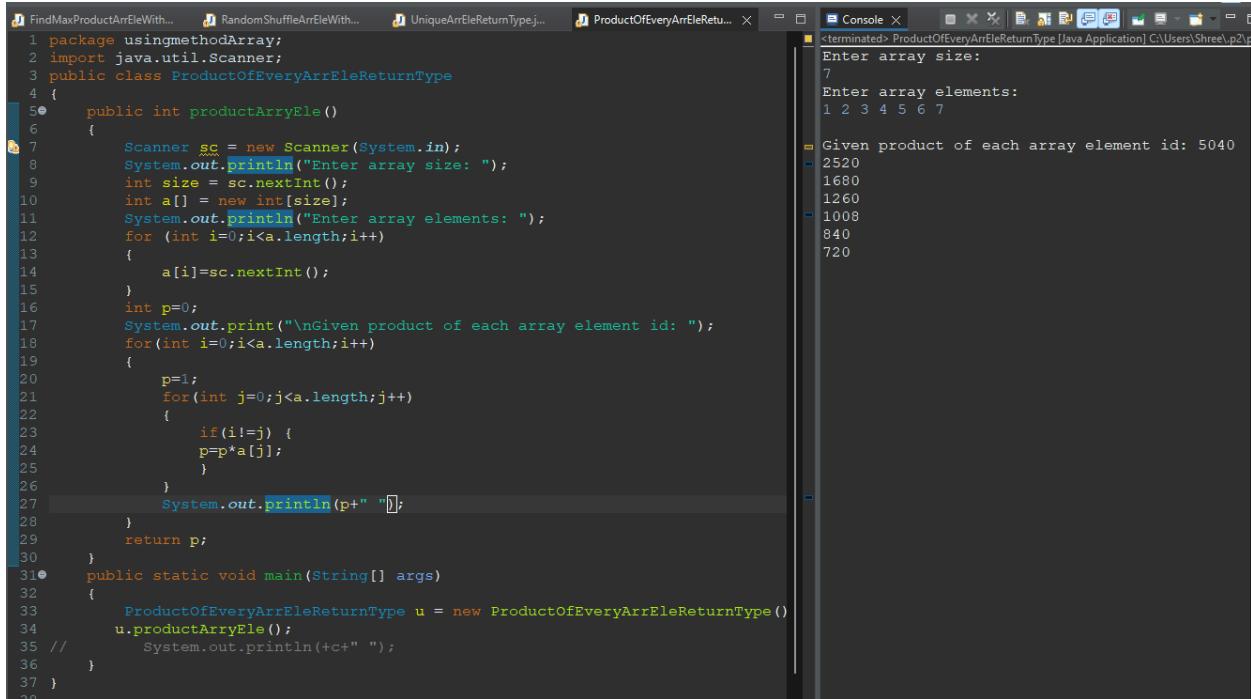
Output:

Array with product of every other element:

[5040, 2520, 1680, 1260, 1008, 840, 720]

Array with product of every other element:

[5040, 0, 0, 0, 0, 0, 0]



The screenshot shows an IDE interface with two tabs open: "ProductOfEveryArrEleReturnType" and "Console".

Code (ProductOfEveryArrEleReturnType.java):

```
1 package usingmethodArray;
2 import java.util.Scanner;
3 public class ProductOfEveryArrEleReturnType
4 {
5     public int productArrEle()
6     {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Enter array size: ");
9         int size = sc.nextInt();
10        int a[] = new int[size];
11        System.out.println("Enter array elements: ");
12        for (int i=0;i<a.length;i++)
13        {
14            a[i]=sc.nextInt();
15        }
16        int p=1;
17        System.out.print("\nGiven product of each array element id: ");
18        for(int i=0;i<a.length;i++)
19        {
20            p=1;
21            for(int j=0;j<a.length;j++)
22            {
23                if(i!=j)
24                p=p*a[j];
25            }
26        }
27        System.out.println(p+"[]");
28    }
29    return p;
30 }
31 public static void main(String[] args)
32 {
33     ProductOfEveryArrEleReturnType u = new ProductOfEveryArrEleReturnType();
34     u.productArrEle();
35 //     System.out.println(+c+" ");
36 }
37 }
```

Console Output:

```
Enter array size:
7
Enter array elements:
1 2 3 4 5 6 7

Given product of each array element id: 5040
2520
1680
1260
1008
840
720
```

5. Write a Java program to find maximum difference between two elements in a given array of integers such that smaller element appears before larger element.

Example:

Input :nums = { 2, 3, 1, 7, 9, 5, 11, 3, 5 }

Output: The maximum difference between two elements of the said array elements:10

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class MaxDiffrenceArrEleReturnType
6 {
7     public int maxDiffrence()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        System.out.println("Given array element is: ");
19        for(int i=0;i<a.length;i++)
20        {
21            System.out.println(a[i]+" ");
22        }
23        int max=0,min=Integer.MAX_VALUE;
24        for(int i=0;i<a.length;i++)
25        {
26            if(a[i]>max)
27            {
28                max=a[i];
29            }
30            else if(a[i]<min)
31            {
32                min=a[i];
33            }
34        }
35        int diff=max-min;
36        return diff;
37    }
38}
39
40 public static void main(String[] args)
41 {
42     MaxDiffrenceArrEleReturnType a1 = new MaxDiffrenceArrEleReturnType();
43     int d=a1.maxDiffrence();
44     System.out.println("Given diffrence between two integer number is: "+d);
45 }
```

```
<terminated> MaxDiffrenceArrEleReturnType [Java Application] C:\Users\Shree\p2\pool\plugins\MaxDiffrenceArrEleReturnType.jar
Enter array size :
9
Enter array element:
2 3 1 7 9 5 11 3 5
Given array element is:
2
3
1
7
9
5
11
3
5
Given diffrence between two integer number is: 10

<terminated> MaxDiffrenceArrEleReturnType [Java Application] C:\Users\Shree\p2\pool\plugins\MaxDiffrenceArrEleReturnType.jar
Enter array size :
9
Enter array element:
2 3 1 7 9 5 11 3 5
Given array element is:
2
3
1
7
9
5
11
3
5
Given diffrence between two integer number is: 10
```

6. Find a peak element which is not smaller than its neighbors

Input: array[] = {5, 10, 20, 15}

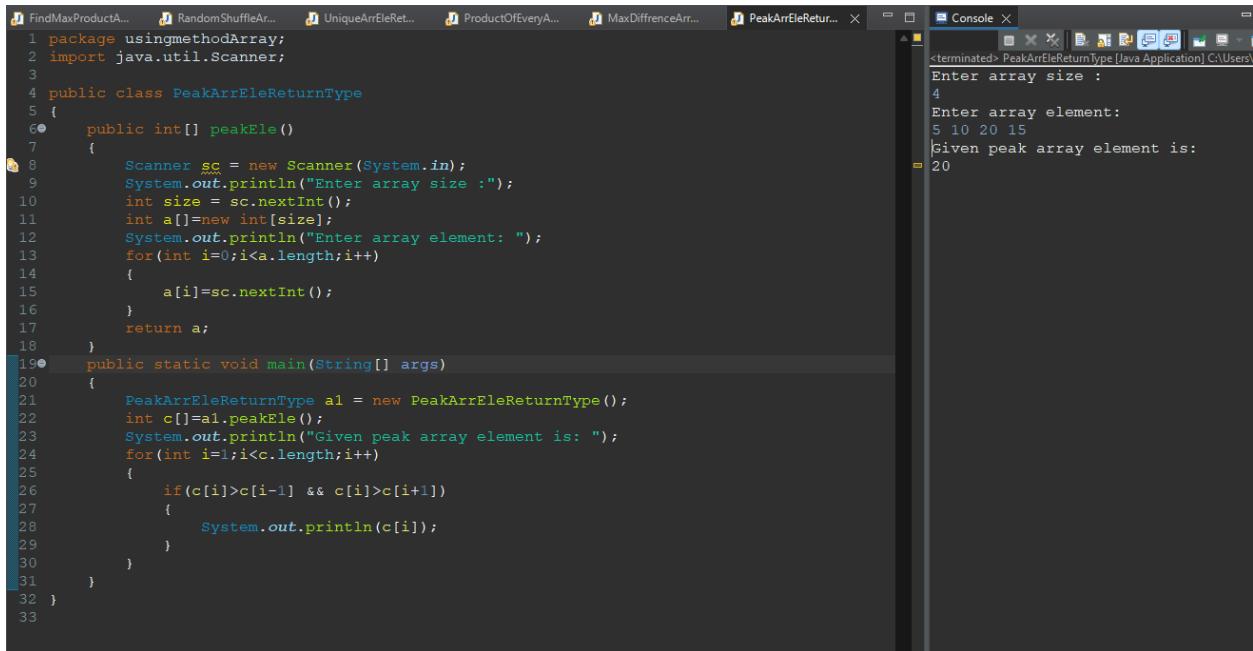
Output: 20

Explanation: The element 20 has neighbors 10 and 15, both of them are less than 20.

Input: array[] = {10, 20, 15, 2, 23, 90, 67}

Output: 20 or 90

Explanation: The element 20 has neighbors 10 and 15, both of them are less than 20, similarly 90 has neighbors 23 and 67.



The screenshot shows a Java development environment with multiple tabs open. The active tab contains the following Java code:

```
1 package usingmethodArray;
2 import java.util.Scanner;
3
4 public class PeakArrEleReturnType
5 {
6     public int[] peakEle()
7     {
8         Scanner sc = new Scanner(System.in);
9         System.out.println("Enter array size :");
10        int size = sc.nextInt();
11        int a[] = new int[size];
12        System.out.println("Enter array element: ");
13        for(int i=0;i<a.length;i++)
14        {
15            a[i] = sc.nextInt();
16        }
17        return a;
18    }
19    public static void main(String[] args)
20    {
21        PeakArrEleReturnType a1 = new PeakArrEleReturnType();
22        int c[] = a1.peakEle();
23        System.out.println("Given peak array element is: ");
24        for(int i=1;i<c.length;i++)
25        {
26            if(c[i]>c[i-1] && c[i]>c[i+1])
27            {
28                System.out.println(c[i]);
29            }
30        }
31    }
32 }
33
```

The console window shows the execution of the program:

```
<terminated> PeakArrEleReturnType [Java Application] C:\Users\...
Enter array size :
4
Enter array element:
5 10 20 15
Given peak array element is:
20
```

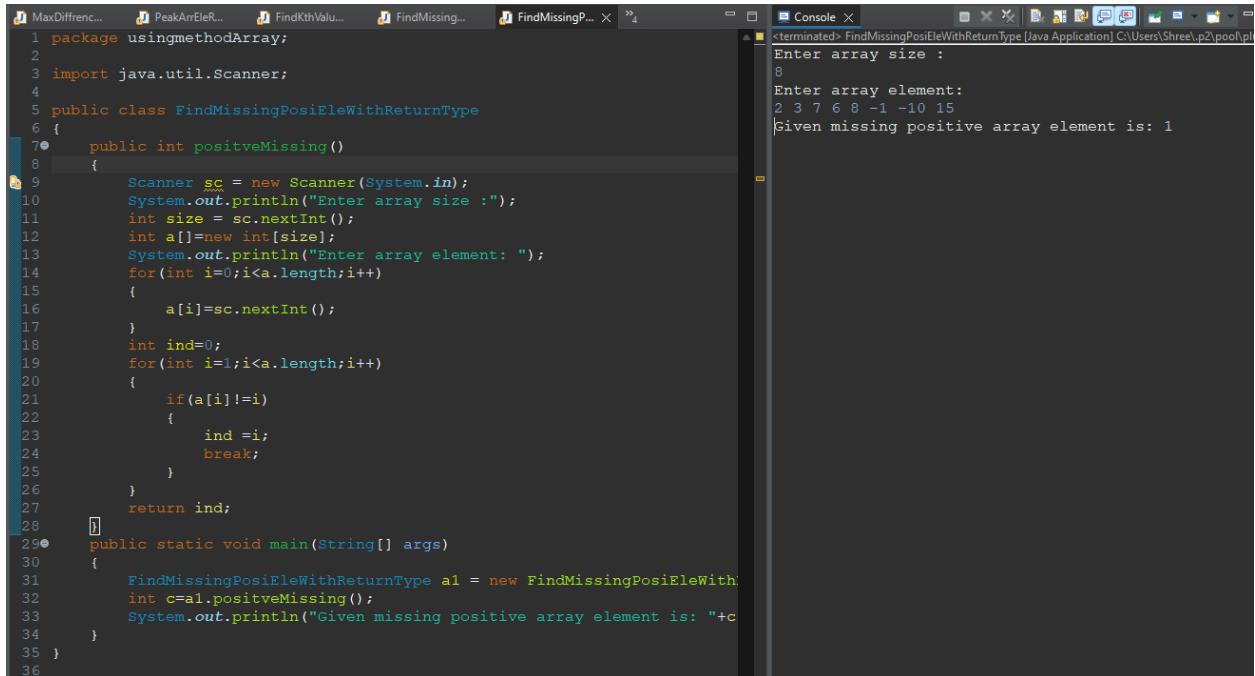
Q11. Given an unsorted array arr[] with both positive and negative elements, the task is to find the smallest positive number missing from the array.

Note: You can modify the original array.

Examples:

Input: arr[] = {2, 3, 7, 6, 8, -1, -10, 15}

Output: 1



The screenshot shows a Java development environment with two panes. The left pane displays the source code of a Java class named `FindMissingPosiEleWithReturnType`. The right pane shows the console window with the following interaction:

```
Enter array size : 8
Enter array element:
2 3 7 6 8 -1 -10 15
Given missing positive array element is: 1
```

The code implements a solution to find the first missing positive integer in an array. It uses a scanner to read the array size and elements. It then iterates through the array to find the first index where the value does not match its index. This index is returned as the result.

```
1 package usingmethodArray;
2
3 import java.util.Scanner;
4
5 public class FindMissingPosiEleWithReturnType
6 {
7     public int positiveMissing()
8     {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter array size :");
11        int size = sc.nextInt();
12        int a[] = new int[size];
13        System.out.println("Enter array element: ");
14        for(int i=0;i<a.length;i++)
15        {
16            a[i]=sc.nextInt();
17        }
18        int ind=0;
19        for(int i=1;i<a.length;i++)
20        {
21            if(a[i]!=i)
22            {
23                ind =i;
24                break;
25            }
26        }
27        return ind;
28    }
29    public static void main(String[] args)
30    {
31        FindMissingPosiEleWithReturnType a1 = new FindMissingPosiEleWithReturnType();
32        int c=a1.positiveMissing();
33        System.out.println("Given missing positive array element is: "+c);
34    }
35}
36
```

Q12. Find the majority element in the array. A majority element in an array $A[]$ of size n is an element that appears more than $n/2$ times (and hence there is at most one such element).

Examples :

Input : {3, 3, 4, 2, 4, 4, 2, 4, 4}

Output : 4

The screenshot shows a Java development environment with two panes. The left pane displays the source code for a class named `FindMajorityEleWithReturnType`. The right pane shows the console output of the application's execution.

```
1 package usingmethodArray;
2 import java.util.Arrays;
3 import java.util.Scanner;
4 public class FindMajorityEleWithReturnType
5 {
6     public int[] majorityArryEle()
7     {
8         Scanner sc = new Scanner(System.in);
9         System.out.println("Enter array size :");
10        int size = sc.nextInt();
11        int a[] = new int[size];
12        System.out.println("Enter array element: ");
13        for(int i=0;i<a.length;i++)
14        {
15            a[i]=sc.nextInt();
16        }
17        int c=0;
18        for(int i=1;i<a.length;i++)
19        {
20            for(int j=i+1;j<a.length;j++)
21            {
22                if(a[i]==a[j])
23                {
24                    c++;
25                }
26            }
27            if(c==size/2)
28            {
29                System.out.println(a[i]+" ");
30            }
31        }
32        return a;
33    }
34    public static void main(String[] args)
35    {
36        FindMajorityEleWithReturnType a1 = new FindMajorityEleWithReturnType();
37        int c[] = a1.majorityArryEle();
38        System.out.println(Arrays.toString(c));
39    }
40 }
```

Console Output:

```
<terminated> FindMajorityEleWithReturnType [Java Application] C:\Users\Shre
Enter array size :
9
Enter array element:
3 3 4 2 4 4 2 4 4
4
[3, 3, 4, 2, 4, 4, 2, 4, 4]
```

Q13. Given two sorted arrays A and B of size p and q, write a Java program to merge elements of A with B by maintaining the sorted order i.e.

fill A with first p smallest elements and fill B with remaining elements.

Example:Input :int[] A = { 1, 5, 6, 7, 8, 10 }int[] B = { 2, 4, 9 } Output:Sorted Arrays:A: [1, 2, 4, 5, 6, 7] B: [8, 9, 10]

```
1 package usingmethodArray;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class ChangeSortedArrWithReturn
7 {
8     public int[] sorteArr()
9     {
10         Scanner sc = new Scanner(System.in);
11         System.out.println("Enter array size :");
12         int size = sc.nextInt();
13         int size1 = sc.nextInt();
14         int a[] = new int[size];
15         int b[] = new int[size];
16         System.out.println("Enter 1 array element: ");
17         for(int i=0;i<a.length;i++)
18         {
19             a[i]=sc.nextInt();
20         }
21         System.out.println("Enter 2 array element: ");
22         for(int i=0;i<b.length;i++)
23         {
24             b[i]=sc.nextInt();
25         }
26         int c=0;
27         for(int i=0;i<a.length;i++)
28         {
29             for(int j=0;j<b.length;j++)
30             {
31                 if(i==4)
32                 {
33                     int t=b[0];
34                     b[0]=a[i];
35                     a[i]=t;
36                 }
37                 else if(i==5)
38                 {
39                     int t=b[1];
40                     b[1]=a[i];
41                     a[i]=t;
42                 }
43             }
44         }
45         System.out.println("Enter 1 array element is: ");
46         Arrays.sort(a);
47         System.out.println(Arrays.toString(a));
48         System.out.println("Enter 2 array element is: ");
49         Arrays.sort(b);
50         System.out.println( Arrays.toString(b));
51         return a;
52     }
53     public static void main(String[] args)
54     {
55         ChangeSortedArrWithReturn a1 = new ChangeSortedArrWithReturn();
56         int c[] = a1.sorteArr();
57         //System.out.println(Arrays.toString(c));
58     }
59 }
60
```

Console Output:

```
<terminated> ChangeSortedArrWithReturn [Java Application] C:\Users\Shree\p2\pool
Enter array size :
6 3
Enter 1 array element:
1 5 6 7 8 10
Enter 2 array element:
2 4 9
Enter 1 array element is:
[1, 2, 4, 5, 6, 7]
Enter 2 array element is:
[8, 9, 10]
```

Console Output (Second Run):

```
<terminated> ChangeSortedArrWithReturn [Java Application] C:\Users\Shree\p2\pool\plugins
Enter array size :
6 3
Enter 1 array element:
1 5 6 7 8 10
Enter 2 array element:
2 4 9
Enter 1 array element is:
[1, 2, 4, 5, 6, 7]
Enter 2 array element is:
[8, 9, 10]
```

