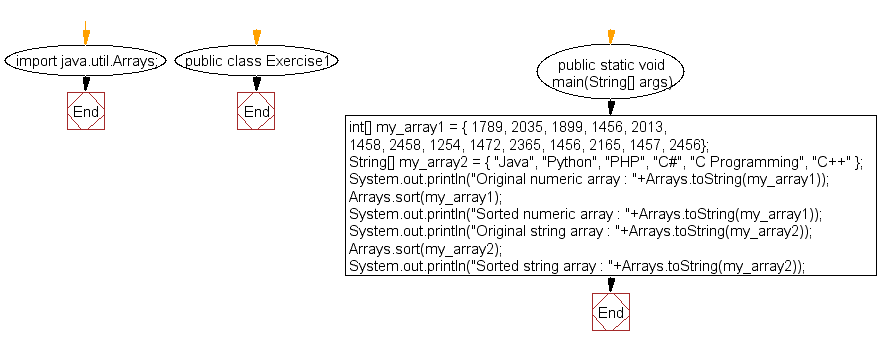
**Java Array Programs**

1) **Write a Java program to sort a numeric array and a string array**

Flowchart:



import java.util.Arrays;

public class Program\_1

{

public static void main(String[] args)

{

// Numeric Array

int[] arr = { 10,15,2,7,4,3,23,89,90,15,14,18,17,100 };

// String array

String[] str = { "apples", "mangoes", "bananas", "strawberries", "guavas" };

// to print original array

System.out.println("Original numeric Array : " + Arrays.toString(arr));

//sort the array

Arrays.sort(arr);

// print the sorted array

System.out.println("Sorted numeric Array : " + Arrays.toString(arr));

// to print original string array

System.out.println("Original String Array : " + Arrays.toString(str));

// sort original array

Arrays.sort(str);

//print the sorted array

System.out.println("Sorted String Array :" + Arrays.toString(str));

}

}

Output:

Original numeric Array : [10, 15, 2, 7, 4, 3, 23, 89, 90, 15, 14, 18, 17, 100]

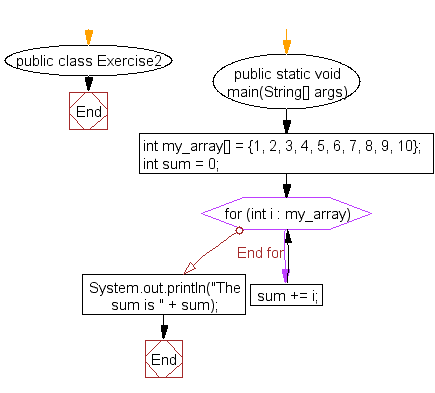
Sorted numeric Array : [2, 3, 4, 7, 10, 14, 15, 15, 17, 18, 23, 89, 90, 100]

Original String Array : [apples, mangoes, bananas, strawberries, guavas]

Sorted String Array :[apples, bananas, guavas, mangoes, strawberries]

**2) Write a Java program to sum values of an array.**

Flowchart:



public class Program\_2

{

public static void main(String[] args)

{

int[] arr = {1,2,3,4,5,6,7};

int sum=0;

for (int i : arr)

sum += i;

System.out.println("The Sum of the array is :" + sum);

}

}

Output:

The Sum of the array is :28

**3) Write a Java program to print the following grid.**

**Expected Output :**

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

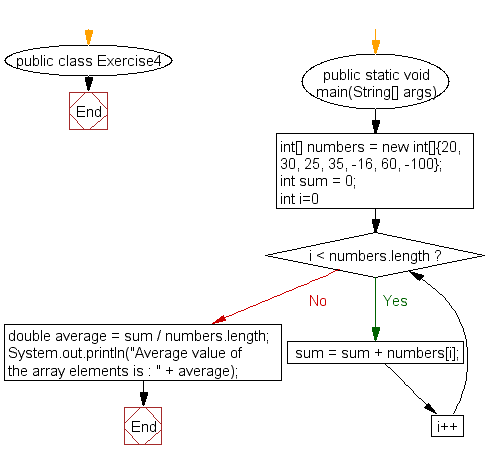
- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

Flowchart:



public class Program\_3

{

public static void main(String[] args)

{

int[][] arr = new int[10][10];

for ( int i=0 ; i < arr.length ;i++ )

{

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

{

System.out.print("- ");

}

System.out.println();

}

}

}

Output:

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

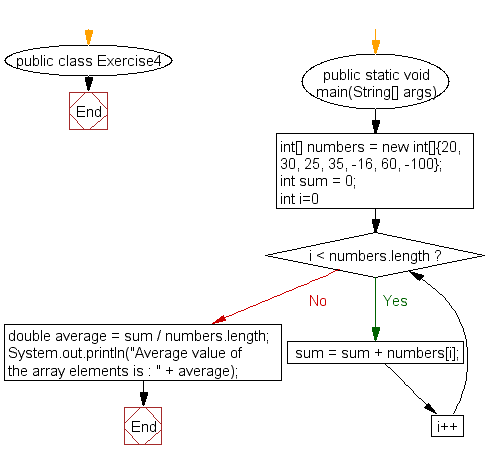
- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

**4) Write a Java program to calculate the average value of array elements.**

Flowchart:



public class Program\_4

{

public static void main(String[] args)

{

int[] arr = {1,2,3,4,5,6,7,8};

int sum = 0;

for(int i =0;i<arr.length;i++)

sum += arr[i];

double avg = sum/arr.length;

System.out.println("Average value is : " + avg);

}

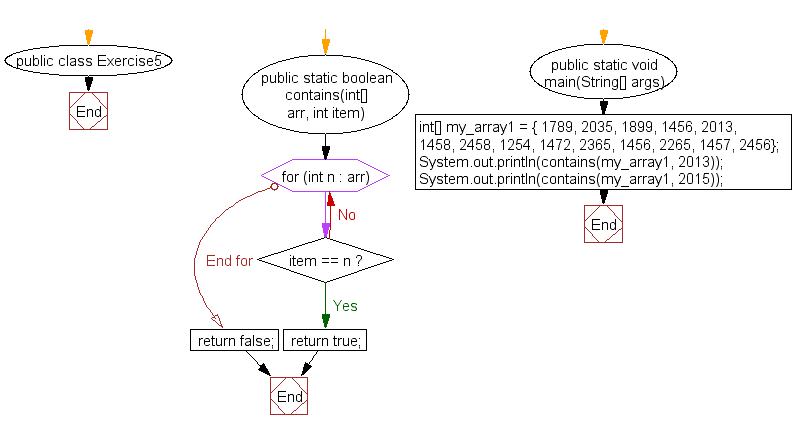
}

Output:

Average value is : 4.0

**5) Write a Java program to test if an array contains a specific value.**

Flowchart:



public class Program\_5

{

public static boolean contains(int[] arr, int item)

{

for (int n : arr) {

if (item == n)

{

return true;

}

}

return false;

}

public static void main(String[] args)

{

int[] my\_array1 = {

1789, 2035, 1899, 1456, 2013,

1458, 2458, 1254, 1472, 2365,

1456, 2265, 1457, 2456};

System.out.println(contains(my\_array1, 2013));

System.out.println(contains(my\_array1, 2015));

}

}

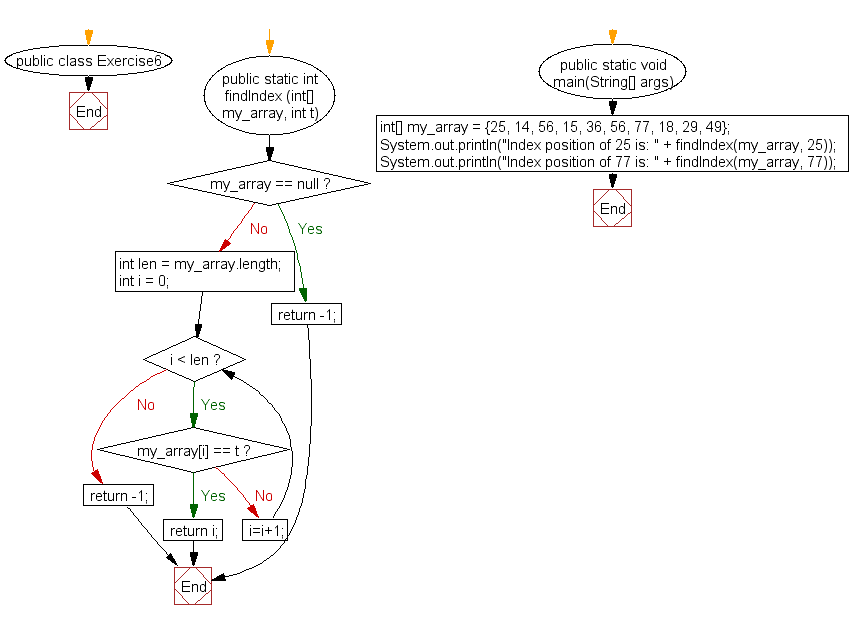
Output:

true

false

**6) Write a Java program to find the index of an array element.**

Flowchart:



public class Program\_6

{

public static int findIndex (int[] my\_array, int t)

{

if (my\_array == null) return -1;

int len = my\_array.length;

int i = 0;

while (i < len)

{

if (my\_array[i] == t) return i;

else i=i+1;

}

return -1;

}

public static void main(String[] args)

{

int[] my\_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

System.out.println("Index position of 25 is: " + findIndex(my\_array, 25));

System.out.println("Index position of 77 is: " + findIndex(my\_array, 77));

}

}

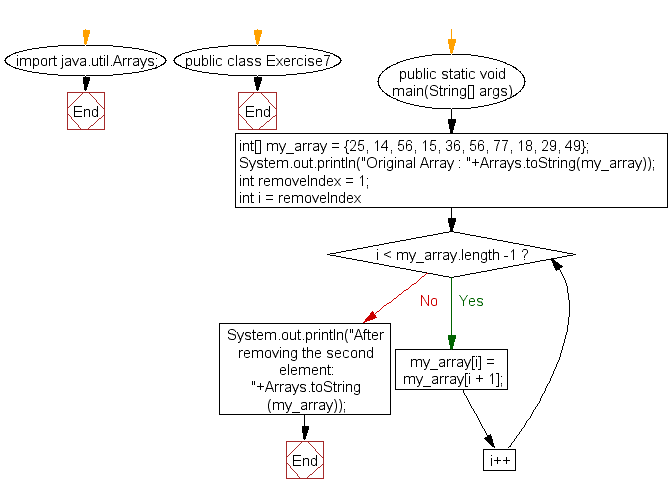
Output:

Index position of 25 is: 0

Index position of 77 is: 6

**7) Write a Java program to remove a specific element from an array.**

Flowchart:



import java.util.Arrays;

public class Program\_7

{

public static void main(String[] args)

{

int[] my\_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

System.out.println("Original Array : "+Arrays.toString(my\_array));

// Remove the second element (index->1, value->14) of the array

int removeIndex = 1;

for(int i = removeIndex; i < my\_array.length -1; i++)

{

my\_array[i] = my\_array[i + 1];

}

// We cannot alter the size of an array , after the removal, the last and second last element in the array will exist twice

System.out.println("After removing the second element: "+Arrays.toString(my\_array));

}

}

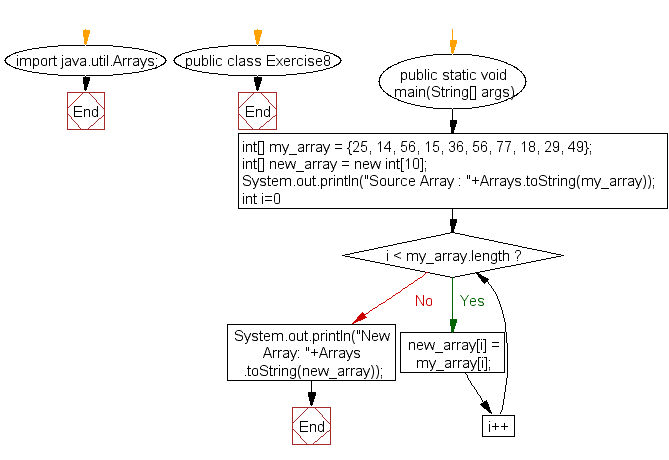
Output:

Original Array : [25, 14, 56, 15, 36, 56, 77, 18, 29, 49]

After removing the second element: [25, 56, 15, 36, 56, 77, 18, 29, 49, 49]

**8) Write a Java program to copy an array by iterating the array.**

Flowchart:



import java.util.Arrays;

public class Program\_8

{

public static void main(String[] args)

{

int[] my\_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

int[] new\_array = new int[10];

System.out.println("Source Array : "+Arrays.toString(my\_array));

for(int i=0; i < my\_array.length; i++)

{

new\_array[i] = my\_array[i];

}

System.out.println("New Array: "+Arrays.toString(new\_array));

}

}

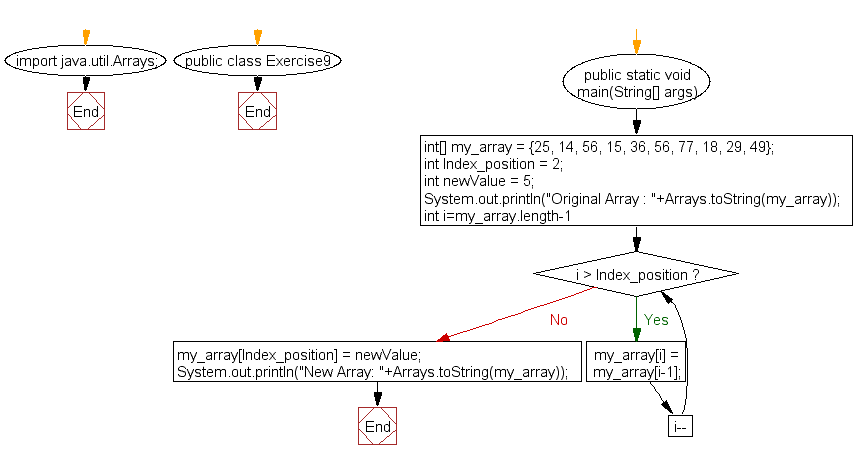
Output:

Source Array : [25, 14, 56, 15, 36, 56, 77, 18, 29, 49]

New Array: [25, 14, 56, 15, 36, 56, 77, 18, 29, 49]

**9) Write a Java program to insert an element (specific position) into an array.**

Flowchart:



import java.util.Arrays;

public class Program\_9

{

public static void main(String[] args)

{

int[] my\_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

// Insert an element in 3rd position of the array (index->2, value->5)

int Index\_position = 2;

int newValue = 5;

System.out.println("Original Array : "+Arrays.toString(my\_array));

for(int i=my\_array.length-1; i > Index\_position; i--)

{

my\_array[i] = my\_array[i-1];

}

my\_array[Index\_position] = newValue;

System.out.println("New Array: "+Arrays.toString(my\_array));

}

}

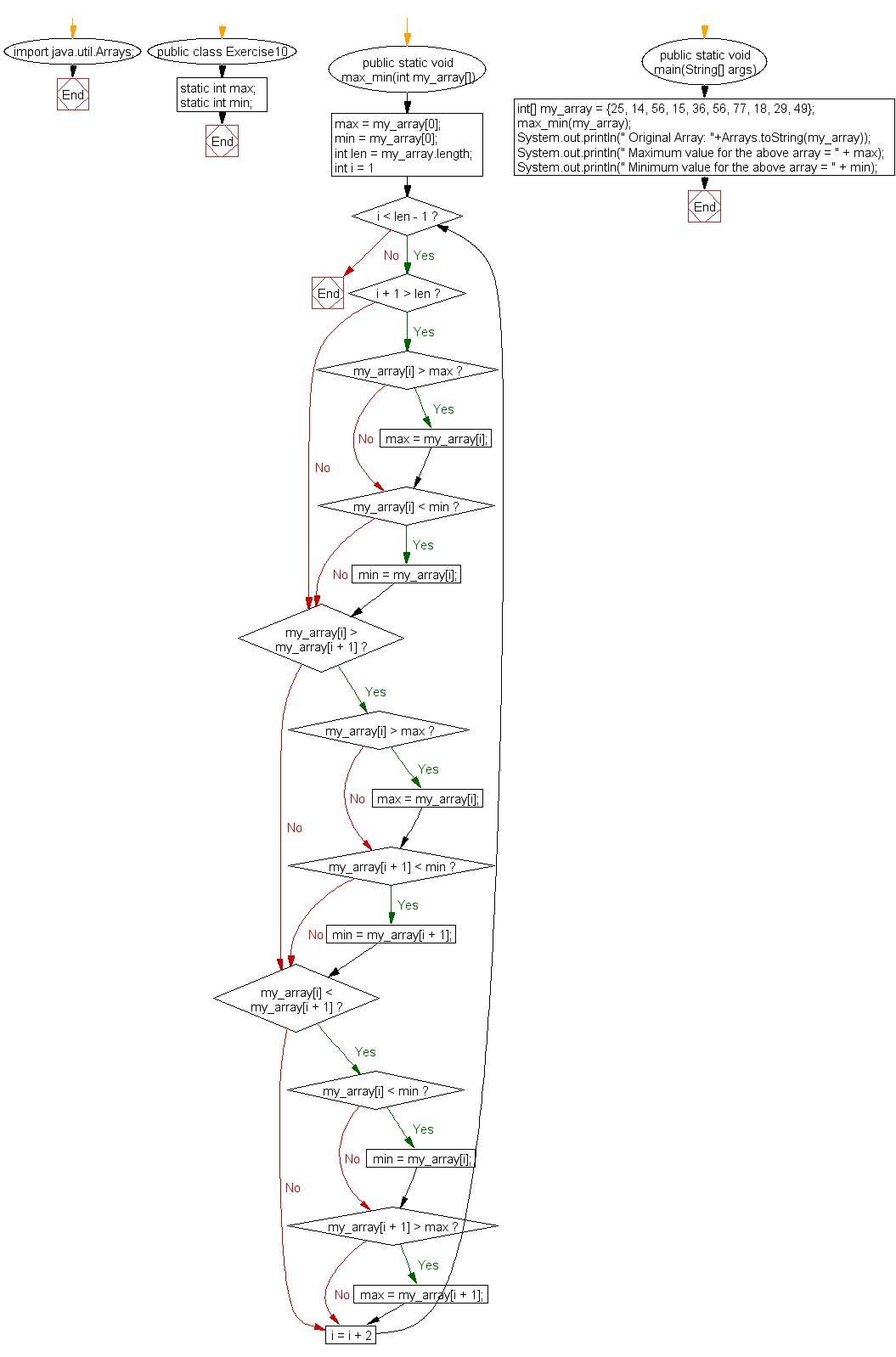
Output:

Original Array : [25, 14, 56, 15, 36, 56, 77, 18, 29, 49]

New Array: [25, 14, 5, 56, 15, 36, 56, 77, 18, 29]

**10) Write a Java program to find the maximum and minimum value of an array.**

Flowchart:



import java.util.Arrays;

public class Program\_10

{

static int max;

static int min;

public static void max\_min(int[] my\_array)

{

max = my\_array[0];

min = my\_array[0];

int len = my\_array.length;

for (int i = 1; i < len - 1; i = i + 2)

{

if (i + 1 > len)

{

if (my\_array[i] > max) max = my\_array[i];

if (my\_array[i] < min) min = my\_array[i];

}

if (my\_array[i] > my\_array[i + 1])

{

if (my\_array[i] > max) max = my\_array[i];

if (my\_array[i + 1] < min) min = my\_array[i + 1];

}

if (my\_array[i] < my\_array[i + 1])

{

if (my\_array[i] < min) min = my\_array[i];

if (my\_array[i + 1] > max) max = my\_array[i + 1];

}

}

}

public static void main(String[] args)

{

int[] my\_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

max\_min(my\_array);

System.out.println(" Original Array: "+Arrays.toString(my\_array));

System.out.println(" Maximum value for the above array = " + max);

System.out.println(" Minimum value for the above array = " + min);

}

}

Output:

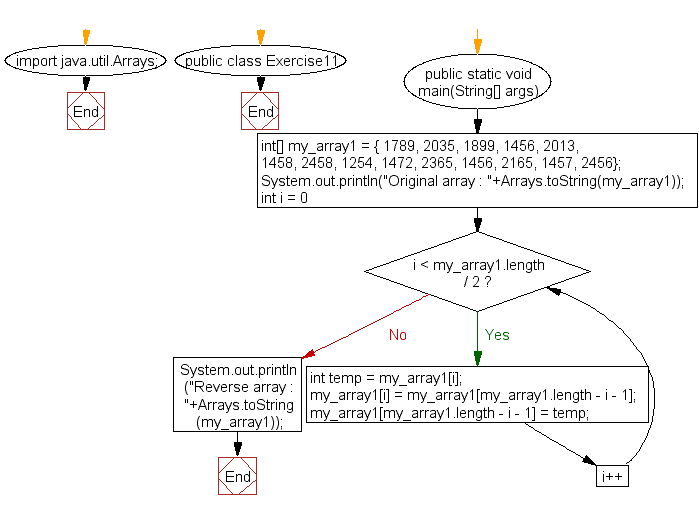
Original Array: [25, 14, 56, 15, 36, 56, 77, 18, 29, 49]

Maximum value for the above array = 77

Minimum value for the above array = 14

**11) Write a Java program to reverse an array of integer values.**

Flowchart:



import java.util.Arrays;

public class Program\_11

{

public static void main(String[] args)

{

int[] my\_array1 = {

1789, 2035, 1899, 1456, 2013,

1458, 2458, 1254, 1472, 2365,

1456, 2165, 1457, 2456};

System.out.println("Original array : "+Arrays.toString(my\_array1));

for(int i = 0; i < my\_array1.length / 2; i++)

{

int temp = my\_array1[i];

my\_array1[i] = my\_array1[my\_array1.length - i - 1];

my\_array1[my\_array1.length - i - 1] = temp;

}

System.out.println("Reverse array : "+Arrays.toString(my\_array1));

}

}

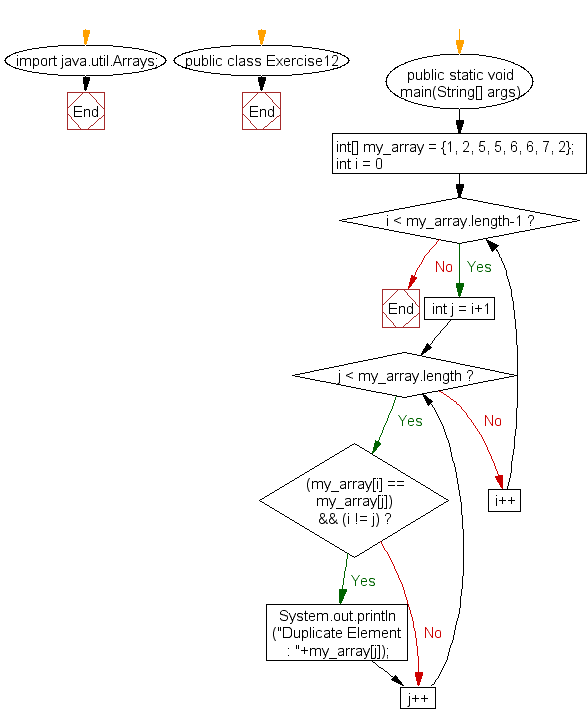
Output:

Original array : [1789, 2035, 1899, 1456, 2013, 1458, 2458, 1254, 1472, 2365, 1456, 2165, 1457, 2456]

Reverse array : [2456, 1457, 2165, 1456, 2365, 1472, 1254, 2458, 1458, 2013, 1456, 1899, 2035, 1789]

**12) Write a Java program to find the duplicate values of an array of integer values.**

Flowchart:



import java.util.Arrays;

public class Program\_12

{

public static void main(String[] args)

{

int[] my\_array = {1, 2, 5, 5, 6, 6, 7, 2};

for (int i = 0; i < my\_array.length-1; i++)

{

for (int j = i+1; j < my\_array.length; j++)

{

if ((my\_array[i] == my\_array[j]) && (i != j))

{

System.out.println("Duplicate Element : "+my\_array[j]);

}

}

}

}

}

Output:

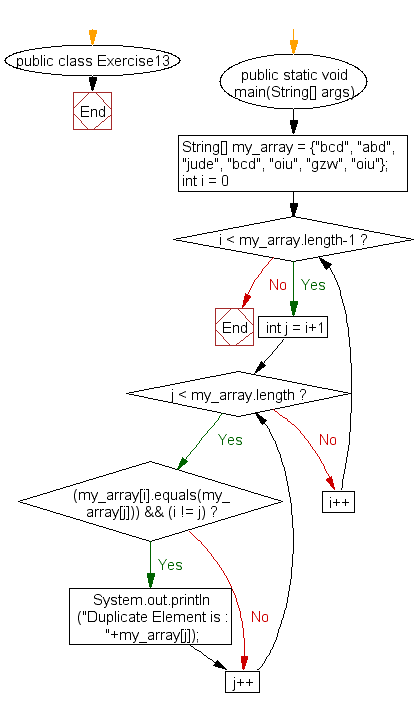
Duplicate Element : 2

Duplicate Element : 5

Duplicate Element : 6

**13)** **Write a Java program to find the duplicate values of an array of string values.**

Flowchart:



import java.util.Arrays;

public class Program\_13

{

public static void main(String[] args)

{

String[] my\_array = {"bcd", "abd", "jude", "bcd", "oiu", "gzw", "oiu"};

for (int i = 0; i < my\_array.length-1; i++)

{

for (int j = i+1; j < my\_array.length; j++)

{

if( (my\_array[i].equals(my\_array[j])) && (i != j) )

{

System.out.println("Duplicate Element is : "+my\_array[j]);

}

}

}

}

}

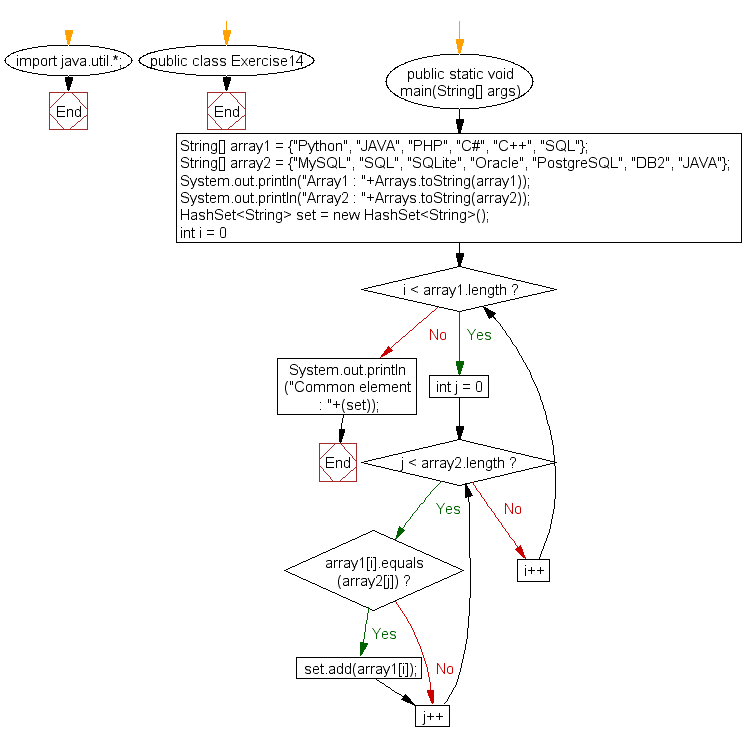
Output:

Duplicate Element is : bcd

Duplicate Element is : oiu

**14) Write a Java program to find the common elements between two arrays (string values).**

Flowchart:



import java.util.\*;

public class Program\_14

{

public static void main(String[] args)

{

String[] array1 = {"Python", "JAVA", "PHP", "C#", "C++", "SQL"};

String[] array2 = {"MySQL", "SQL", "SQLite", "Oracle", "PostgreSQL", "DB2", "JAVA"};

System.out.println("Array1 : "+Arrays.toString(array1));

System.out.println("Array2 : "+Arrays.toString(array2));

HashSet<String> set = new HashSet<String>();

for (int i = 0; i < array1.length; i++)

{

for (int j = 0; j < array2.length; j++)

{

if(array1[i].equals(array2[j]))

{

set.add(array1[i]);

}

}

}

System.out.println("Common element : "+(set));

}

}

Output:

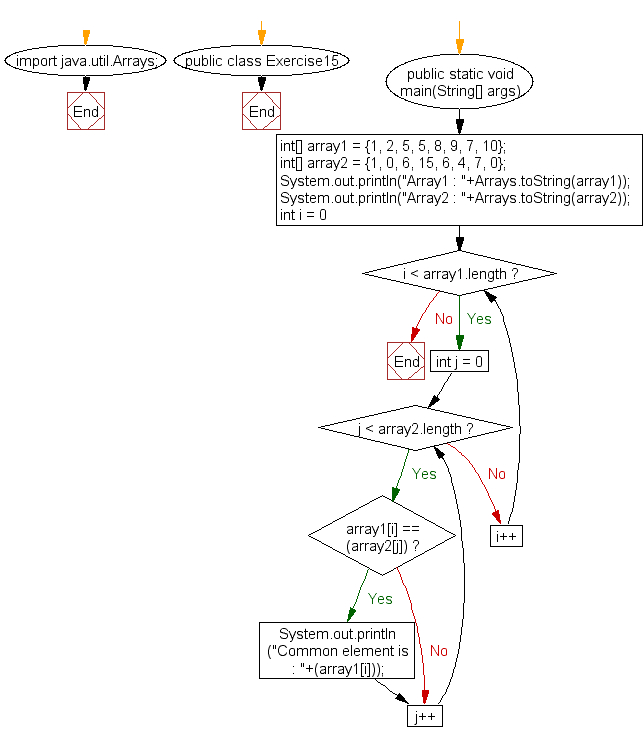
Array1 : [Python, JAVA, PHP, C#, C++, SQL]

Array2 : [MySQL, SQL, SQLite, Oracle, PostgreSQL, DB2, JAVA]

Common element : [JAVA, SQL]

**15) Write a Java program to find the common elements between two arrays of integers.**

Flowchart:



import java.util.Arrays;

public class Program\_15

{

public static void main(String[] args)

{

int[] array1 = {1, 2, 5, 5, 8, 9, 7, 10};

int[] array2 = {1, 0, 6, 15, 6, 4, 7, 0};

System.out.println("Array1 : "+Arrays.toString(array1));

System.out.println("Array2 : "+Arrays.toString(array2));

for (int i = 0; i < array1.length; i++)

{

for (int j = 0; j < array2.length; j++)

{

if(array1[i] == (array2[j]))

{

System.out.println("Common element is : "+(array1[i]));

}

}

}

}

}

Output:

Array1 : [1, 2, 5, 5, 8, 9, 7, 10]

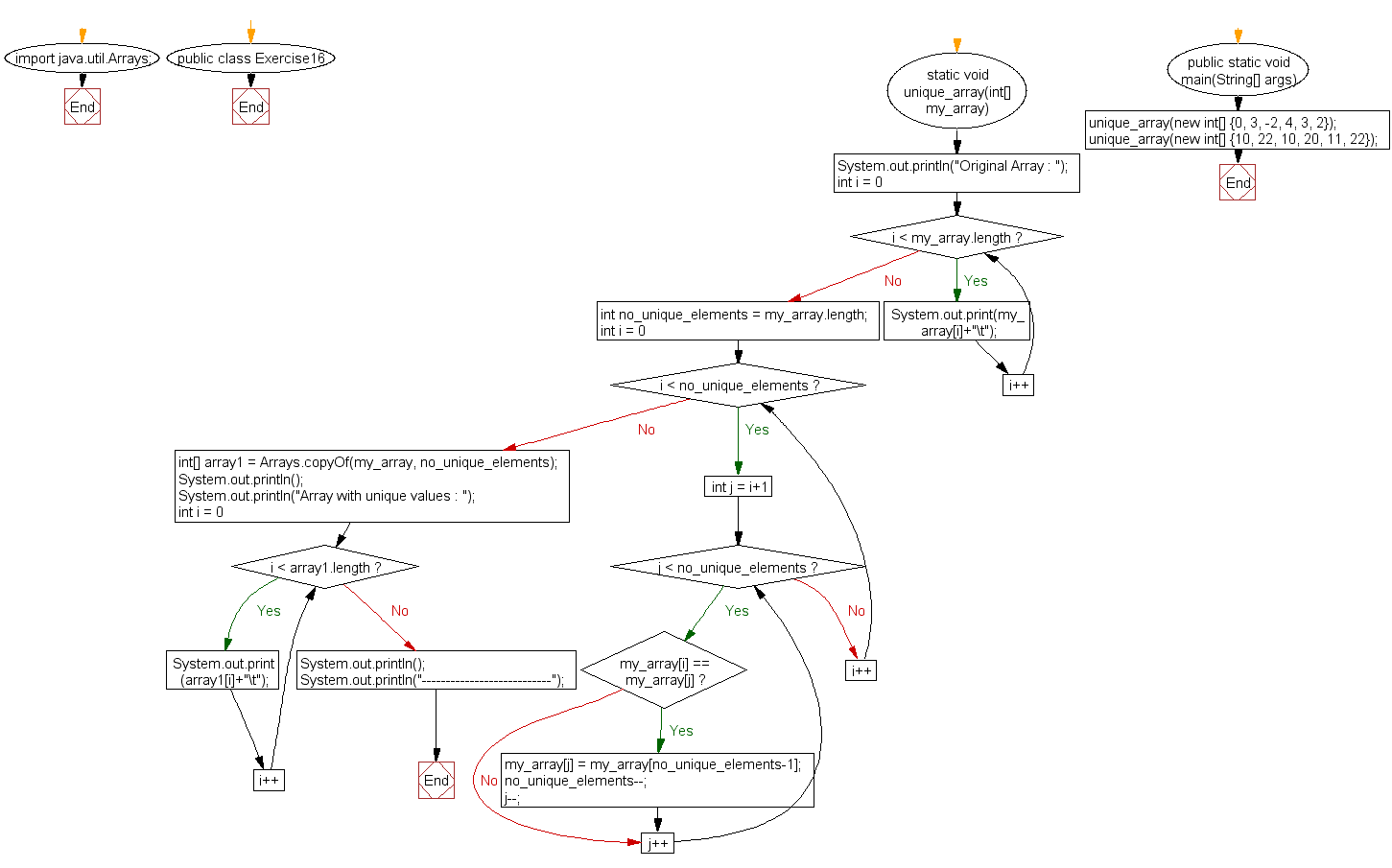
Array2 : [1, 0, 6, 15, 6, 4, 7, 0]

Common element is : 1

Common element is : 7

**16) Write a Java program to remove duplicate elements from an array.**

Flowchart:



import java.util.Arrays;

public class Program\_16

{

static void unique\_array(int[] my\_array)

{

System.out.println("Original Array : ");

for (int i = 0; i < my\_array.length; i++)

{

System.out.print(my\_array[i]+"\t");

}

//Assuming all elements in input array are unique

int no\_unique\_elements = my\_array.length;

//Comparing each element with all other elements

for (int i = 0; i < no\_unique\_elements; i++)

{

for (int j = i+1; j < no\_unique\_elements; j++)

{

//If any two elements are found equal

if(my\_array[i] == my\_array[j])

{

//Replace duplicate element with last unique element

my\_array[j] = my\_array[no\_unique\_elements-1];

no\_unique\_elements--;

j--;

}

}

}

//Copying only unique elements of my\_array into array1

int[] array1 = Arrays.copyOf(my\_array, no\_unique\_elements);

//Printing arrayWithoutDuplicates

System.out.println();

System.out.println("Array with unique values : ");

for (int i = 0; i < array1.length; i++)

{

System.out.print(array1[i]+"\t");

}

System.out.println();

System.out.println("---------------------------");

}

public static void main(String[] args)

{

unique\_array(new int[] {0, 3, -2, 4, 3, 2});

unique\_array(new int[] {10, 22, 10, 20, 11, 22});

}

}

Output:

Original Array :

0 3 -2 4 3 2

Array with unique values :

0 3 -2 4 2

---------------------------

Original Array :

10 22 10 20 11 22

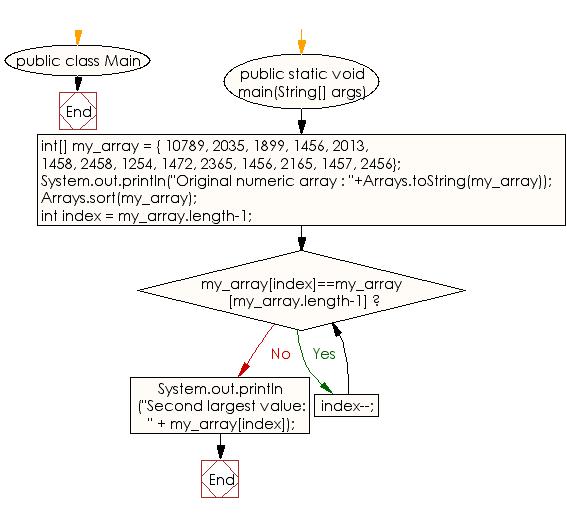
Array with unique values :

10 22 11 20

---------------------------

**17)  Write a Java program to find the second largest element in an array.**

Flowchart:



import java.util.Arrays;

public class Program\_17

{

public static void main(String[] args)

{

int[] my\_array = {

10789, 2035, 1899, 1456, 2013,

1458, 2458, 1254, 1472, 2365,

1456, 2165, 1457, 2456};

System.out.println("Original numeric array : "+Arrays.toString(my\_array));

Arrays.sort(my\_array);

int index = my\_array.length-1;

while(my\_array[index]==my\_array[my\_array.length-1])

{

index--;

}

System.out.println("Second largest value: " + my\_array[index]);

}

}

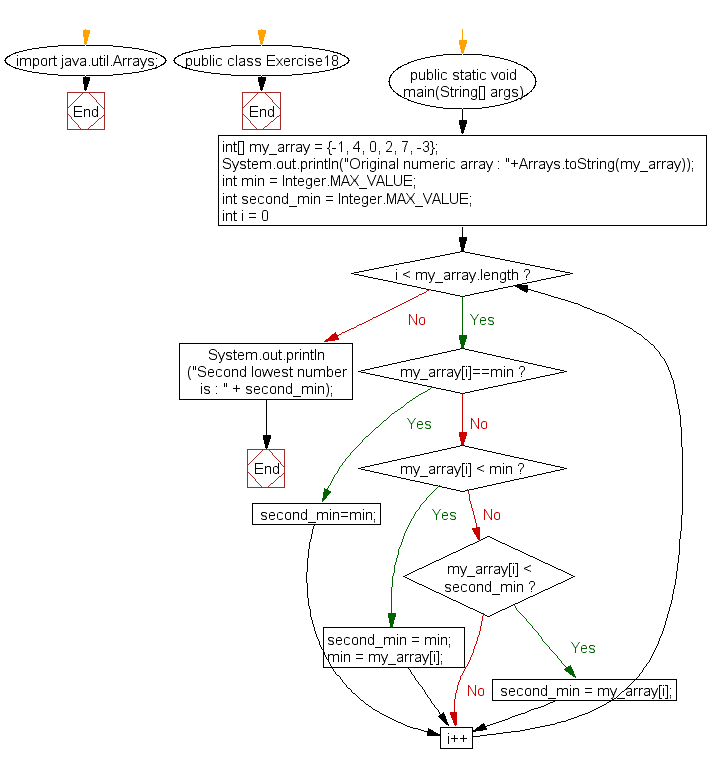
Output:

Original numeric array : [10789, 2035, 1899, 1456, 2013, 1458, 2458, 1254, 1472, 2365, 1456, 2165, 1457, 2456]

Second largest value: 2458

**18) Write a Java program to find the second smallest element in an array.**

Flowchart:



import java.util.Arrays;

public class Program\_18

{

public static void main(String[] args)

{

int[] my\_array = {-1, 4, 0, 2, 7, -3};

System.out.println("Original numeric array : "+Arrays.toString(my\_array));

int min = Integer.MAX\_VALUE;

int second\_min = Integer.MAX\_VALUE;

for (int i = 0; i < my\_array.length; i++) {

if(my\_array[i]==min){

second\_min=min;

} else if (my\_array[i] < min) {

second\_min = min;

min = my\_array[i];

} else if (my\_array[i] < second\_min) {

second\_min = my\_array[i];

}

}

System.out.println("Second lowest number is : " + second\_min);

}

}

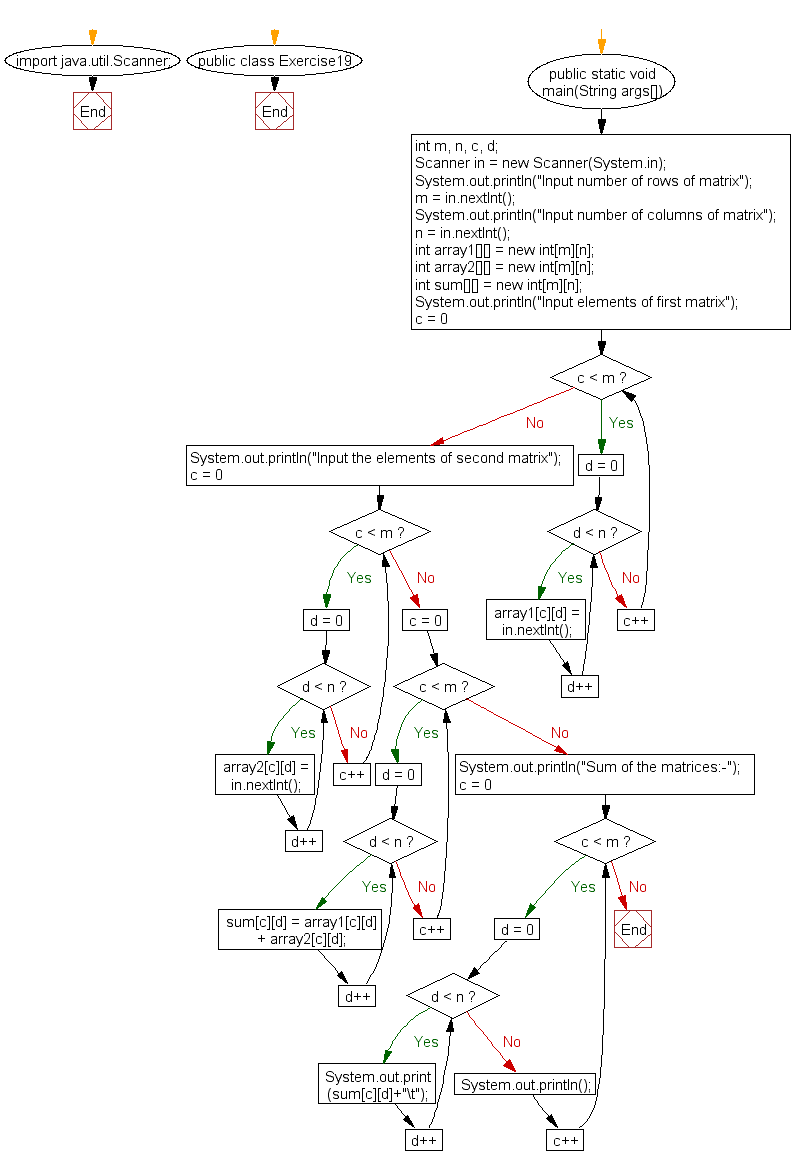
Output:

Original numeric array : [-1, 4, 0, 2, 7, -3]

Second lowest number is : -1

**19) Write a Java program to add two matrices of the same size.**

Flowchart:



import java.util.Scanner;

public class Program\_19

{

public static void main(String[] args)

{

int m, n, c, d;

Scanner in = new Scanner(System.in);

System.out.println("Input number of rows of matrix");

m = in.nextInt();

System.out.println("Input number of columns of matrix");

n = in.nextInt();

int array1[][] = new int[m][n];

int array2[][] = new int[m][n];

int sum[][] = new int[m][n];

System.out.println("Input elements of first matrix");

for ( c = 0 ; c < m ; c++ )

for ( d = 0 ; d < n ; d++ )

array1[c][d] = in.nextInt();

System.out.println("Input the elements of second matrix");

for ( c = 0 ; c < m ; c++ )

for ( d = 0 ; d < n ; d++ )

array2[c][d] = in.nextInt();

for ( c = 0 ; c < m ; c++ )

for ( d = 0 ; d < n ; d++ )

sum[c][d] = array1[c][d] + array2[c][d];

System.out.println("Sum of the matrices:-");

for ( c = 0 ; c < m ; c++ )

{

for ( d = 0 ; d < n ; d++ )

System.out.print(sum[c][d]+"\t");

System.out.println();

}

}

}

Output:

Input number of rows of matrix

2

Input number of columns of matrix

3

Input elements of first matrix

1 2 3

4 5 6

Input the elements of second matrix

5 6 7

1 3 4

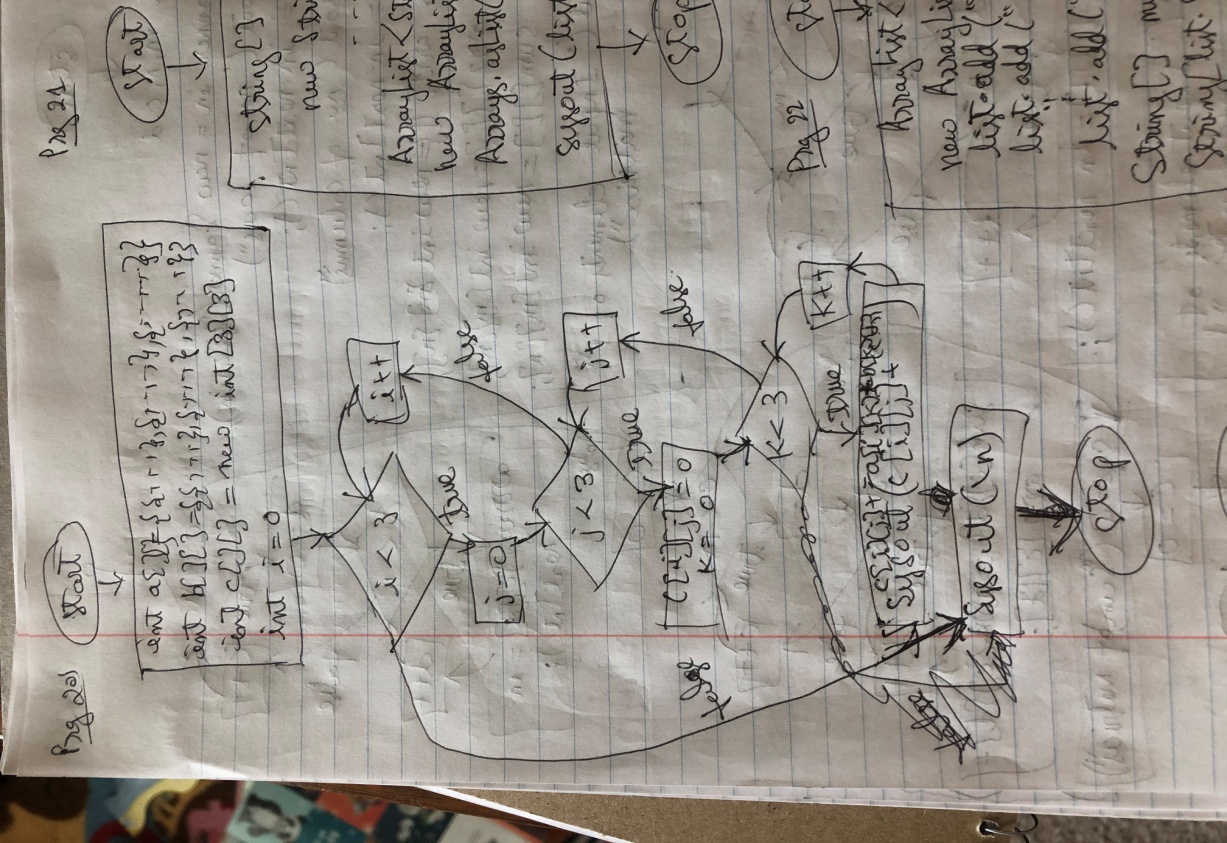
Sum of the matrices:-

6 8 10

5 8 10

**20) Write a java program to multiply two matrices of the same size.**

Flowchart:



public class Program\_20

{

public static void main(String[] args)

{

//creating two matrices

int a[][]={{1,1,1},{2,2,2},{3,3,3}};

int b[][]={{1,1,1},{2,2,2},{3,3,3}};

//creating another matrix to store the multiplication of two matrices

int c[][]=new int[3][3]; //3 rows and 3 columns

//multiplying and printing multiplication of 2 matrices

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

{

c[i][j]=0;

for(int k=0;k<3;k++)

{

c[i][j]+=a[i][k]\*b[k][j];

}//end of k loop

System.out.print(c[i][j]+" "); //printing matrix element

}//end of j loop

System.out.println();//new line

}

}

}

Output:

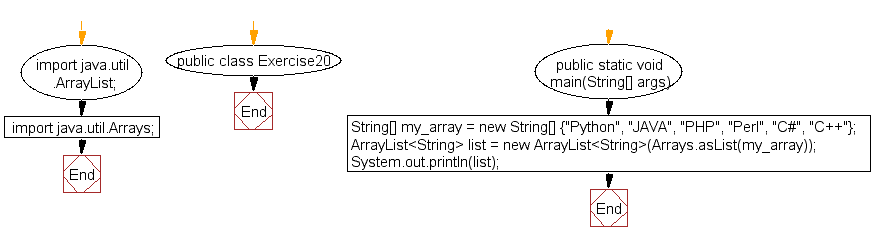
6 6 6

12 12 12

18 18 18

**21) Write a Java program to convert an array to ArrayList.**

Flowchart:



import java.util.ArrayList;

import java.util.Arrays;

public class Program\_21

{

public static void main(String[] args)

{

String[] my\_array = new String[] {"Python", "JAVA", "PHP", "Perl", "C#", "C++"};

ArrayList<String> list = new ArrayList<String>(Arrays.asList(my\_array));

System.out.println(list);

}

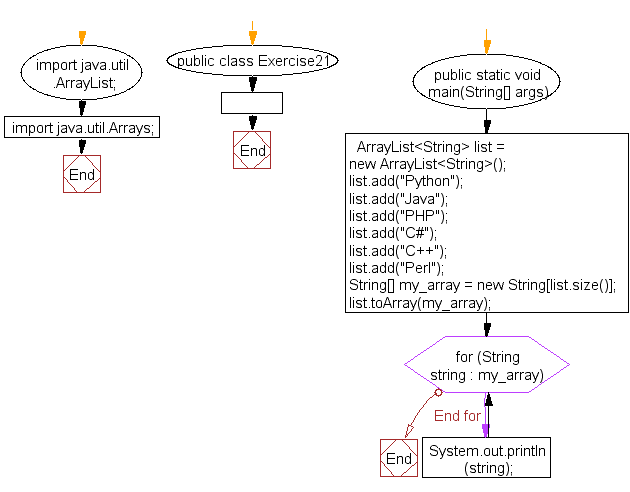
}

Output:

[Python, JAVA, PHP, Perl, C#, C++]

**22)**  **Write a Java program to convert an ArrayList to an array.**

Flowchart:



import java.util.ArrayList;

import java.util.Arrays;

public class Program\_22

{

public static void main(String[] args)

{

ArrayList<String> list = new ArrayList<String>();

list.add("Python");

list.add("Java");

list.add("PHP");

list.add("C#");

list.add("C++");

list.add("Perl");

String[] my\_array = new String[list.size()];

list.toArray(my\_array);

for (String string : my\_array)

{

System.out.println(string);

}

}

}

Output:

Python

Java

PHP

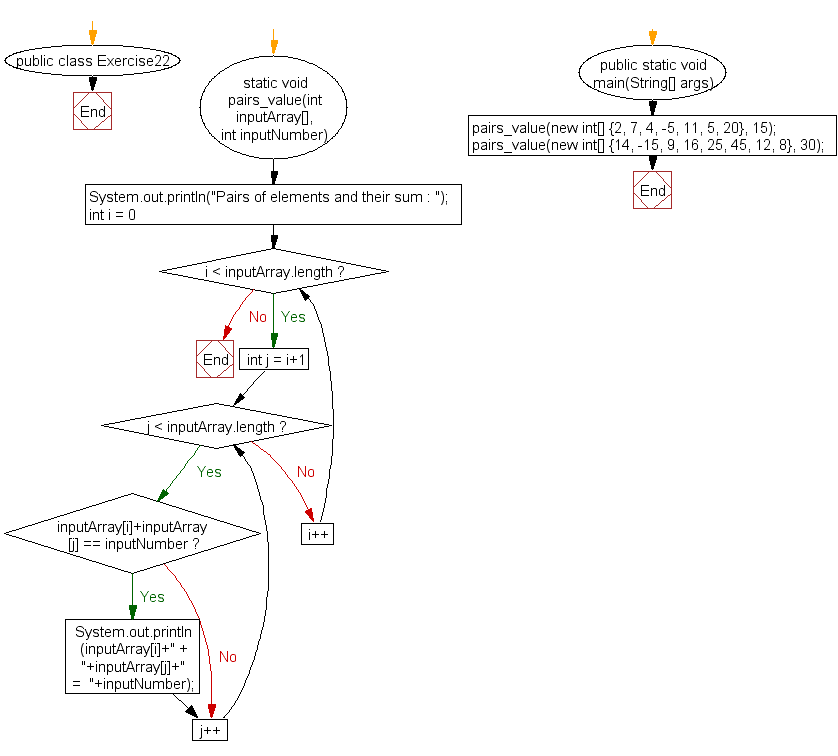
C#

C++

Perl

**23) Write a Java program to find all pairs of elements in an array whose sum is equal to a specified number**.

Flowchart:



public class Program\_23

{

static void pairs\_value(int inputArray[], int inputNumber)

{

System.out.println("Pairs of elements and their sum : ");

for (int i = 0; i < inputArray.length; i++)

{

for (int j = i+1; j < inputArray.length; j++)

{

if(inputArray[i]+inputArray[j] == inputNumber)

{

System.out.println(inputArray[i]+" + "+inputArray[j]+" = "+inputNumber);

}

}

}

}

public static void main(String[] args)

{

pairs\_value(new int[] {2, 7, 4, -5, 11, 5, 20}, 15);

pairs\_value(new int[] {14, -15, 9, 16, 25, 45, 12, 8}, 30);

}

}

Output:

Pairs of elements and their sum :

4 + 11 = 15

-5 + 20 = 15

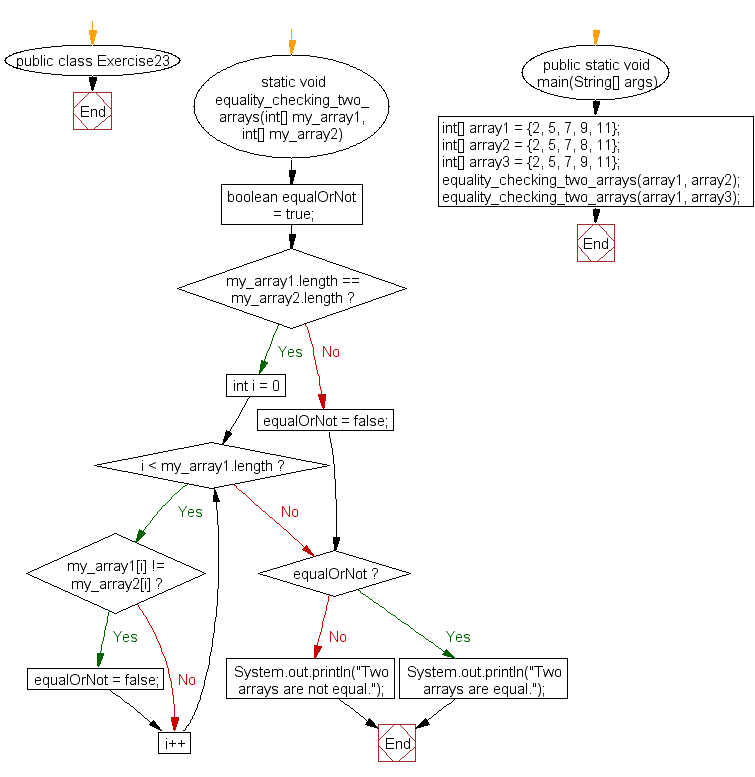
Pairs of elements and their sum :

14 + 16 = 30

-15 + 45 = 30

**24) Write a Java program to test the equality of two arrays.**

Flowchart:



public class Program\_24

{

static void equality\_checking\_two\_arrays(int[] my\_array1, int[] my\_array2)

{

boolean equalOrNot = true;

if(my\_array1.length == my\_array2.length)

{

for (int i = 0; i < my\_array1.length; i++)

{

if(my\_array1[i] != my\_array2[i])

{

equalOrNot = false;

}

}

}

else

{

equalOrNot = false;

}

if (equalOrNot)

{

System.out.println("Two arrays are equal.");

}

else

{

System.out.println("Two arrays are not equal.");

}

}

public static void main(String[] args)

{

int[] array1 = {2, 5, 7, 9, 11};

int[] array2 = {2, 5, 7, 8, 11};

int[] array3 = {2, 5, 7, 9, 11};

equality\_checking\_two\_arrays(array1, array2);

equality\_checking\_two\_arrays(array1, array3);

}

}

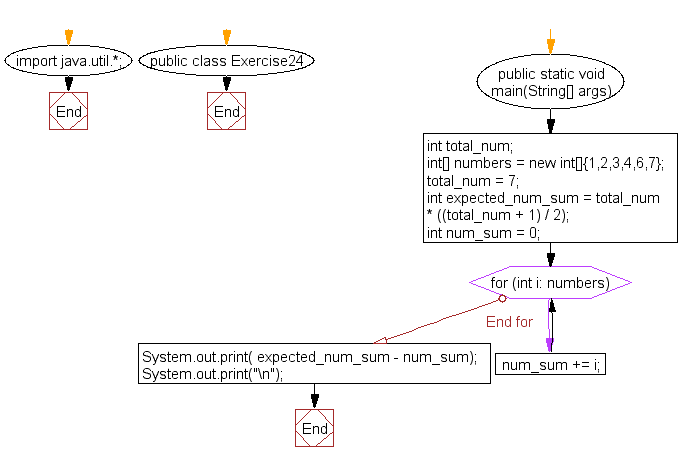
Output:

Two arrays are not equal.

Two arrays are equal.

**25) Write a Java program to find a missing number in an array.**

Flowchart:



import java.util.\*;

public class Program\_25

{

public static void main(String[] args)

{

int total\_num;

int[] numbers = new int[]{1,2,3,4,6,7};

total\_num = 7;

int expected\_num\_sum = total\_num \* ((total\_num + 1) / 2);

int num\_sum = 0;

for (int i: numbers) {

num\_sum += i;

}

System.out.print( expected\_num\_sum - num\_sum);

System.out.print("\n");

}

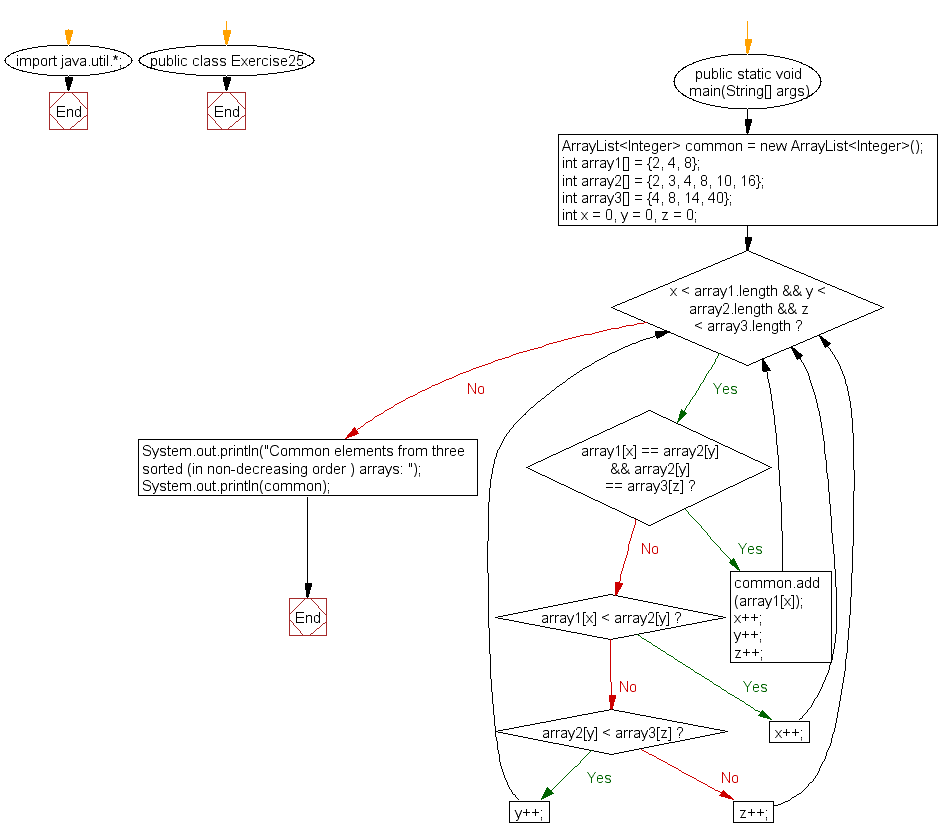
}

Output:

5

**26) Write a Java program to find common elements from three sorted (in non-decreasing order) arrays**

Flowchart:



import java.util.\*;

public class Program\_26

{

public static void main(String[] args)

{

ArrayList<Integer> common = new ArrayList<Integer>();

int array1[] = {2, 4, 8};

int array2[] = {2, 3, 4, 8, 10, 16};

int array3[] = {4, 8, 14, 40};

int x = 0, y = 0, z = 0;

while (x < array1.length && y < array2.length && z < array3.length)

{

if (array1[x] == array2[y] && array2[y] == array3[z])

{

common.add(array1[x]);

x++;

y++;

z++;

}

else if (array1[x] < array2[y])

x++;

else if (array2[y] < array3[z])

y++;

else

z++;

}

System.out.println("Common elements from three sorted (in non-decreasing order ) arrays: ");

System.out.println(common);

}

}

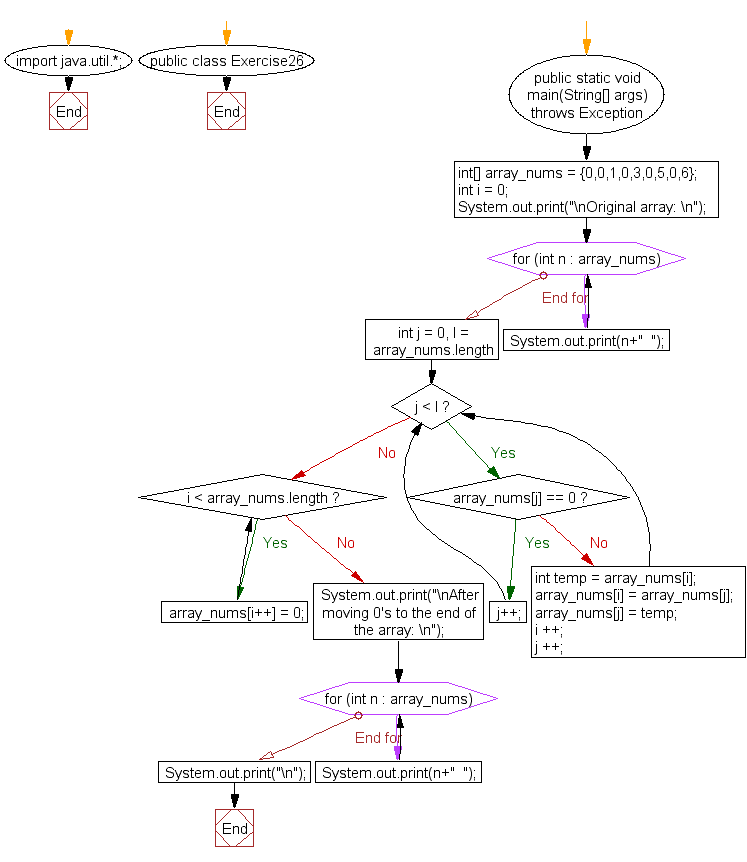
Output:

Common elements from three sorted (in non-decreasing order ) arrays:

[4, 8]

**27) Write a Java program to move all 0's to the end of an array. Maintain the relative order of the other (non-zero) array elements.**

Flowchart:



public static void main(String[] args)

{

int[] array\_nums = {0,0,1,0,3,0,5,0,6};

int i = 0;

System.out.print("\nOriginal array: \n");

for (int n : array\_nums)

System.out.print(n+" ");

for(int j = 0, l = array\_nums.length; j < l;)

{

if(array\_nums[j] == 0)

j++;

else

{

int temp = array\_nums[i];

array\_nums[i] = array\_nums[j];

array\_nums[j] = temp;

i ++;

j ++;

}

}

while (i < array\_nums.length)

array\_nums[i++] = 0;

System.out.print("\nAfter moving 0's to the end of the array: \n");

for (int n : array\_nums)

System.out.print(n+" ");

System.out.print("\n");

}

}

Output:

Original array:

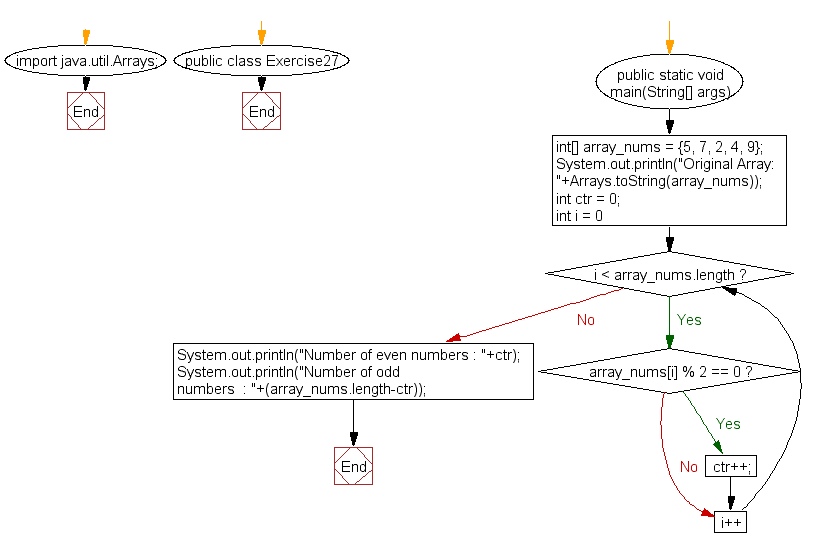
0 0 1 0 3 0 5 0 6

After moving 0's to the end of the array:

1 3 5 6 0 0 0 0 0

**28) Write a Java program to find the number of even and odd integers in a given array of integers.**

Flowchart:



import java.util.Arrays;

public class Program\_28

{

public static void main(String[] args)

{

int[] array\_nums = {5, 7, 2, 4, 9};

System.out.println("Original Array: "+Arrays.toString(array\_nums));

int ctr = 0;

for(int i = 0; i < array\_nums.length; i++)

{

if(array\_nums[i] % 2 == 0)

ctr++;

}

System.out.println("Number of even numbers : "+ctr);

System.out.println("Number of odd numbers : "+(array\_nums.length-ctr));

}

}

Output:

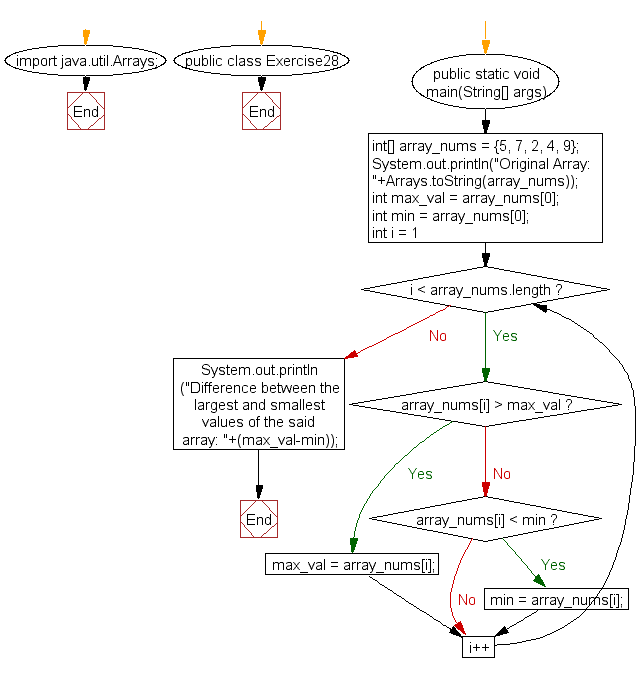
Original Array: [5, 7, 2, 4, 9]

Number of even numbers : 2

Number of odd numbers : 3

**29) Write a Java program to get the difference between the largest and smallest values in an array of integers. The length of the array must be 1 and above.**

FLowchart:



import java.util.Arrays;

public class Program\_29

{

public static void main(String[] args)

{

int[] array\_nums = {5, 7, 2, 4, 9};

System.out.println("Original Array: "+Arrays.toString(array\_nums));

int max\_val = array\_nums[0];

int min = array\_nums[0];

for(int i = 1; i < array\_nums.length; i++)

{

if(array\_nums[i] > max\_val)

max\_val = array\_nums[i];

else if(array\_nums[i] < min)

min = array\_nums[i];

}

System.out.println("Difference between the largest and smallest values of the said array: "+(max\_val-min));

}

}

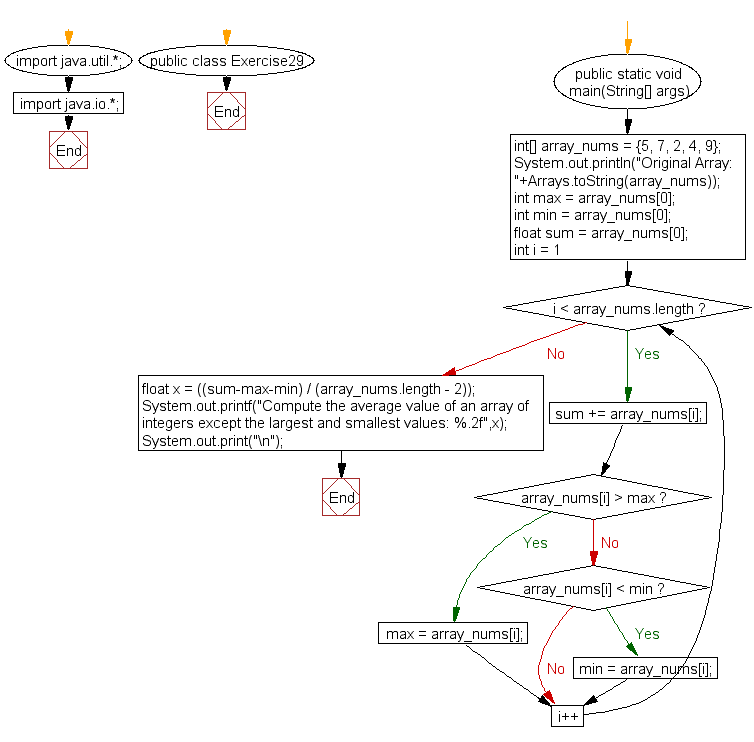
Output:

Original Array: [5, 7, 2, 4, 9]

Difference between the largest and smallest values of the said array: 7

**30) Write a Java program to compute the average value of an array of integers except the largest and smallest values.**

Flowchart:



import java.util.\*;

import java.io.\*;

public class Program\_30

{

public static void main(String[] args)

{

int[] array\_nums = {5, 7, 2, 4, 9};

System.out.println("Original Array: "+Arrays.toString(array\_nums));

int max = array\_nums[0];

int min = array\_nums[0];

float sum = array\_nums[0];

for(int i = 1; i < array\_nums.length; i++)

{

sum += array\_nums[i];

if(array\_nums[i] > max)

max = array\_nums[i];

else if(array\_nums[i] < min)

min = array\_nums[i];

}

float x = ((sum-max-min) / (array\_nums.length - 2));

System.out.printf("Compute the average value of an array of integers except the largest and smallest values: %.2f",x);

System.out.print("\n");

}

}

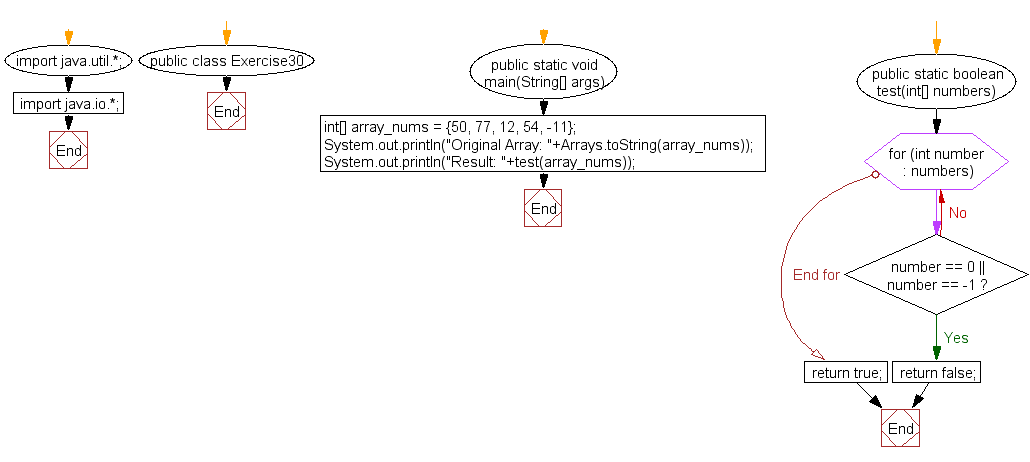
Output:

Original Array: [5, 7, 2, 4, 9]

Compute the average value of an array of integers except the largest and smallest values: 5.33

**31) Write a Java program to check if an array of integers without 0 and -1.**

Flowchart:



import java.util.\*;

import java.io.\*;

public class Program\_31

{

public static void main(String[] args)

{

int[] array\_nums = {50, 77, 12, 54, -11};

int[] array\_nums1 = {50, 77, 12, 54, 0,-1};

System.out.println("Original Array1: "+Arrays.toString(array\_nums));

System.out.println("Result: "+test(array\_nums));

System.out.println("Original Array2: "+Arrays.toString(array\_nums1));

System.out.println("Result: "+test(array\_nums1));

}

public static boolean test(int[] numbers)

{

for (int number : numbers)

{

if (number == 0 || number == -1)

{

return false;

}

}

return true;

}

}

Output:

Original Array1: [50, 77, 12, 54, -11]

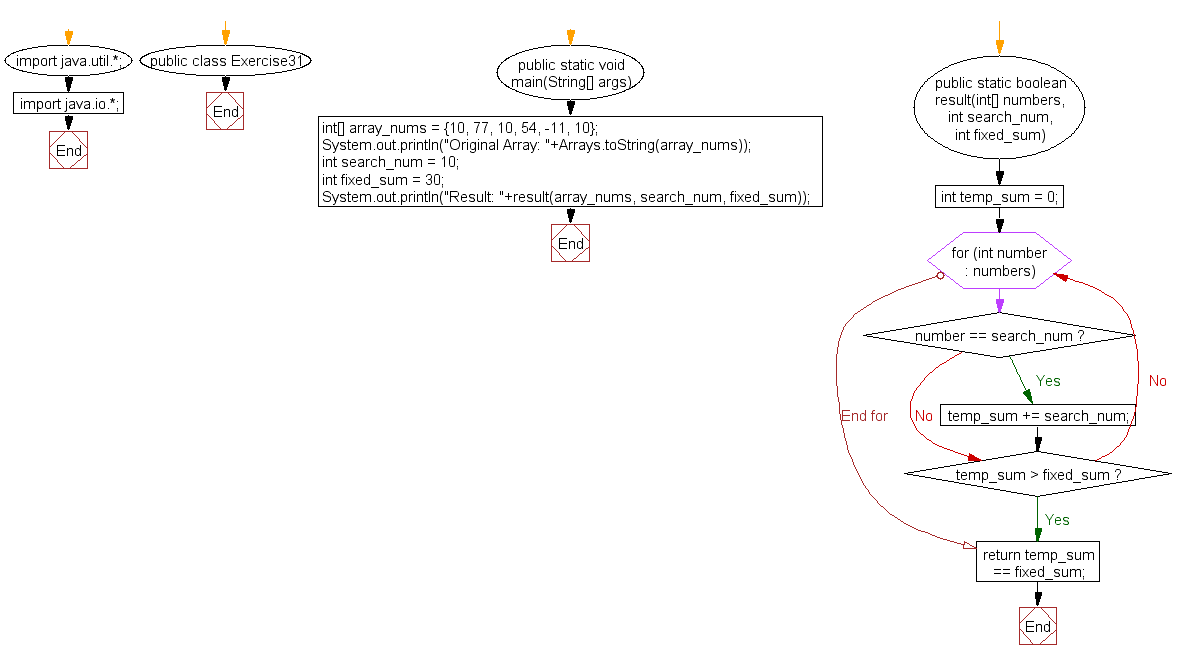
Result: true

Original Array2: [50, 77, 12, 54, 0, -1]

Result: false

**32) Write a Java program to check if the sum of all the 10's in the array is exactly 30. Return false if the condition does not satisfy, otherwise true.**

Flowchart:



import java.util.\*;

import java.io.\*;

public class Program\_32

{

public static void main(String[] args)

{

int[] array\_nums = {10, 77, 10, 54, -11, 10};

System.out.println("Original Array: "+Arrays.toString(array\_nums));

int search\_num = 10;

int fixed\_sum = 30;

System.out.println("Result: "+result(array\_nums, search\_num, fixed\_sum));

}

public static boolean result(int[] numbers, int search\_num, int fixed\_sum)

{

int temp\_sum = 0;

for (int number : numbers)

{

if (number == search\_num)

{

temp\_sum += search\_num;

}

if (temp\_sum > fixed\_sum)

{

break;

}

}

return temp\_sum == fixed\_sum;

}

}

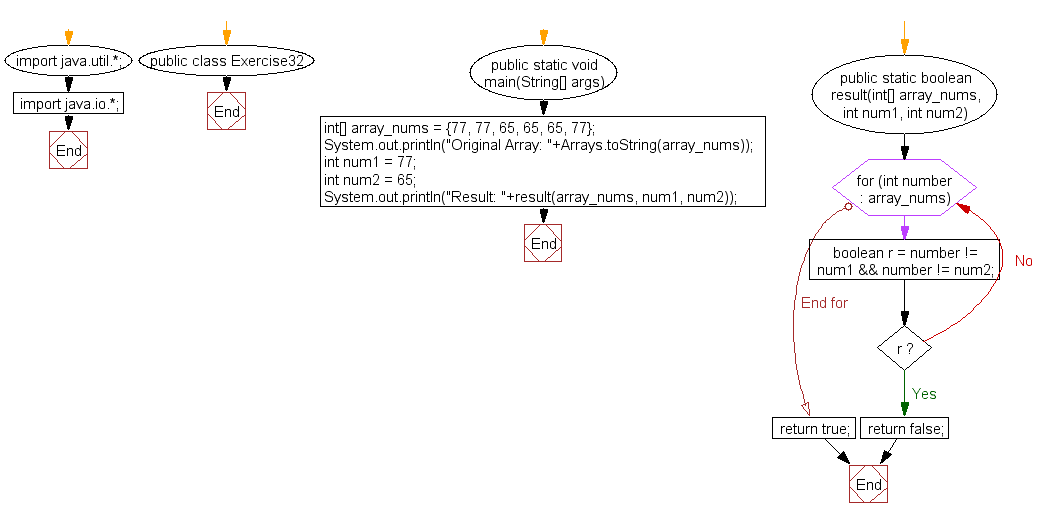
Output:

Original Array: [10, 77, 10, 54, -11, 10]

Result: true

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

Flowchart:



import java.util.\*;

import java.io.\*;

public class Program\_33

{

public static void main(String[] args)

{

int[] array\_nums = {77, 77, 65, 65, 65, 77};

System.out.println("Original Array: "+Arrays.toString(array\_nums));

int num1 = 77;

int num2 = 65;

System.out.println("Result: "+result(array\_nums, num1, num2));

}

public static boolean result(int[] array\_nums, int num1, int num2)

{

for (int number : array\_nums)

{

boolean r = number != num1 && number != num2;

if (r)

{

return false;

}

}

return true;

}

}

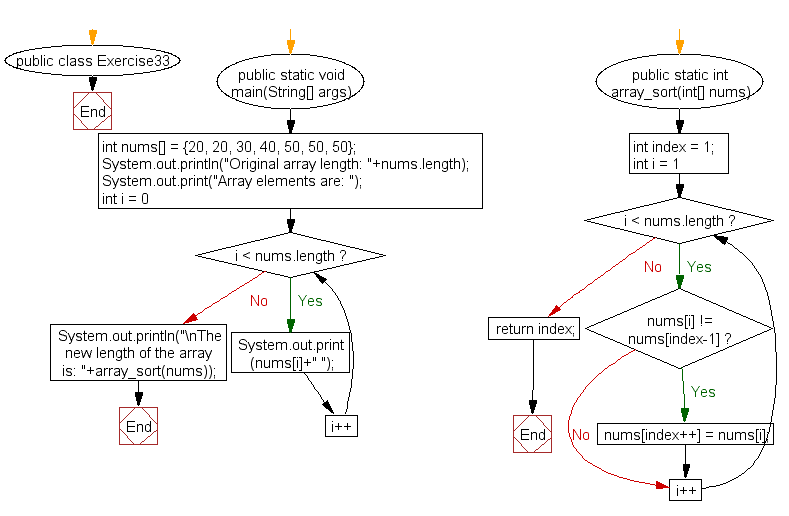
Output:

Original Array: [77, 77, 65, 65, 65, 77]

Result: true

**34)** **Write a Java program to remove the duplicate elements of a given array and return 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.**

Flowchart:



public class Program\_34

{

public static void main(String[] args)

{

int nums[] = {20, 20, 30, 40, 50, 50, 50};

System.out.println("Original array length: "+nums.length);

System.out.print("Array elements are: ");

for (int i = 0; i < nums.length; i++)

{

System.out.print(nums[i]+" ");

}

System.out.println("\nThe new length of the array is: "+array\_sort(nums));

}

public static int array\_sort(int[] nums)

{

int index = 1;

for (int i = 1; i < nums.length; i++)

{

if (nums[i] != nums[index-1])

nums[index++] = nums[i];

}

return index;

}

}

Output::

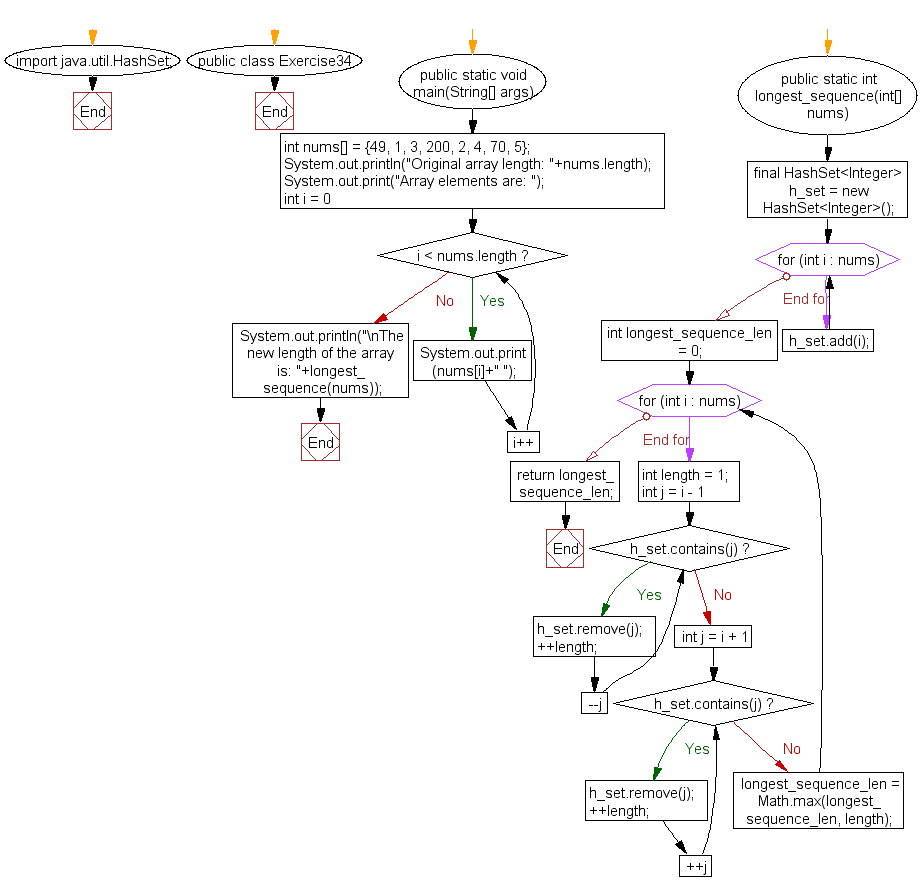
Original array length: 7

Array elements are: 20 20 30 40 50 50 50

The new length of the array is: 4

**35) Write a Java program to find the length of the longest consecutive elements sequence from a given unsorted array of integers.  
Sample array: [49, 1, 3, 200, 2, 4, 70, 5]  
The longest consecutive elements sequence is [1, 2, 3, 4, 5], therefore the program will return its length 5.**

Flowchart:



import java.util.HashSet;

public class Program\_35

{

public static void main(String[] args)

{

int nums[] = {49, 1, 3, 200, 2, 4, 70, 5};

System.out.println("Original array length: "+nums.length);

System.out.print("Array elements are: ");

for (int i = 0; i < nums.length; i++)

{

System.out.print(nums[i]+" ");

}

System.out.println("\nThe new length of the array is: "+longest\_sequence(nums));

}

public static int longest\_sequence(int[] nums)

{

final HashSet<Integer> h\_set = new HashSet<Integer>();

for (int i : nums) h\_set.add(i);

int longest\_sequence\_len = 0;

for (int i : nums)

{

int length = 1;

for (int j = i - 1; h\_set.contains(j); --j)

{

h\_set.remove(j);

++length;

}

for (int j = i + 1; h\_set.contains(j); ++j)

{

h\_set.remove(j);

++length;

}

longest\_sequence\_len = Math.max(longest\_sequence\_len, length);

}

return longest\_sequence\_len;

}

}

Output:

Original array length: 8

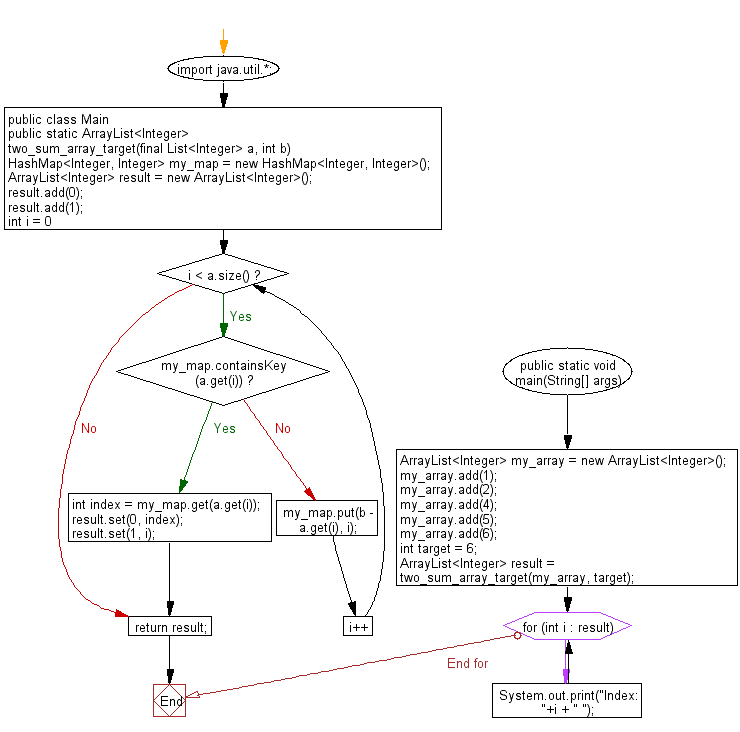
Array elements are: 49 1 3 200 2 4 70 5

The new length of the array is: 5

**36) 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.**

Flowchart:



import java.util.\*;

public class Program\_36

{

public static ArrayList<Integer> two\_sum\_array\_target(final List<Integer> a, int b)

{

HashMap<Integer, Integer> my\_map = new HashMap<Integer, Integer>();

ArrayList<Integer> result = new ArrayList<Integer>();

result.add(0);

result.add(1);

for(int i = 0; i < a.size(); i++)

{

if(my\_map.containsKey(a.get(i)))

{

int index = my\_map.get(a.get(i));

result.set(0, index );

result.set(1, i );

break;

}

else

{

my\_map.put(b - a.get(i), i);

}

}

return result;

}

public static void main(String[] args)

{

ArrayList<Integer> my\_array = new ArrayList<Integer>();

my\_array.add(1);

my\_array.add(2);

my\_array.add(4);

my\_array.add(5);

my\_array.add(6);

int target = 6;

ArrayList<Integer> result = two\_sum\_array\_target(my\_array, target);

for(int i : result)

System.out.print("Index: "+i + " ");

}

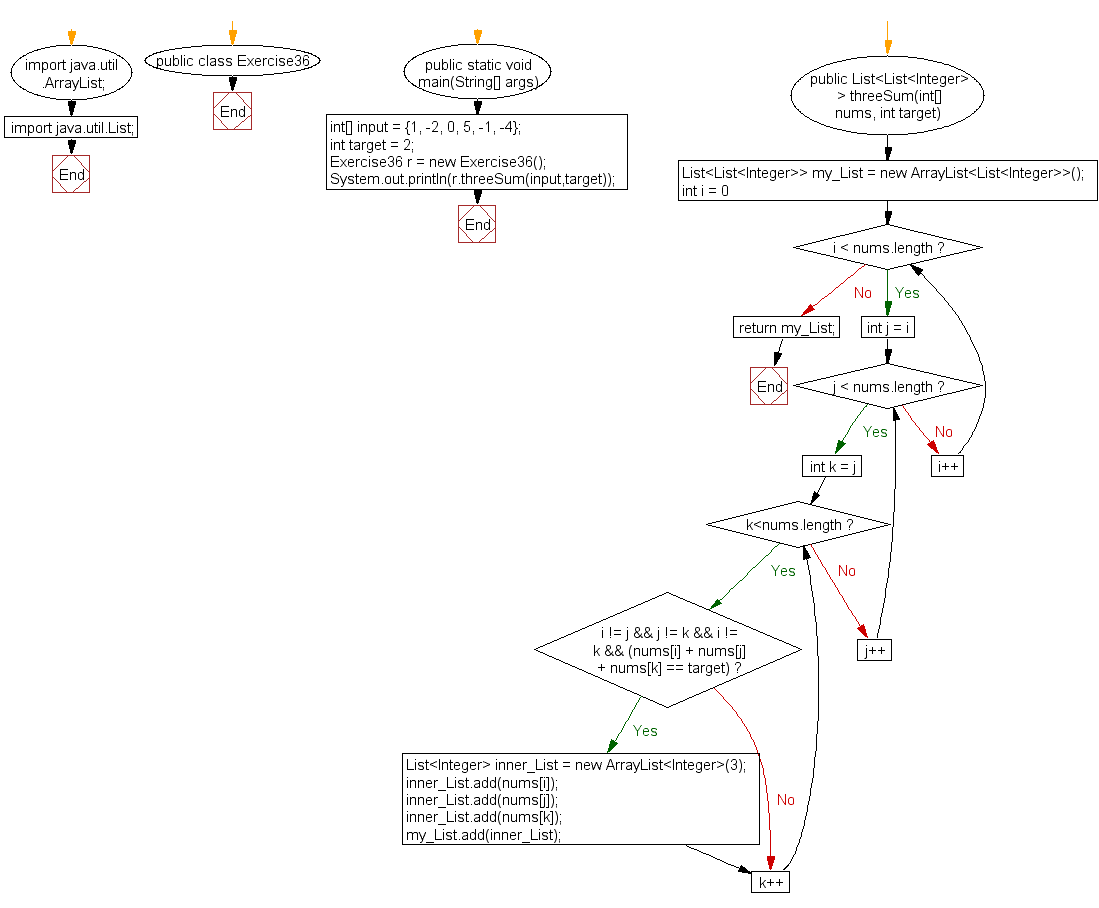
}

Output:

Index: 1 Index: 2

**37) Write a Java program to find all the unique triplets such that sum of all the three elements [x, y, z (x ≤ y ≤ z)] equal to a specified number.  
Sample array: [1, -2, 0, 5, -1, -4]  
Target value: 2.**

Flowchart:



import java.util.ArrayList;

import java.util.List;

public class Program\_37

{

public static void main(String[] args)

{

int[] input = {1, -2, 0, 5, -1, -4};

int target = 2;

Program\_37 r = new Program\_37();

System.out.println(r.threeSum(input,target));

}

public List<List<Integer>> threeSum(int[] nums, int target)

{

List<List<Integer>> my\_List = new ArrayList<List<Integer>>();

for(int i = 0; i < nums.length; i++)

{

for(int j = i; j < nums.length ;j++)

{

for(int k = j; k<nums.length;k++)

{

if ( i != j && j != k && i != k && (nums[i] + nums[j] + nums[k] == target)){

List<Integer> inner\_List = new ArrayList<Integer>(3);

inner\_List.add(nums[i]);

inner\_List.add(nums[j]);

inner\_List.add(nums[k]);

my\_List.add(inner\_List);

}

}

}

}

return my\_List;

}

}

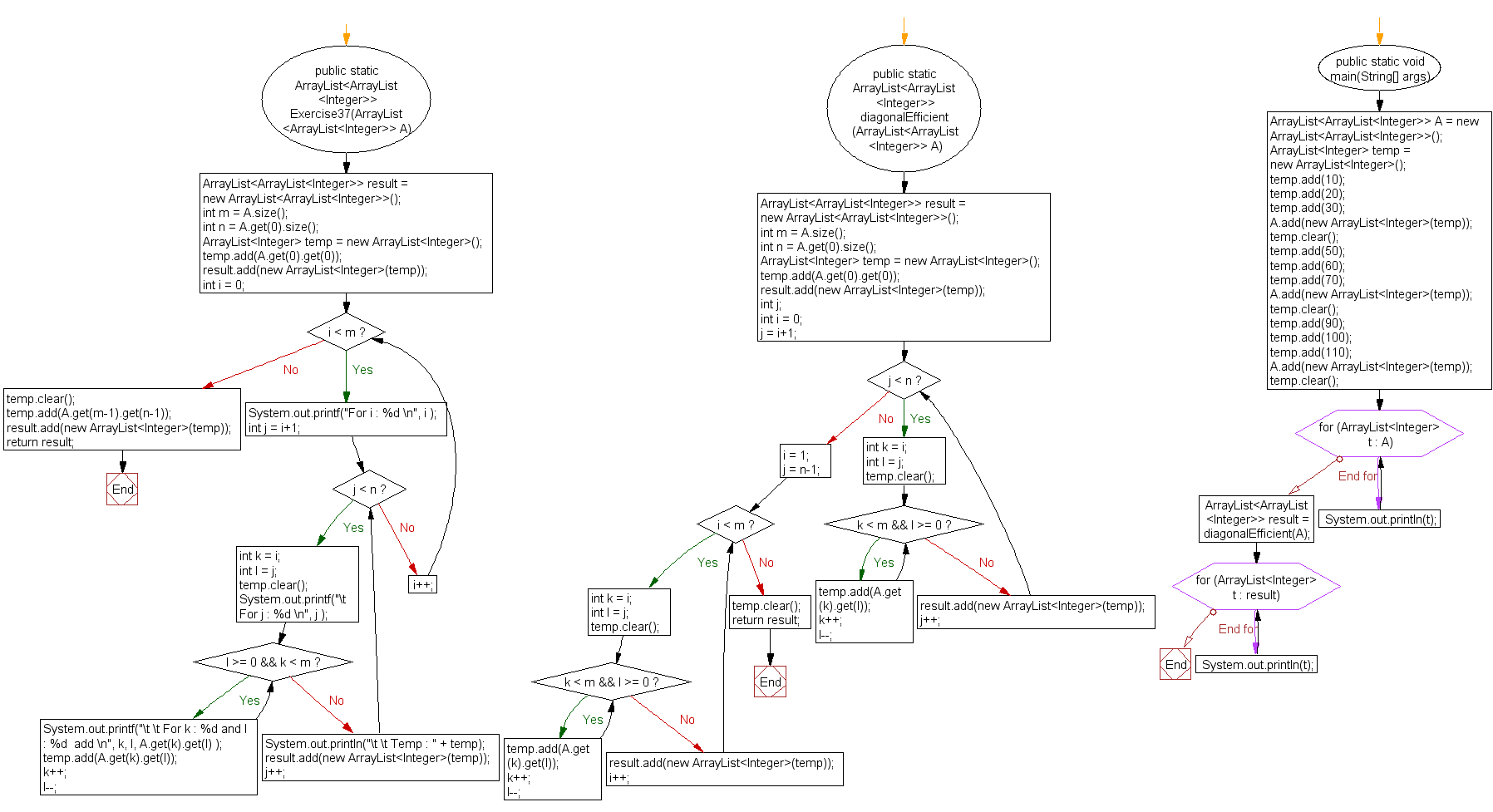
Output:

[[1, 5, -4], [-2, 5, -1]]

**38) Write a Java program to create an array of its anti-diagonals from a given square matrix.**

**Example:  
Input :  
1 2  
3 4  
Output:  
[  
[1],  
[2, 3],  
[4]  
]**

Flowchart:



import java.util.\*;

public class Program\_38

{

public static ArrayList<ArrayList<Integer>> Program\_38(ArrayList<ArrayList<Integer>> A)

{

ArrayList<ArrayList<Integer>> result = new ArrayList<ArrayList<Integer>>();

int m = A.size();

int n = A.get(0).size();

ArrayList<Integer> temp = new ArrayList<Integer>();

temp.add(A.get(0).get(0));

result.add(new ArrayList<Integer>(temp));

int i = 0;

while(i < m)

{

System.out.printf("For i : %d \n", i );

int j = i+1;

while(j < n)

{

int k = i;

int l = j;

temp.clear();

System.out.printf("\t For j : %d \n", j );

while(l >= 0 && k < m)

{

System.out.printf("\t \t For k : %d and l : %d add \n", k, l, A.get(k).get(l) );

temp.add(A.get(k).get(l));

k++;

l--;

}

System.out.println("\t \t Temp : " + temp);

result.add(new ArrayList<Integer>(temp));

j++;

}

i++;

}

temp.clear();

temp.add(A.get(m-1).get(n-1));

result.add(new ArrayList<Integer>(temp));

return result;

}

public static ArrayList<ArrayList<Integer>> diagonalEfficient(ArrayList<ArrayList<Integer>> A)

{

ArrayList<ArrayList<Integer>> result = new ArrayList<ArrayList<Integer>>();

int m = A.size();

int n = A.get(0).size();

ArrayList<Integer> temp = new ArrayList<Integer>();

temp.add(A.get(0).get(0));

result.add(new ArrayList<Integer>(temp));

int j;

int i = 0;

j = i+1;

while(j < n)

{

int k = i;

int l = j;

temp.clear();

while(k < m && l >= 0)

{

temp.add(A.get(k).get(l));

k++;

l--;

}

result.add(new ArrayList<Integer>(temp));

j++;

}

i = 1;

j = n-1;

while(i < m)

{

int k = i;

int l = j;

temp.clear();

while(k < m && l >= 0)

{

temp.add(A.get(k).get(l));

k++;

l--;

}

result.add(new ArrayList<Integer>(temp));

i++;

}

temp.clear();

return result;

}

public static void main(String[] args)

{

ArrayList<ArrayList<Integer>> A = new ArrayList<ArrayList<Integer>>();

ArrayList<Integer> temp = new ArrayList<Integer>();

temp.add(10);

temp.add(20);

temp.add(30);

A.add(new ArrayList<Integer>(temp));

temp.clear();

temp.add(50);

temp.add(60);

temp.add(70);

A.add(new ArrayList<Integer>(temp));

temp.clear();

temp.add(90);

temp.add(100);

temp.add(110);

A.add(new ArrayList<Integer>(temp));

temp.clear();

for(ArrayList<Integer> t : A)

System.out.println(t);

ArrayList<ArrayList<Integer>> result = diagonalEfficient(A);

for(ArrayList<Integer> t : result)

System.out.println(t);

}

}

Output:

[10, 20, 30]

[50, 60, 70]

[90, 100, 110]

[10]

[20, 50]

[30, 60, 90]

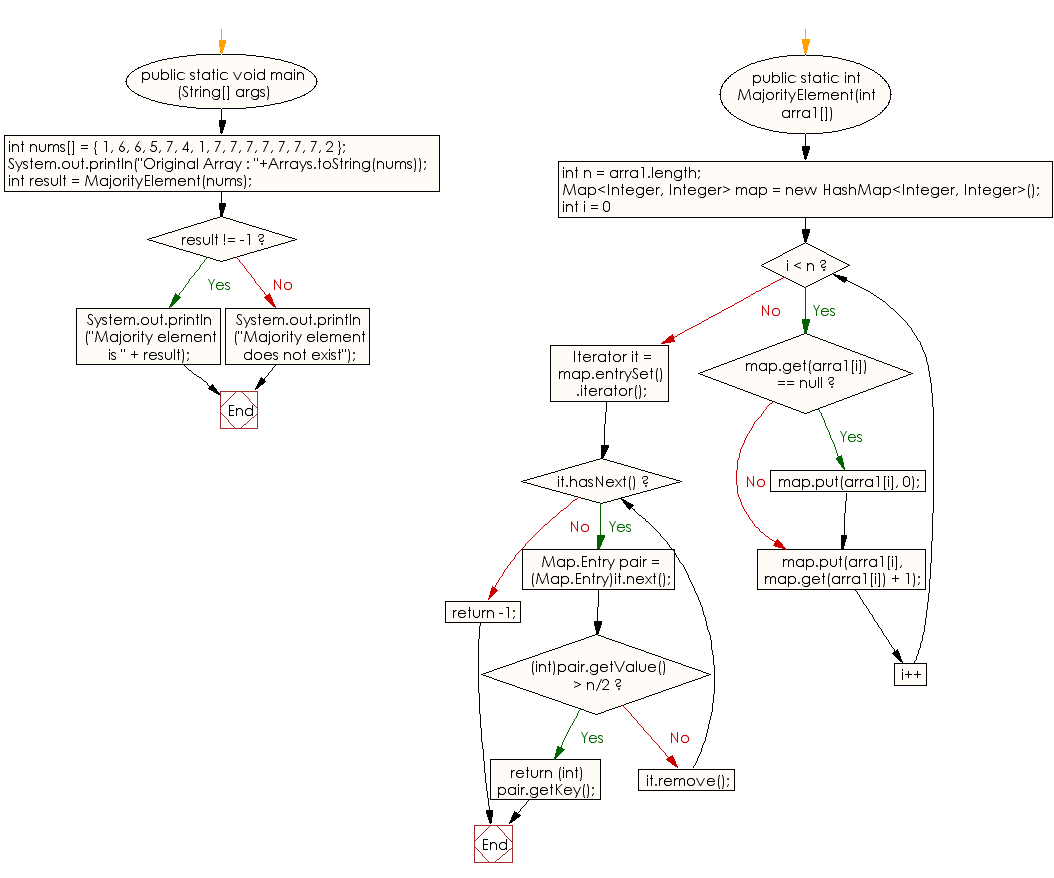
[70, 100]

[110]

**39) Write a Java program to get the majority element from a given array of integers containing duplicates.**

**Majority element: A majority element is an element that appears more than n/2 times where n is the size of the array**.

Flowchart:



import java.util.HashMap;

import java.util.Map;

import java.util.Iterator;

import java.util.Arrays;

public class Program\_39

{

public static void main(String[] args)

{

// Array - test majority element

int nums[] = { 1, 6, 6, 5, 7, 4, 1, 7, 7, 7, 7, 7, 7, 7, 2 };

System.out.println("Original Array : "+Arrays.toString(nums));

int result = MajorityElement(nums);

if (result != -1)

System.out.println("Majority element is " + result);

else

System.out.println("Majority element does not exist");

}

public static int MajorityElement(int arra1[])

{

int n = arra1.length;

// Hash Map

Map<Integer, Integer> map = new HashMap<Integer, Integer>();

// Element's frequency in a map

for (int i = 0; i < n; i++)

{

if (map.get(arra1[i]) == null)

map.put(arra1[i], 0);

map.put(arra1[i], map.get(arra1[i]) + 1);

}

// Return the element if its count is more than n/2

Iterator it = map.entrySet().iterator();

while (it.hasNext())

{

Map.Entry pair = (Map.Entry)it.next();

if ((int)pair.getValue() > n/2)

return (int)pair.getKey();

it.remove();

}

// no majority element

return -1;

}

}

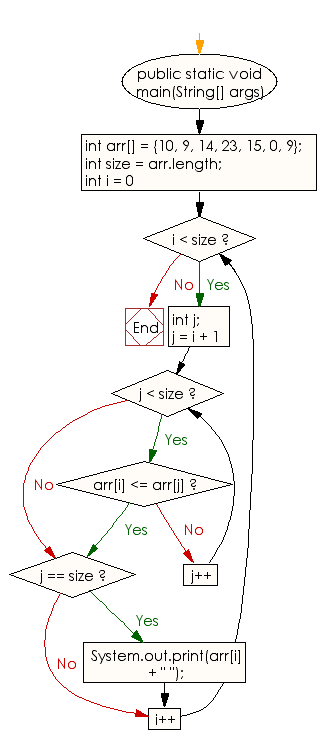
Output:

Original Array : [1, 6, 6, 5, 7, 4, 1, 7, 7, 7, 7, 7, 7, 7, 2]

Majority element is 7

**40) 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.**

Flowchart:



public class Program\_40

{

public static void main(String[] args)

{

int arr[] = {10, 9, 14, 23, 15, 0, 9};

int size = arr.length;

for (int i = 0; i < size; i++)

{

int j;

for (j = i + 1; j < size; j++)

{

if (arr[i] <= arr[j])

break;

}

if (j == size)

System.out.print(arr[i] + " ");

}

}

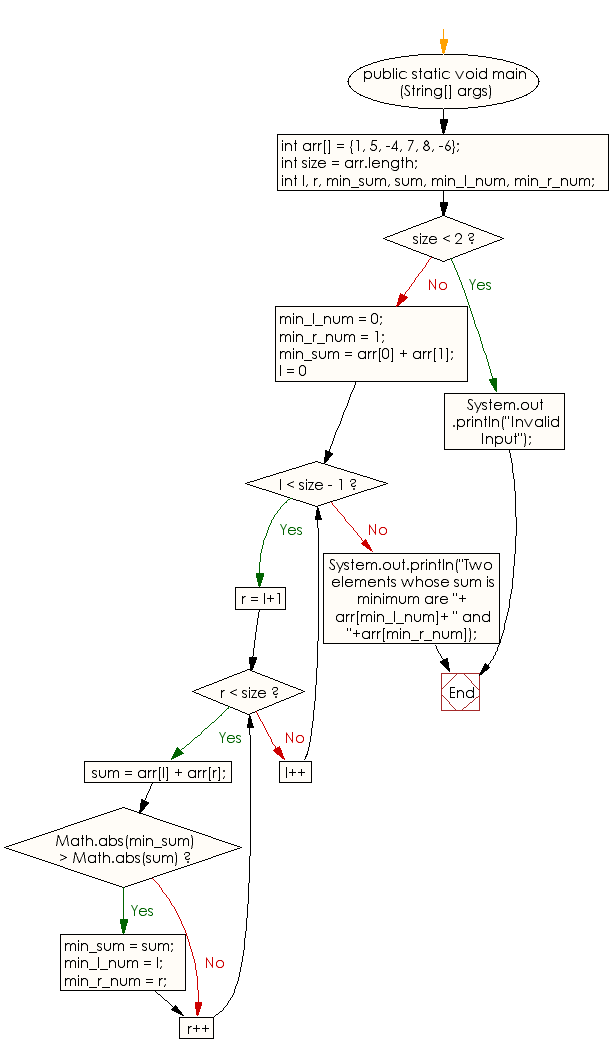
}

Output:

23 15 9

**41) Write a Java program to find the two elements from a given array of positive and negative numbers such that their sum is closest to zero.**

Flowchart:



public class Program\_41

{

public static void main(String[] args)

{

int arr[] = {1, 5, -4, 7, 8, -6};

int size = arr.length;

int l, r, min\_sum, sum, min\_l\_num, min\_r\_num;

if(size < 2)

{

System.out.println("Invalid Input");

return;

}

min\_l\_num = 0;

min\_r\_num = 1;

min\_sum = arr[0] + arr[1];

for(l = 0; l < size - 1; l++)

{

for(r = l+1; r < size; r++)

{

sum = arr[l] + arr[r];

if(Math.abs(min\_sum) > Math.abs(sum))

{

min\_sum = sum;

min\_l\_num = l;

min\_r\_num = r;

}

}

}

System.out.println("Two elements whose sum is minimum are "+

arr[min\_l\_num]+ " and "+arr[min\_r\_num]);

}

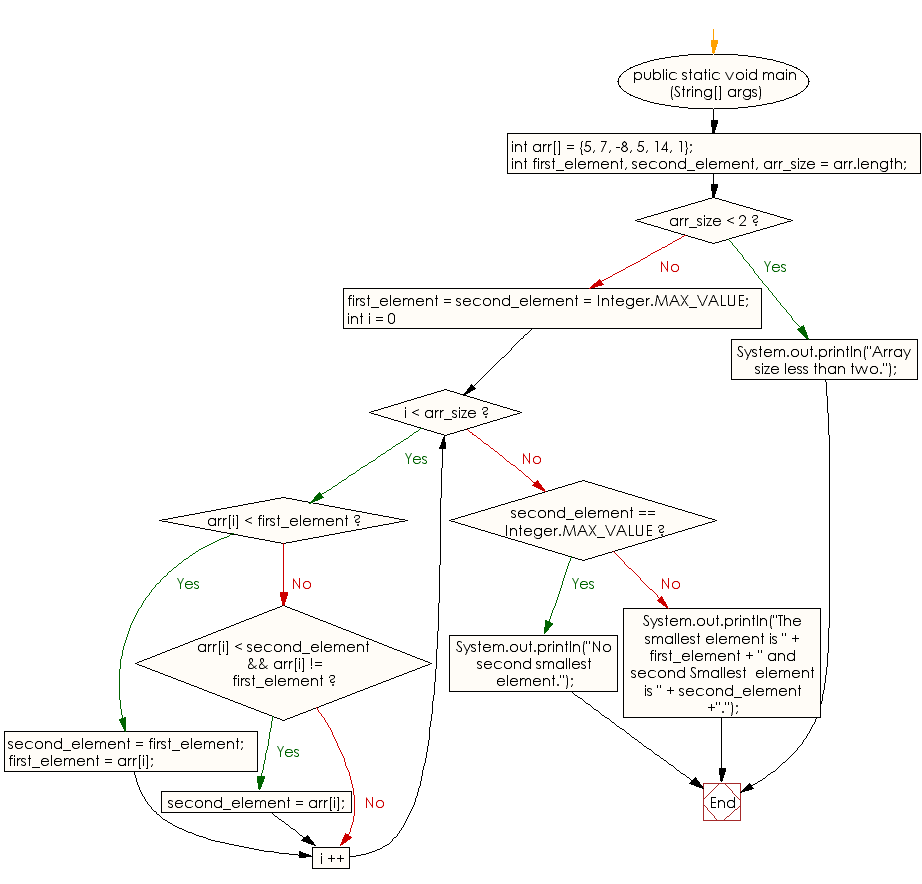
}

Output:

Two elements whose sum is minimum are 5 and -4

**42) Write a Java program to find smallest and second smallest elements of a given array.**

Flowchart



public class Program\_42

{

public static void main(String[] args)

{

int arr[] = {5, 7, -8, 5, 14, 1};

int first\_element, second\_element, arr\_size = arr.length;

/\* Return if the array size less than two \*/

if (arr\_size < 2)

{

System.out.println("Array size less than two.");

return;

}

first\_element = second\_element = Integer.MAX\_VALUE;

for (int i = 0; i < arr\_size ; i ++)

{

/\* Update both first and second if current element is smaller than first. \*/

if (arr[i] < first\_element)

{

second\_element = first\_element;

first\_element = arr[i];

}

/\* Update second if arr[i] is between first and second

elements.\*/

else if (arr[i] < second\_element && arr[i] != first\_element)

second\_element = arr[i];

}

if (second\_element == Integer.MAX\_VALUE)

System.out.println("No second smallest element.");

else

System.out.println("The smallest element is " +

first\_element + " and second Smallest element is " + second\_element +".");

}

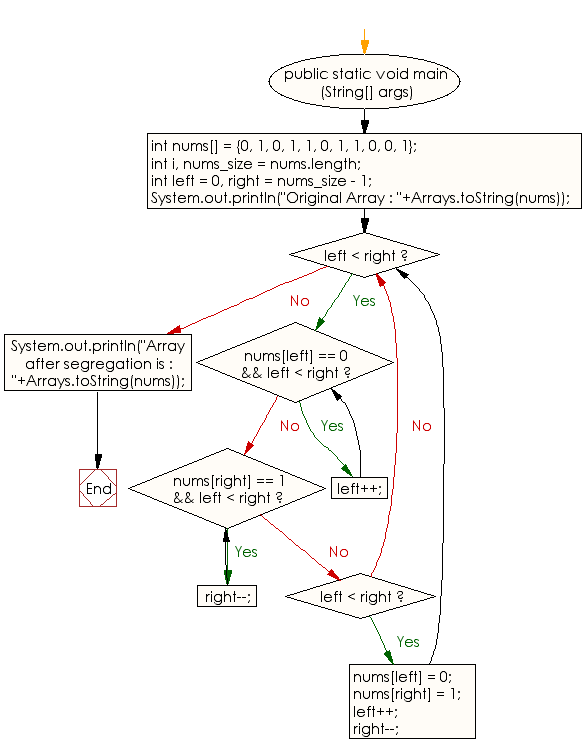
}

Output:

The smallest element is -8 and second Smallest element is 1.

**43) Write a Java program to segregate all 0s on left side and all 1s on right side of a given array of 0s and 1s.**

Flowchart:



import java.util.\*;

import java.lang.\*;

public class Program\_43

{

public static void main(String[] args)

{

int nums[] = {0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1};

int i, nums\_size = nums.length;

int left = 0, right = nums\_size - 1;

System.out.println("Original Array : "+Arrays.toString(nums));

while (left < right)

{

/\* While 0 at left increment left index \*/

while (nums[left] == 0 && left < right)

left++;

/\* While we see 1 at right decrement right index\*/

while (nums[right] == 1 && left < right)

right--;

if (left < right)

{

nums[left] = 0;

nums[right] = 1;

left++;

right--;

}

}

System.out.println("Array after segregation is : "+Arrays.toString(nums));

}

}

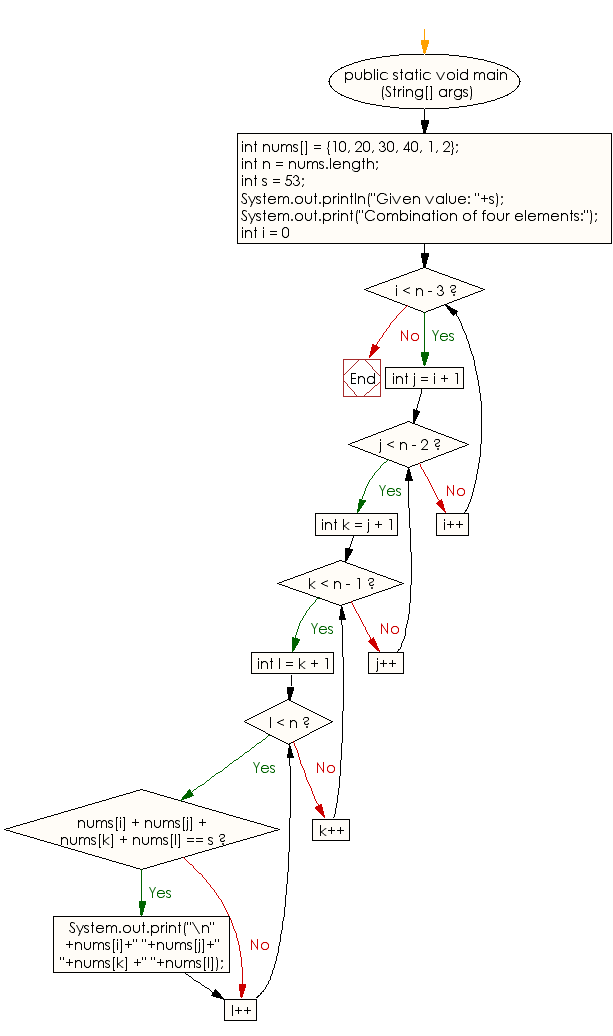
Output:

Original Array : [0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1]

Array after segregation is : [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]

**44) Write a Java program to find all combination of four elements of a given array whose sum is equal to a given value.**

Flowchart:



public class Program\_44

{

public static void main(String[] args)

{

int nums[] = {10, 20, 30, 40, 1, 2};

int n = nums.length;

// given value

int s = 53;

System.out.println("Given value: "+s);

System.out.print("Combination of four elements:");

// Find other three elements after fixing first element

for (int i = 0; i < n - 3; i++)

{

// Find other two elements after fixing second element

for (int j = i + 1; j < n - 2; j++)

{

// Find the fourth element after fixing third element

for (int k = j + 1; k < n - 1; k++)

{

// find the fourth

for (int l = k + 1; l < n; l++)

{

if (nums[i] + nums[j] + nums[k] + nums[l] == s)

System.out.print("\n"+nums[i]+" "+nums[j]+" "+nums[k]

+" "+nums[l]);

}

}

}

}

}

}

Output:

Given value: 53

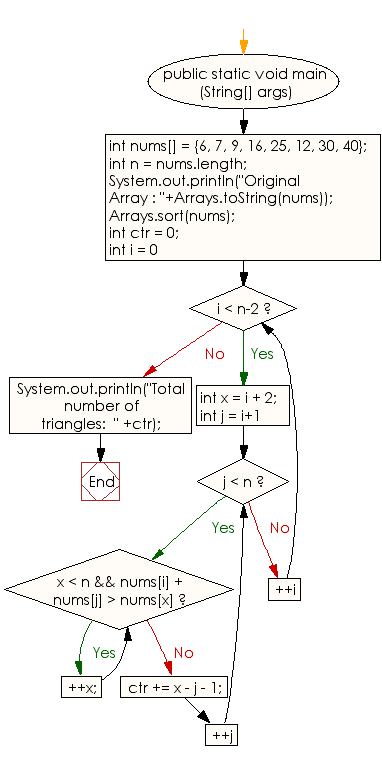
Combination of four elements:

10 40 1 2

20 30 1 2

**45) Write a Java program to count the number of possible triangles from a given unsorted array of positive integers**.

Flowchart:



import java.util.\*;

import java.lang.\*;

public class Program\_46

{

public static void main(String[] args)

{

int nums[] = {6, 7, 9, 16, 25, 12, 30, 40};

int n = nums.length;

System.out.println("Original Array : "+Arrays.toString(nums));

// Sort the array elements in non-decreasing order

Arrays.sort(nums);

// Initialize count of triangles

int ctr = 0;

for (int i = 0; i < n-2; ++i)

{

int x = i + 2;

for (int j = i+1; j < n; ++j)

{

while (x < n && nums[i] + nums[j] > nums[x])

++x;

ctr += x - j - 1;

}

}

System.out.println("Total number of triangles: " +ctr);

}

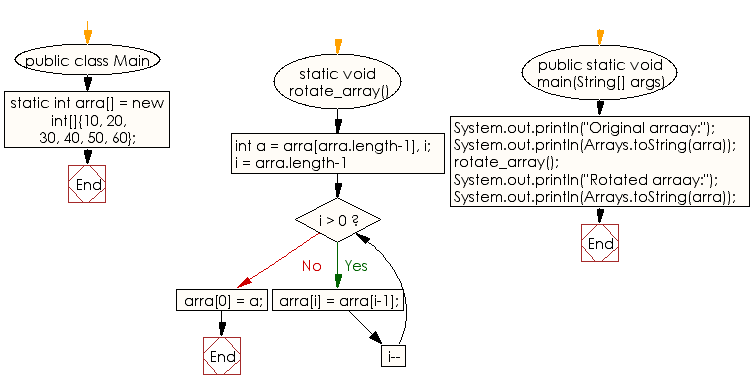
}

Output:

Original Array : [6, 7, 9, 16, 25, 12, 30, 40]

Total number of triangles: 17

**46) Write a Java program to cyclically rotate a given array clockwise by one.**

Flowchart: 

import java.util.Arrays;

public class Program\_47

{

static int arra[] = new int[]{10, 20, 30, 40, 50, 60};

static void rotate\_array()

{

int a = arra[arra.length-1], i;

for (i = arra.length-1; i > 0; i--)

arra[i] = arra[i-1];

arra[0] = a;

}

public static void main(String[] args)

{

System.out.println("Original arraay:");

System.out.println(Arrays.toString(arra));

rotate\_array();

System.out.println("Rotated arraay:");

System.out.println(Arrays.toString(arra));

}

}

Output:

Original arraay:

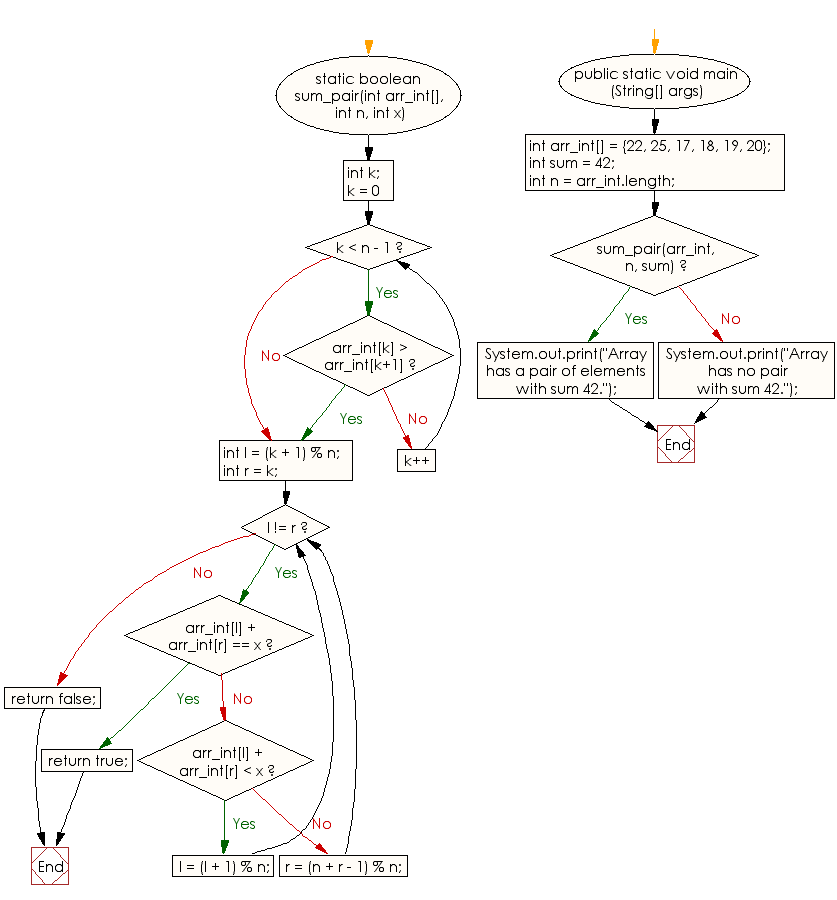
[10, 20, 30, 40, 50, 60]

Rotated arraay:

[60, 10, 20, 30, 40, 50]

**47) Write a Java program to check whether there is a pair with a specified sum of a given sorted and rotated array.**

Flowchart:



public class Program\_48

{

static boolean sum\_pair(int arr\_int[],

int n, int x)

{

int k;

for (k = 0; k < n - 1; k++)

if (arr\_int[k] > arr\_int[k+1])

break;

int l = (k + 1) % n;

int r = k;

while (l != r)

{

if (arr\_int[l] + arr\_int[r] == x)

return true;

if (arr\_int[l] + arr\_int[r] < x)

l = (l + 1) % n;

else

r = (n + r - 1) % n;

}

return false;

}

public static void main(String[] args)

{

int arr\_int[] = {22, 25, 17, 18, 19, 20};

int sum = 42;

int n = arr\_int.length;

if (sum\_pair(arr\_int, n, sum))

System.out.print("Array has a pair of elements with sum 42.");

else

System.out.print("Array has no pair with sum 42.");

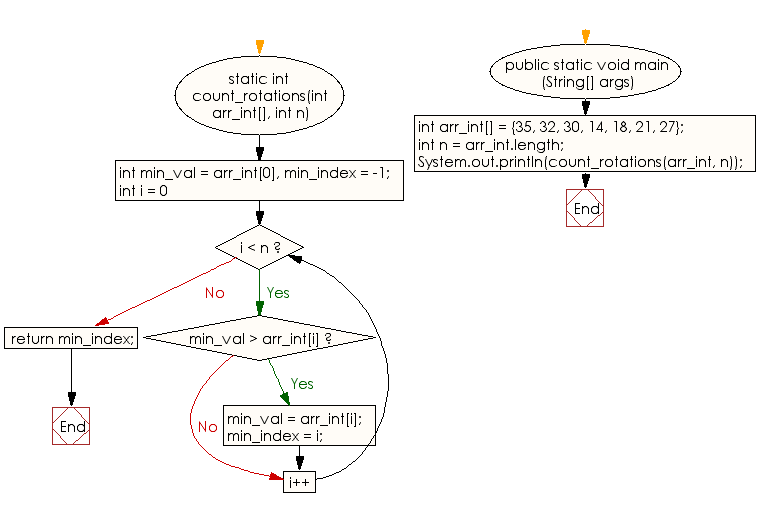
}

}

Output:

Array has a pair of elements with sum 42.

**48) Write a Java program to find the rotation count in a given rotated sorted array of integers**.

Flowchart: 

public class Program\_48

{

static int count\_rotations(int arr\_int[], int n)

{

int min\_val = arr\_int[0], min\_index = -1;

for (int i = 0; i < n; i++)

{

if (min\_val > arr\_int[i])

{

min\_val = arr\_int[i];

min\_index = i;

}

}

return min\_index;

}

public static void main(String[] args)

{

int arr\_int[] = {35, 32, 30, 14, 18, 21, 27};

// int arr\_int[] = {35, 32, 14, 18, 21, 27};

// int arr\_int[] = {35, 14, 18, 21, 27};

int n = arr\_int.length;

System.out.println(count\_rotations(arr\_int, n));

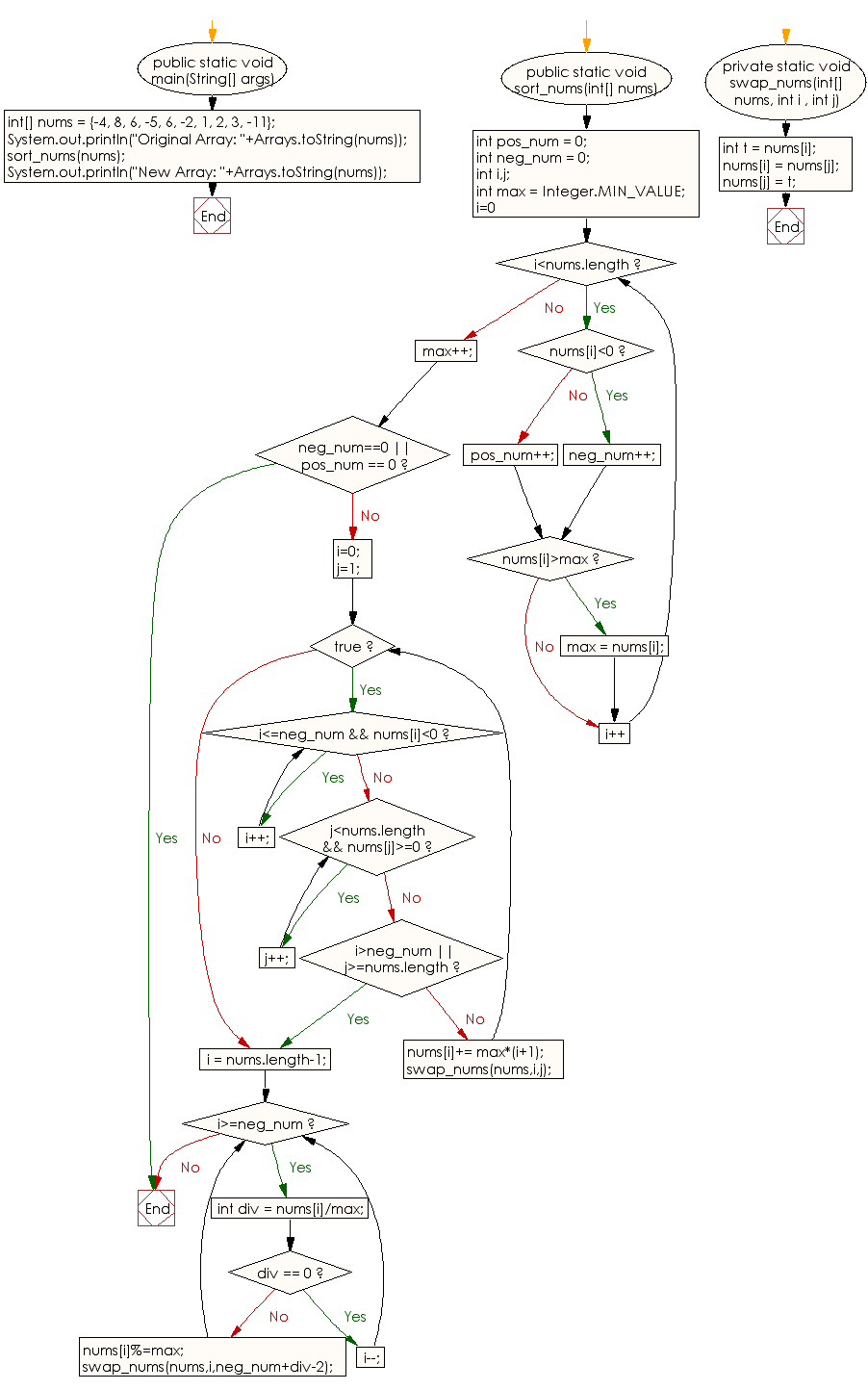
}

}

Output:

3

**49) Write a Java program to arrange the elements of a given array of integers where all negative integers appear before all the positive integers.**

Flowchart: 

import java.util.Arrays;

public class Program\_49

{

public static void main(String[] args)

{

int[] nums = {-4, 8, 6, -5, 6, -2, 1, 2, 3, -11};

System.out.println("Original Array: "+Arrays.toString(nums));

sort\_nums(nums);

System.out.println("New Array: "+Arrays.toString(nums));

}

public static void sort\_nums(int[] nums)

{

int pos\_num = 0;

int neg\_num = 0;

int i,j;

int max = Integer.MIN\_VALUE;

for(i=0; i<nums.length; i++)

{

if(nums[i]<0) neg\_num++;

else pos\_num++;

if(nums[i]>max) max = nums[i];

}

max++;

if(neg\_num==0 || pos\_num == 0) return;

i=0;

j=1;

while(true)

{

while(i<=neg\_num && nums[i]<0) i++;

while(j<nums.length && nums[j]>=0) j++;

if(i>neg\_num || j>=nums.length) break;

nums[i]+= max\*(i+1);

swap\_nums(nums,i,j);

}

i = nums.length-1;

while(i>=neg\_num)

{

int div = nums[i]/max;

if(div == 0) i--;

else{

nums[i]%=max;

swap\_nums(nums,i,neg\_num+div-2);

}

}

}

private static void swap\_nums(int[] nums, int i , int j

{

int t = nums[i];

nums[i] = nums[j];

nums[j] = t;

}

}

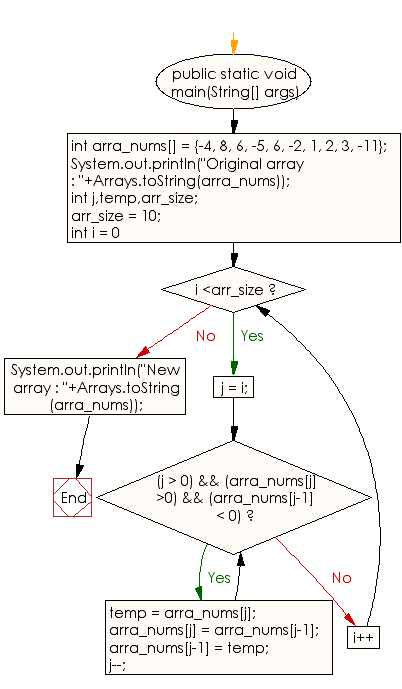
Output:

Original Array: [-4, 8, 6, -5, 6, -2, 1, 2, 3, -11]

New Array: [-4, -5, -2, -11, 6, 6, 1, 2, 8, 3]

**50) Write a Java program to arrange the elements of a given array of integers where all positive integers appear before all the negative integers**

Flowchart:



import java.util.Arrays;

public class Program\_45

{

public static void main(String[] args)

{

int arra\_nums[] = {-4, 8, 6, -5, 6, -2, 1, 2, 3, -11};

System.out.println("Original array : "+Arrays.toString(arra\_nums));

int j,temp,arr\_size;

arr\_size = 8;

for (int i = 0; i <arr\_size; i++)

{

j = i;

//Shift positive numbers left, negative numbers right

while ((j > 0) && (arra\_nums[j] >0) && (arra\_nums[j-1] < 0)){

temp = arra\_nums[j];

arra\_nums[j] = arra\_nums[j-1];

arra\_nums[j-1] = temp;

j--;

}

}

System.out.println("New array : "+Arrays.toString(arra\_nums));

}

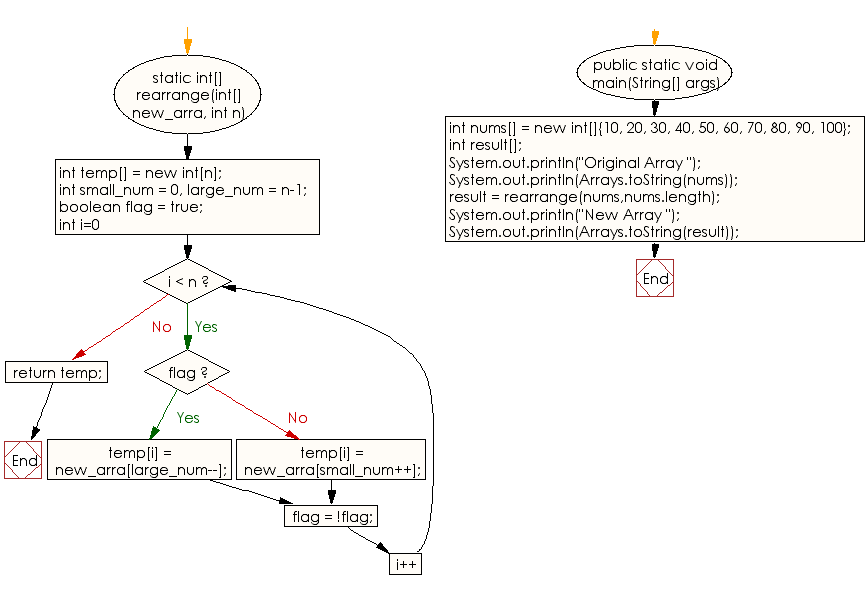
}

Output:

Original array : [-4, 8, 6, -5, 6, -2, 1, 2, 3, -11]

New array : [8, 6, 6, 1, 2, -4, -5, -2, 3, -11]

**51) Write a Java program to sort an array of positive integers of a given array, in the sorted array the value of the first element should be maximum, second value should be minimum value, third should be second maximum, fourth second be second minimum and so on.**

Flowchart: 

import java.util.Arrays;

public class Program\_51

{

static int[] rearrange(int[] new\_arra, int n)

{

int temp[] = new int[n];

int small\_num = 0, large\_num = n-1;

boolean flag = true;

for (int i=0; i < n; i++)

{

if (flag)

temp[i] = new\_arra[large\_num--];

else

temp[i] = new\_arra[small\_num++];

flag = !flag;

}

return temp;

}

public static void main(String[] args)

{

int nums[] = new int[]{10, 20, 30, 40, 50, 60, 70, 80, 90, 100};

int result[];

System.out.println("Original Array ");

System.out.println(Arrays.toString(nums));

result = rearrange(nums,nums.length);

System.out.println("New Array ");

System.out.println(Arrays.toString(result));

}

}

Output:

Original Array

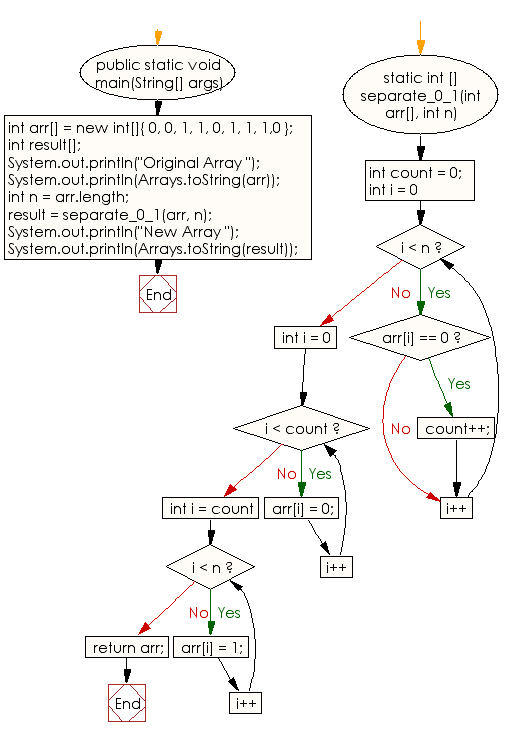
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

New Array

[100, 10, 90, 20, 80, 30, 70, 40, 60, 50]

**52) Write a Java program to separate 0s on left side and 1s on right side of an array of 0s and 1s in random order.**

Flowchart:



import java.util.Arrays;

public class Program\_52

{

public static void main(String[] args)

{

int arr[] = new int[]{ 0, 0, 1, 1, 0, 1, 1, 1,0 };

int result[];

System.out.println("Original Array ");

System.out.println(Arrays.toString(arr));

int n = arr.length;

result = separate\_0\_1(arr, n);

System.out.println("New Array ");

System.out.println(Arrays.toString(result));

}

static int [] separate\_0\_1(int arr[], int n)

{

int count = 0;

for (int i = 0; i < n; i++)

{

if (arr[i] == 0)

count++;

}

for (int i = 0; i < count; i++)

arr[i] = 0;

for (int i = count; i < n; i++)

arr[i] = 1;

return arr;

}

}

Output:

Original Array

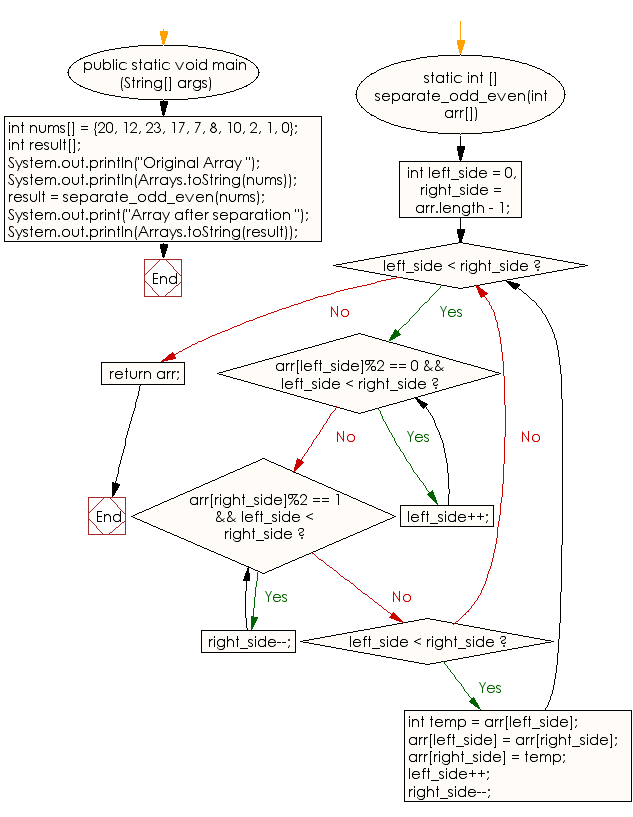
[0, 0, 1, 1, 0, 1, 1, 1, 0]

New Array

[0, 0, 0, 0, 1, 1, 1, 1, 1]

**53) 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.**

Flowchart:



import java.util.Arrays;

public class Program\_53

{

public static void main(String[] args)

{

int nums[] = {20, 12, 23, 17, 7, 8, 10, 2, 1, 0};

int result[];

System.out.println("Original Array ");

System.out.println(Arrays.toString(nums));

result = separate\_odd\_even(nums);

System.out.print("Array after separation ");

System.out.println(Arrays.toString(result));

}

static int [] separate\_odd\_even(int arr[])

{

int left\_side = 0, right\_side = arr.length - 1;

while (left\_side < right\_side)

{

while (arr[left\_side]%2 == 0 && left\_side < right\_side)

left\_side++;

while (arr[right\_side]%2 == 1 && left\_side < right\_side)

right\_side--;

if (left\_side < right\_side)

{

int temp = arr[left\_side];

arr[left\_side] = arr[right\_side];

arr[right\_side] = temp;

left\_side++;

right\_side--;

}

}

return arr;

}

}

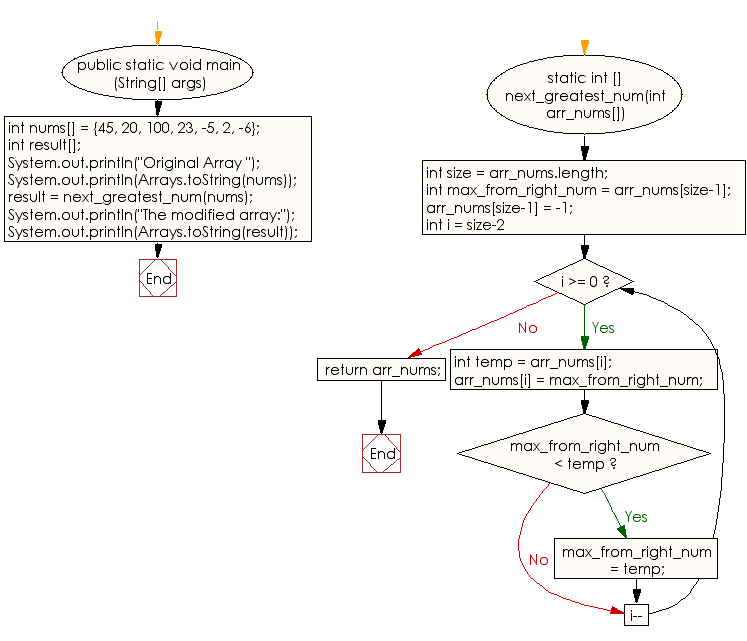
Output:

Original Array

[20, 12, 23, 17, 7, 8, 10, 2, 1, 0]

Array after separation [20, 12, 0, 2, 10, 8, 7, 17, 1, 23]

**54) Write a Java program to replace every element with the next greatest element (from right side) in a given array of integers.**

Flowchart: 

import java.io.\*;

import java.util.Arrays;

public class Program\_54

{

public static void main(String[] args)

{

int nums[] = {45, 20, 100, 23, -5, 2, -6};

int result[];

System.out.println("Original Array ");

System.out.println(Arrays.toString(nums));

result = next\_greatest\_num(nums);

System.out.println("The modified array:");

System.out.println(Arrays.toString(result));

}

static int [] next\_greatest\_num(int arr\_nums[])

{

int size = arr\_nums.length;

int max\_from\_right\_num = arr\_nums[size-1];

arr\_nums[size-1] = -1;

for (int i = size-2; i >= 0; i--)

{

int temp = arr\_nums[i];

arr\_nums[i] = max\_from\_right\_num;

if(max\_from\_right\_num < temp)

max\_from\_right\_num = temp;

}

return arr\_nums;

}

}

Output:

Original Array

[45, 20, 100, 23, -5, 2, -6]

The modified array:

[100, 100, 23, 2, 2, -6, -1]

**55) Java Program to Delete the Specified Integer from an Array**

import java.util.Scanner;

public class Program\_55

{

public static void main(String[] args)

{

int n, x, flag = 1, loc = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:");

n = s.nextInt();

int a[] = new int[n];

System.out.println("Enter all the elements:");

for (int i = 0; i < n; i++)

{

a[i] = s.nextInt();

}

System.out.print("Enter the element you want to delete:");

x = s.nextInt();

for (int i = 0; i < n; i++)

{

if(a[i] == x)

{

flag =1;

loc = i;

break;

}

else

{

flag = 0;

}

}

if(flag == 1)

{

for(int i = loc+1; i < n; i++)

{

a[i-1] = a[i];

}

System.out.print("After Deleting:");

for (int i = 0; i < n-2; i++)

{

System.out.print(a[i]+",");

}

System.out.print(a[n-2]);

}

else

{

System.out.println("Element not found");

}

}

}

Output:

Enter no. of elements you want in array:5

Enter all the elements:

1

2

3

4

5

Enter the element you want to delete:

3

After Deleting:1,2,4,5

**56) Java Program to Split an Array from Specified Position**

import java.util.Scanner;

public class Program\_56

{

public static void main(String[] args)

{

int n, x, flag = 1, loc = 0, k = 0,j = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:");

n = s.nextInt();

int a[] = new int[n];

int b[] = new int[n];

int c[] = new int[n];

System.out.println("Enter all the elements:");

for (int i = 0; i < n; i++)

{

a[i] = s.nextInt();

}

System.out.print("Enter the position from where you want to split:");

loc = s.nextInt();

for(int i = 0; i < loc; i++)

{

b[k] = a[i];

k++;

}

for(int i = loc; i < n; i++)

{

c[j] = a[i];

j++;

}

System.out.print("First array:");

for(int i = 0;i < k; i++)

{

System.out.print(b[i]+" ");

}

System.out.println("");

System.out.print("Second array:");

for(int i = 0; i < j; i++)

{

System.out.print(c[i]+" ");

}

}

}

Output:

Enter no. of elements you want in array:8

Enter all the elements:

1

2

3

4

5

6

7

8

Enter the position from where you want to split:

4

First array:1 2 3 4

Second array:5 6 7 8

**57)** **Java Program to Find Union & Intersection of 2 Arrays**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class Program\_57

{

// Function to find and display the union and intersection

static void displayProgram\_57(int[] arrayOne,int[] arrayTwo){

Set<Integer> obj = new HashSet<>();

int i,j;

for(i=0; i<arrayOne.length; i++)

{

obj.add(arrayOne[i]);

}

for(j=0; j<arrayTwo.length; j++)

{

obj.add(arrayTwo[j]);

}

System.out.println("The union of both the arrays is");

for(Integer I: obj)

{

System.out.print(I + " ");

}

System.out.println();

obj.clear();

System.out.println("The intersection of both the arrays is");

for(i=0; i<arrayOne.length; i++)

{

obj.add(arrayOne[i]);

}

for(j=0; j<arrayTwo.length; j++)

{

if(obj.contains(arrayTwo[j]))

System.out.print(arrayTwo[j] + " ");

}

}

// Function to read the input

public static void main(String[] args)

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int m,n;

System.out.println("Enter the size of the two arrays");

try {

n = Integer.parseInt(br.readLine());

m = Integer.parseInt(br.readLine());

}

catch (IOException e)

{

System.out.println("Invalid input");

return;

}

int[] arrayOne = new int[n];

int[] arrayTwo = new int[m];

System.out.println("Enter the first array elements");

int i,j;

for(i=0; i<arrayOne.length; i++){

try {

arrayOne[i] = Integer.parseInt(br.readLine());

}

catch (IOException e)

{

System.out.println("Invalid array element. Enter it again");

i--;

}

}

System.out.println("Enter the second array elements");

for(i=0; i<arrayTwo.length; i++){

try {

arrayTwo[i] = Integer.parseInt(br.readLine());

}

catch (IOException e)

{

System.out.println("Invalid array element. Enter it again");

i--;

}

}

displayProgram\_57(arrayOne,arrayTwo);

}

}

Output:

Enter the size of the two arrays

5

5

Enter the first array elements

1

23

3

4

5

Enter the second array elements

2

3

7

8

9

The union of both the arrays is

1 2 3 4 5 23 7 8 9

The intersection of both the arrays is

3

**58) Java Program to Accept the Marks of a Student into a 1D Array and find Total Marks and Percentage**

import java.util.Scanner;

public class Program\_58

{

public static void main(String[] args)

{

int n, total = 0, percentage;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of subject:");

n = s.nextInt();

int marks[] = new int[n];

System.out.println("Enter marks out of 100:");

for(int i = 0; i < n; i++)

{

marks[i] = s.nextInt();

total = total + marks[i];

}

percentage = total / n;

System.out.println("Sum:"+total);

System.out.println("Percentage:"+percentage);

}

}

Output:

Enter no. of subject:6

Enter marks out of 100:

82

81

90

45

98

78

Sum:474

Percentage:79

**59) Java Program to Print the kth Element in the Array**

import java.util.Scanner;

public class Program\_59

{

public static void main(String[] args)

{

int n;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:");

n = s.nextInt();

int a[] = new int[n];

System.out.println("Enter all the elements:");

for (int i = 0; i < n; i++)

{

a[i] = s.nextInt();

}

System.out.print("Enter the k th position at which you want to check number:");

int k = s.nextInt();

System.out.println("Number:"+a[k-1]);

}

}

Output:

Enter no. of elements you want in array:7

Enter all the elements:

1

3

4

5

6

7

8

Enter the k th position at which you want to check number:4

Number:5

**60) Java Program to Display Transpose Matrix**

import java.util.Scanner;

public class Program\_60

{

public static void main(String[] args)

{

int i, j;

System.out.println("Enter total rows and columns: ");

Scanner s = new Scanner(System.in);

int row = s.nextInt();

int column = s.nextInt();

int array[][] = new int[row][column];

System.out.println("Enter matrix:");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

array[i][j] = s.nextInt();

System.out.print(" ");

}

}

System.out.println("The above matrix before Transpose is ");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

System.out.print(array[i][j]+" ");

}

System.out.println(" ");

}

System.out.println("The above matrix after Transpose is ");

for(i = 0; i < column; i++)

{

for(j = 0; j < row; j++)

{

System.out.print(array[j][i]+" ");

}

System.out.println(" ");

}

}

}

Output:

Enter total rows and columns:

3 3

Enter matrix:

1

2

3

4

5

6

7

8

9

The above matrix before Transpose is

1 2 3

4 5 6

7 8 9

The above matrix after Transpose is

1 4 7

2 5 8

3 6 9

**61) Java Program to Find the Trace & Normal of a given Matrix**

import java.util.Scanner;

public class Program\_61

{

public static void main(String[] args)

{

int array[][]=new int[10][10];

int i, j;

double sum = 0, square = 0, result = 0;

System.out.println("Enter total rows and columns: ");

Scanner s = new Scanner(System.in);

int row = s.nextInt();

int column = s.nextInt();

System.out.println("Enter matrix:");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

array[i][j] = s.nextInt();

System.out.print(" ");

}

}

System.out.println("The entered Matrix is :");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

System.out.print(array[i][j]+" ");

}

System.out.println(" ");

}

System.out.println("The Trace of the above matrix is ");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

if(i == j)

{

sum = sum + (array[i][j]);

}

}

}

System.out.println(sum);

System.out.println("The Normal of the above matrix is ");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

square = square + (array[i][j])\*(array[i][j]);

}

}

result = Math.sqrt(square);

System.out.println(result);

}

}

Output:

Enter total rows and columns:

3 3

Enter matrix:

1

2

3

4

5

6

7

8

9

The entered Matrix is :

1 2 3

4 5 6

7 8 9

The Trace of the above matrix is

15.0

The Normal of the above matrix is

16.881943016134134

**62) Java Program to Display Upper/Lower Triangle of a Matrix**

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Program\_62

{

static void displayProgram\_62(int[][] matrix)

{

int order = matrix.length;

int i,j;

for(i=0; i<order; i++)

{

for(j=0; j<order;j++)

{

if((i+j) <order)

System.out.print(matrix[i][j] + "\t");

}

System.out.println();

}

for(i=0; i<order; i++)

{

for(j=0; j<order;j++)

{

if((i+j) >=order)

System.out.print(matrix[i][j] + "\t");

}

System.out.println();

}

}

public static void main(String[] args)

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int order;

System.out.println("Enter the order of the matrix");

try{

order = Integer.parseInt(br.readLine());

}

catch(Exception e)

{

System.out.println("An error occurred");

return;

}

int[][] matrix = new int[order][order];

System.out.println("Enter matrix elements");

int i,j;

for(i=0; i<order; i++)

{

for(j=0; j<order; j++)

{

try{

matrix[i][j] = Integer.parseInt(br.readLine());

}

catch(Exception e){

System.out.println("An error occurred");

return;

}

}

}

System.out.println("Tha matrix is");

for(i=0; i<order; i++){

for(j=0; j<order; j++){

System.out.print(matrix[i][j] + "\t");

}

System.out.println();

}

System.out.println("The upper and lower triangle is");

displayProgram\_62(matrix);

}

}

Output:

Enter the order of the matrix

4

Enter matrix elements

1

2

3

4

5

6

7

8

1

2

3

4

1

2

3

4

The matrix is

1 2 3 4

5 6 7 8

1 2 3 4

1 2 3 4

The upper and lower triangle is

1 2 3 4

5 6 7

1 2

1

8

3 4

2 3 4

**63) Java Program to Display Lower Triangular Matrix**

import java.util.Scanner;

public class Program\_63

{

public static void main(String[] args)

{

int a[][] = new int[5][5];

System.out.println("Enter the order of your Matrics ");

System.out.println("Enter the rows:");

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

System.out.println("Enter the columns:");

Scanner s = new Scanner(System.in);

int m = s.nextInt();

if(n == m)

{

System.out.println("Enter your elements:");

for(int i = 0; i < n; i++)

{

for(int j = 0; j < n; j++)

{

Scanner ss = new Scanner(System.in);

a[i][j] = ss.nextInt();

System.out.print(" ");

}

}

System.out.println("You have entered:");

for(int i=0; i<n; i++)

{

for(int j=0;j<n;j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

System.out.println("The Lower Triangular Matrix is:");

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

if(i>=j)

{

System.out.print(a[i][j] +" ");

}

else

{

System.out.print("0"+" ");

}

}

System.out.println("");

}

}

else

{

System.out.println("you have entered improper order");

}

}

}

Output:

Enter the order of the matrix

4

Enter matrix elements

1

2

3

4

5

6

7

8

1

2

3

4

1

2

3

4

The matrix is

1 2 3 4

5 6 7 8

1 2 3 4

1 2 3 4

The upper and lower triangle is

1 2 3 4

5 6 7

1 2

1

8

3 4

2 3 4

**64) Java Program to Interchange any two Rows & Columns in the given Matrix**

import java.util.Scanner;

public class Program\_64

{

public static void main(String[] args)

{

int p, q, n, x , y, temp = 0, k = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of rows in matrix:");

p = s.nextInt();

System.out.print("Enter number of columns in matrix:");

q = s.nextInt();

int a[][] = new int[p][q];

System.out.println("Enter all the elements of matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

a[i][j] = s.nextInt();

}

}

System.out.println("Given Matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

while (true)

{

System.out.println("Enter 1 to interchange rows");

System.out.println("Enter 2 to interchange columns");

System.out.println("Enter 3 to Exit");

n=s.nextInt();

switch (n)

{

case 1:

System.out.println("Enter the row numbers:");

x = s.nextInt();

y = s.nextInt();

for(int i = 0; i < p; i++)

{

temp = a[(x-1)][i];

a[x-1][i] = a[y-1][i];

a[y-1][i] = temp;

}

System.out.println("Matrix after interchanging rows:"+x +" and "+y);

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

break;

case 2:

System.out.println("Enter the column numbers:");

x = s.nextInt();

y = s.nextInt();

for(int i = 0; i < p; i++)

{

temp = a[i][(x-1)];

a[i][x-1] = a[i][(y-1)];

a[i][y-1] = temp;

}

System.out.println("Matrix after interchanging columns:"+x +" and "+y);

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

break;

case 3:

System.exit(0);

}

}

}

}

Output:

Enter number of rows in matrix:3

Enter number of columns in matrix:3

Enter all the elements of matrix:

1

2

3

4

5

6

7

8

9

Given Matrix:

1 2 3

4 5 6

7 8 9

Enter 1 to interchange rows

Enter 2 to interchange columns

Enter 3 to Exit

1

Enter the row numbers:

2

3

Matrix after interchanging rows:2 and 3

1 2 3

7 8 9

4 5 6

Enter 1 to interchange rows

Enter 2 to interchange columns

Enter 3 to Exit

2

Enter the column numbers:

1

3

Matrix after interchanging columns:1 and 3

3 2 1

9 8 7

6 5 4

Enter 1 to interchange rows

Enter 2 to interchange columns

Enter 3 to Exit

**65) Java Program to Find the Frequency of Odd & Even Numbers in the given Matrix**

import java.util.Scanner;

public class Program\_65

{

public static void main(String[] args)

{

int p, q, count1 = 0, count2 = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of rows in matrix:");

p = s.nextInt();

System.out.print("Enter number of columns in matrix:");

q = s.nextInt();

int a[][] = new int[p][q];

System.out.println("Enter all the elements of matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

a[i][j] = s.nextInt();

}

}

System.out.println("Given Matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

if((a[i][j] % 2) == 0)

{

count1++;

}

else

{

count2++;

}

}

}

System.out.println("Even number frequency:"+count1);

System.out.println("Odd number frequency:"+count2);

}

}

Output:

Enter number of rows in matrix:3

Enter number of columns in matrix:3

Enter all the elements of matrix:

1

2

3

4

6

8

1

2

3

Given Matrix:

1 2 3

4 6 8

1 2 3

Even number frequency:5

Odd number frequency:4

**66) Java Program to Accept a Matrix of Order MxN & Interchange the Diagonals**

import java.util.Scanner;

public class Program\_66

{

public static void main(String[] args)

{

int p, q, temp = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of rows in matrix:");

p = s.nextInt();

System.out.print("Enter number of columns in matrix:");

q = s.nextInt();

if (p == q)

{

int a[][] = new int[p][q];

System.out.println("Enter all the elements of matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

a[i][j] = s.nextInt();

}

}

System.out.println("Given Matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

for(int j = 0; j < q; j++)

{

temp = a[j][j];

a[j][j] = a[j][q-1-j];

a[j][q-1-j] = temp;

}

System.out.println("Matrix after interchanging diagonals");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j] + " ");

}

System.out.println("");

}

}

else

{

System.out.println("Rows not equal to column");

}

}

}

Output:

Enter number of rows in matrix:3

Enter number of columns in matrix:3

Enter all the elements of matrix:

1

2

3

4

5

6

1

2

3

Given Matrix:

1 2 3

4 5 6

1 2 3

Matrix after interchanging diagonals

3 2 1

4 5 6

3 2 1

**67) Java Program to Determine if a given Matrix is a Sparse Matrix**

import java.util.Scanner;

public class Program\_67

{

public static void main(String[] args)

{

int i, j, zero = 0, count = 0;

int array[][] = new int[10][10];

System.out.println("Enter total rows and columns: ");

Scanner s = new Scanner(System.in);

int row = s.nextInt();

int column = s.nextInt();

System.out.println("Enter matrix:");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

array[i][j] = s.nextInt();

System.out.print(" ");

}

}

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

if(array[i][j] == 0)

{

zero++;

}

else

{

count++;

}

}

}

if(zero>count)

{

System.out.println("the matrix is sparse matrix");

}

else

{

System.out.println("the matrix is not a sparse matrix");

}

}

}

Output:

Enter total rows and columns:

3

3

Enter matrix:

1

2

3

4

5

6

0

0

0

the matrix is not a sparse matrix

**68) Java Program to Add Two MXN Matrix from User Input**

import java.util.Scanner;

public class Program\_68

{

public static void main(String[] args)

{

int p, q, m, n;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of rows in first matrix:");

p = s.nextInt();

System.out.print("Enter number of columns in first matrix:");

q = s.nextInt();

System.out.print("Enter number of rows in second matrix:");

m = s.nextInt();

System.out.print("Enter number of columns in second matrix:");

n = s.nextInt();

if (p == m && q == n)

{

int a[][] = new int[p][q];

int b[][] = new int[m][n];

int c[][] = new int[m][n];

System.out.println("Enter all the elements of first matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

a[i][j] = s.nextInt();

}

}

System.out.println("Enter all the elements of second matrix:");

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

{

b[i][j] = s.nextInt();

}

}

System.out.println("First Matrix:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < q; j++)

{

System.out.print(a[i][j]+" ");

}

System.out.println("");

}

System.out.println("Second Matrix:");

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

{

System.out.print(b[i][j]+" ");

}

System.out.println("");

}

for (int i = 0; i < p; i++)

{

for (int j = 0; j < n; j++)

{

for (int k = 0; k < q; k++)

{

c[i][j] = a[i][j] + b[i][j];

}

}

}

System.out.println("Matrix after addition:");

for (int i = 0; i < p; i++)

{

for (int j = 0; j < n; j++)

{

System.out.print(c[i][j]+" ");

}

System.out.println("");

}

}

else

{

System.out.println("Addition would not be possible");

}

}

}

Output:

Enter number of rows in first matrix:3

Enter number of columns in first matrix:3

Enter number of rows in second matrix:3

Enter number of columns in second matrix:3

Enter all the elements of first matrix:

1

2

3

4

5

6

7

8

9

Enter all the elements of second matrix:

1

2

3

1

2

3

4

5

6

First Matrix:

1 2 3

4 5 6

7 8 9

Second Matrix:

1 2 3

1 2 3

4 5 6

Matrix after addition:

2 4 6

5 7 9

11 13 15

**69) Java Program to Merge Two Arrays Without Extra Space in Order**

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.Arrays;

public class Program\_69 {

static void inPlaceMerge(int[] arrayOne,int[] arrayTwo){

int i,j,temp;

for(i=arrayTwo.length - 1;i>=0;i--){

temp = arrayOne[arrayOne.length-1];

for(j=arrayOne.length-2; j>=0 && arrayTwo[i] < arrayOne[j];j--){

arrayOne[j+1] = arrayOne[j];

}

if(j!=arrayOne.length-2 || temp > arrayTwo[i]){

arrayOne[j+1] = arrayTwo[i];

arrayTwo[i] = temp;

}

}

}// Function to read input and display the output

public static void main(String[] args) {

// TODO Auto-generated method stub

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int size,size1;

System.out.println("Enter the size of the two arrays");

try {

size = Integer.parseInt(br.readLine());

size1 = Integer.parseInt(br.readLine());

} catch (Exception e) {

System.out.println("Invalid Input");

return;

}

int[] arrayOne = new int[size];

int[] arrayTwo = new int[size1];

System.out.println("Enter the first array elements");

int i;

for (i = 0; i < arrayOne.length; i++) {

try {

arrayOne[i] = Integer.parseInt(br.readLine());

} catch (Exception e) {

System.out.println("An error occurred");

}

}

System.out.println("Enter the second array elements");

for (i = 0; i < arrayTwo.length; i++) {

try {

arrayTwo[i] = Integer.parseInt(br.readLine());

} catch (Exception e) {

System.out.println("An error occurred");

}

}

Arrays.sort(arrayOne);

Arrays.sort(arrayTwo);

System.out.println("Initially arrays are");

System.out.println("Array One");

System.out.println(Arrays.toString(arrayOne));

System.out.println("\nArray Two");

System.out.println(Arrays.toString(arrayTwo));

inPlaceMerge(arrayOne,arrayTwo);

System.out.println("\nArrays after merging are");

System.out.println("\nArray One");

System.out.println(Arrays.toString(arrayOne));

System.out.println("\nArray Two");

System.out.println(Arrays.toString(arrayTwo));

}

}

Output:

Enter the size of the two arrays

5

5

Enter the first array elements

2

7

1

8

3

Enter the second array elements

-2

-1

9

0

3

Initially arrays are

Array One

[1, 2, 3, 7, 8]

Array Two

[-2, -1, 0, 3, 9]

Arrays after merging are

Array One

[-2, -1, 0, 1, 2]

Array Two

[3, 3, 7, 8, 9]

**70) Java Program to Find Median of Elements where Elements are Stored in 2 Different Arrays**

import java.util.Random;

public class Program\_70

{

static int N = 10, M = 5;

static int[] sequence1 = new int[N];

static int[] sequence2 = new int[M];

static int[] sequence = new int[N+M];

public static void sort()

{

int i, j, temp;

for (i = 1; i < N+M; i++)

{

j = i;

temp = sequence[i];

while (j > 0 && temp < sequence[j - 1])

{

sequence[j] = sequence[j - 1];

j = j - 1;

}

sequence[j] = temp;

}

}

public static void main(String[] args)

{

Random random = new Random();

for(int i=0; i<N; i++)

sequence1[i] = Math.abs(random.nextInt(100));

for(int i=0; i<M; i++)

sequence2[i] = Math.abs(random.nextInt(100));

for(int i=0; i<N; i++)

System.out.print(sequence1[i] + " ");

System.out.println();

for(int i=0; i<M; i++)

System.out.print(sequence2[i] + " ");

System.out.println();

int j=0;

for(int i=0; i<N+M; i++)

{

if(i >= N && j < M)

sequence[i] = sequence2[j++];

else

sequence[i] = sequence1[i];

}

sort();

if(N+M % 2 == 0)

System.out.println("The Median is : " + (sequence[(N+M)/2-1]+sequence[(N+M)/2])/2);

else

System.out.println("The Median is : " + sequence[(N+M)/2]);

}

}

Output:

73 6 58 14 8 3 1 12 79 31

12 95 82 70 10

The Median is : 14