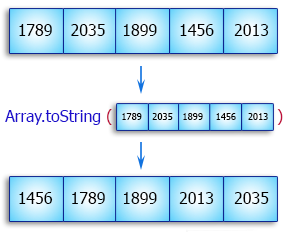
1. **Write a Java program to sort a numeric array and a string array.**

**Pictorial Presentation:**



**Solution:**

import java.util.Arrays;

public class Exercise1 {

public static void main(String[] args){

int[] my\_array1 = {

1789, 2035, 1899, 1456, 2013,

1458, 2458, 1254, 1472, 2365,

1456, 2165, 1457, 2456};

String[] my\_array2 = {

"Java",

"Python",

"PHP",

"C#",

"C Programming",

"C++"

};

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

Arrays.sort(my\_array1);

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

System.out.println("Original string array : "+Arrays.toString(my\_array2));

Arrays.sort(my\_array2);

System.out.println("Sorted string array : "+Arrays.toString(my\_array2));

}

}

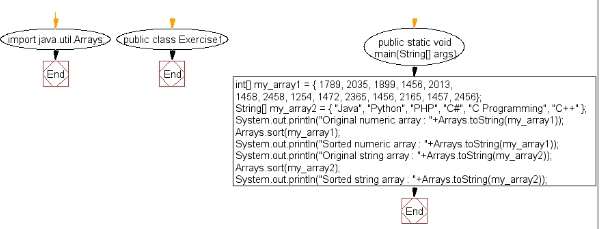
**Sample Output**

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

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

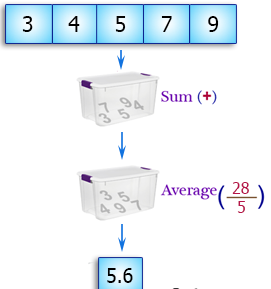
Original string array : [Java, Python, PHP, C#, C Programming, C++]

Sorted string array : [C Programming, C#, C++, Java, PHP, Python



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

**Pictorial Presentation:**



**Solution**:

public class Exercise4 {

public static void main(String[] args) {

int[] numbers = new int[]{20, 30, 25, 35, -16, 60, -100};

//calculate sum of all array elements

int sum = 0;

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

sum = sum + numbers[i];

//calculate average value

double average = sum / numbers.length;

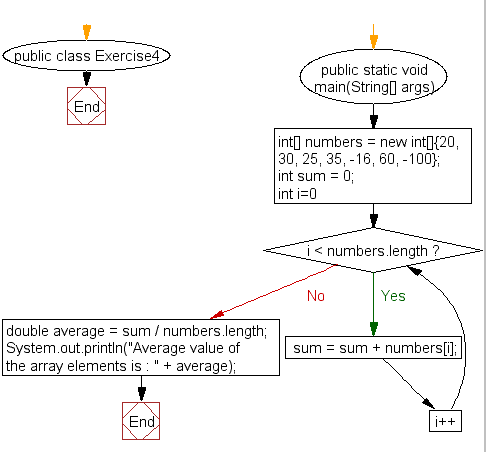
System.out.println("Average value of the array elements is : " + average);

}

}

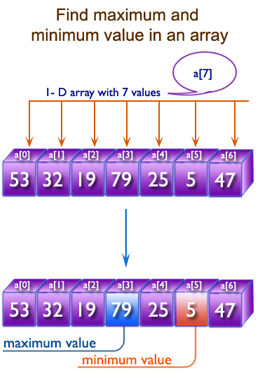
**Sample Output:**

Average value of the array elements is : 7.0



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

**Pictorial Presentation:**



**Solution**:

import java.util.Arrays;

public class Exercise10 {

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);

}

}

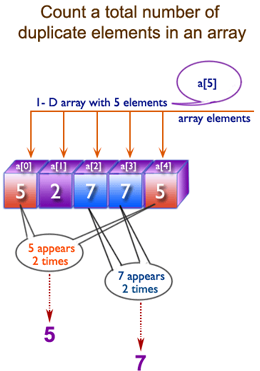
**Sample Output:**

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

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



**Solution**:

import java.util.Arrays;

public class Exercise12 {

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]);

}

}

}

}

}

**Sample Output:**

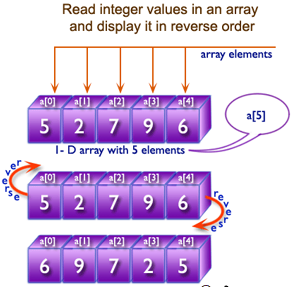
Duplicate Element : 2

Duplicate Element : 5

Duplicate Element : 6

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

**Pictorial Presentation:**



**Solution**:

import java.util.Arrays;

public class Exercise11 {

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));

}

}

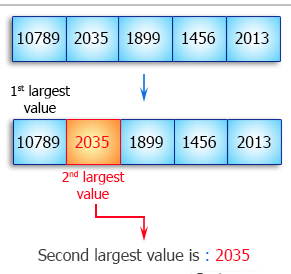
**Sample Output:**

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]

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

**Pictorial Presentation:**



**Solution**:

import java.util.Arrays;

public class Main {

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]);

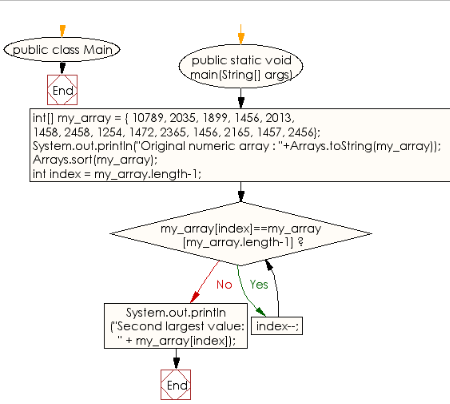
}

}

**Sample Output:**

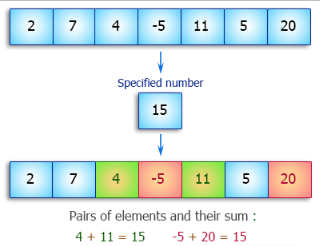
Original numeric array : [10789, 2035, 1899, 1456, 2013, 1458, 2458, 1254, 1472, 2365, 1456, 2165, 1457, 2456]

Second largest value: 2458



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

**Pictorial Presentation:**



**Solution**:

public class Exercise22 {

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);

}

}

**Sample Output:**

Pairs of elements and their sum :

4 + 11 = 15

-5 + 20 = 15

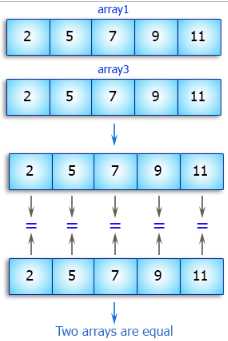
Pairs of elements and their sum :

14 + 16 = 30

-15 + 45 = 30

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

**Pictorial Presentation:**



**Solution**:

public class Exercise23 {

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);

}

}

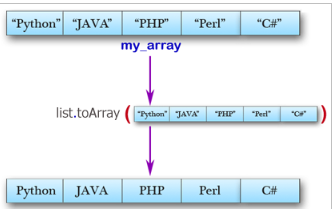
**Sample Output:**

Two arrays are not equal.

Two arrays are equal.

1. **Write a Java program to convert an ArrayList to an array.**

**Pictorial Presentation:**



**Solution**:

import java.util.ArrayList;

import java.util.Arrays;

public class Exercise21 {

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);

}

}

}

**Sample Output:**

Python

Java

PHP

C#

C++

Perl

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

**Pictorial Presentation:**



**Solution**:

import java.util.\*;

public class Exercise24 {

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");

}

}

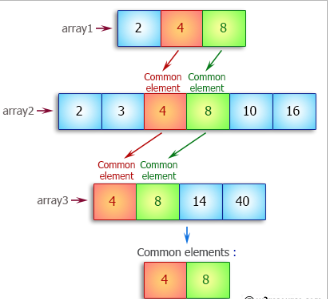
Sample Data: 1,2,3,4,6,7

**Sample Output:**

5

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

**Pictorial Presentation:**



**Solution**:

import java.util.\*;

public class Exercise25 {

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);

}

}

Sample Data: array1 = 2, 4, 8

array2 = 2, 3, 4, 8, 10, 16

array3 = 4, 8, 14, 40

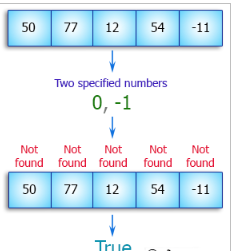
**Sample Output:**

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

[4, 8]

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

**Pictorial Presentation:**



**Solution**:

import java.util.\*;

import java.io.\*;

public class Exercise30 {

public static void main(String[] args)

{

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

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

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

}

public static boolean test(int[] numbers) {

for (int number : numbers) {

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

return false;

}

}

return true;

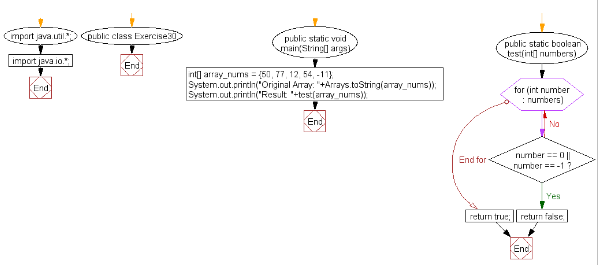
}

}

**Sample Output:**

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

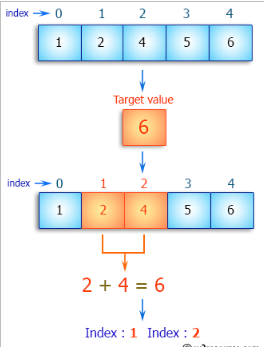
Result: true



1. **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.**

**Pictorial Presentation:**



**Solution**:

import java.util.\*;

public class Exercise35

{

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 + " ");

}

}

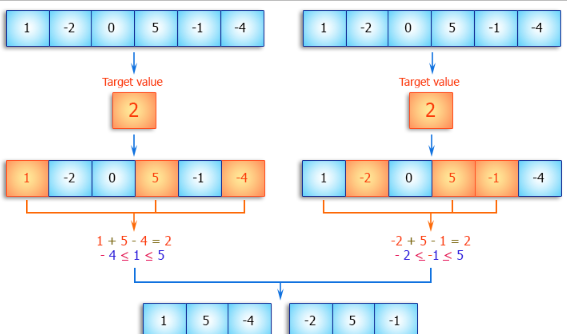
**Sample Output:**

Index: 1 Index: 2

1. **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.**

**Pictorial Presentation:**



**Solution**:

import java.util.ArrayList;

import java.util.List;

public class Exercise36 {

public static void main(String[] args) {

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

int target = 2;

Exercise36 r = new Exercise36();

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;

}

}

**Sample Output:**

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

1. **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.**

**Sample Solution**:

import java.util.\*;

import java.lang.\*;

public class Main

{

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]);

}

}

**Sample Output:**

Two elements whose sum is minimum are 5 and -4