Notes:

I)For all exercises, use the following class to read from keyboard:

**import java.util.Scanner;**

**public class KeyBoard {**

**private static Scanner *s* = new Scanner(System.*in*);**

**public static String readString() {**

**return *s*.nextLine();**

**}**

**public static char readChar() {**

**String str = *s*.nextLine();**

**return str.charAt(0);**

**}**

**public static int readInt() {**

**String str = *s*.nextLine();**

**return Integer.*parseInt*(str);**

**}**

**public static float readFloat() {**

**String str = *s*.nextLine();**

**return Float.*parseFloat*(str);**

**}**

**}**

Example:

System.out.println(“Enter a number”);

int num = KeyBoard.readInt();

Just create a Keyboard.java file and paste the code into it.

II) As much as possible do the user prompting and accepting from keyboard in the main method and not in the called method.

III) Remember that System.out**.print** method prints without adding new line at the end (as compared to System.out**.println**)

1.Write a Java Method to Compute the Sum of Array Elements. Accept array as parameter and return sum

Input: [ 2, 4, 6, 7, 9]

Output: 28

2.Write a Java Method to Find the Largest Element in Array. Accept array as parameter and return largest

Input: [ 7, 2, 5, 1, 4]

Output: 7

3.Write a Java program to reverse an array of integer values. Main only

4.Write a Java Method to rotate an array.Accept array and d as parameter and return a new rotated array

Input: arr[] = {1, 2, 3, 4, 5, 6, 7}, d = 2

Output: 3 4 5 6 7 1 2

Explanation: d=2 so 2 elements are rotated to the end of the array. So, 1 2 is rotated back

So, Final result: 3, 4, 5, 6, 7, 1, 2

5.Write a method to Remove Duplicate Elements From an Array.Accept array as parameter and return a new deduplicated array

Input: [ 1, 2, 2, 3, 3, 3, 4, 5 ]

Output: [ 1, 2, 3, 4, 5 ]

6.Write a method to Remove All Occurrences of an Element in an Array. Accept array as parameter and return a new array

Input: array = [ 1, 2, 1, 3, 5, 1 ] , key = 1

Output: [2, 3, 5]

Explanation: all the occurrences of element 1 is removed from the array So, array becomes from

[ 1, 2, 1, 3, 5, 1 ] to

Final result: [2, 3, 5]

7.Write Java Method to Find the Tranpose of Matrix. Accept 2D array as parameter and return a new 2D array

Input:

[ [ 1, 2, 3 ]

[ 4, 5, 6 ]

[ 7, 8, 9 ] ]

Output:

[ [ 1, 4, 7]

[ 2, 5, 8]

[ 3, 6, 9] ]

8. Java Program for Bubble Sort. Write a method ‘sort’ which accepts an int[] as parameter

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. Bubble Sort in Java is not the best method to sort an array but is one of the most basic implementations for one to learn. In this article, we will learn how to write a program for Bubble Sort in Java.

Algorithm for Bubble Sort in Java

The following is the algorithm to sort array in increasing order using bubble sort in Java:

1.Start

2.Initiate two values n as size of array ,also i and j to traverse array.

3.Put i=0 and j=1.

4.While traversing if array[i] > array[j] swap both the numbers.

5.Increment the value i and j then goto Step 3.

6.If the value of i > n-1 and j > n and n>1 then

n=n-1

goto Step 2

7.Exit

Example of Bubble Sort

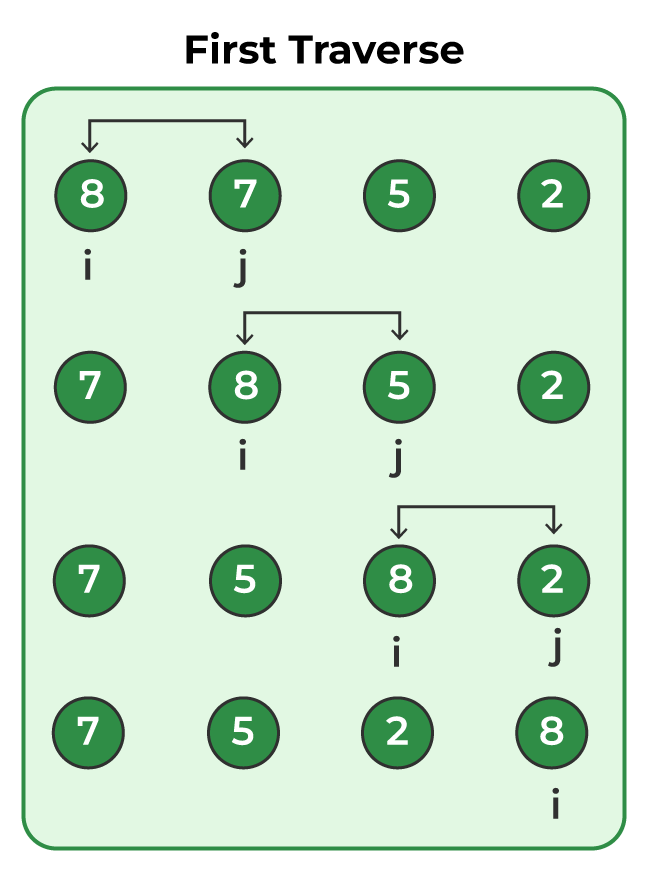
Step1:

Given, array: [ 8,7,5,2] , traverse from pos=1 till pos=4 and swap the number with the previous element if it is less than the previous element.

**Example of Bubble Sort**

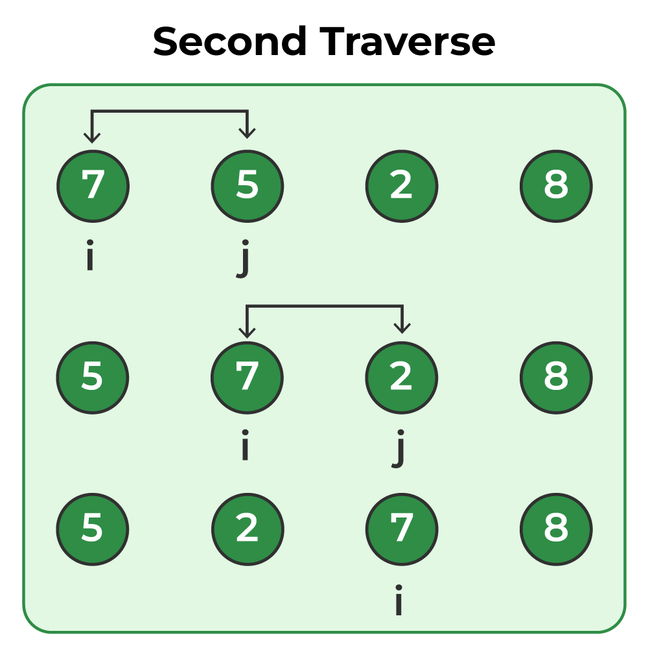
**Step1:**

*Given, array: [ 8,7,5,2] , traverse from pos=1 till pos=4 and swap the number with the previous element if it is less than the previous element.*



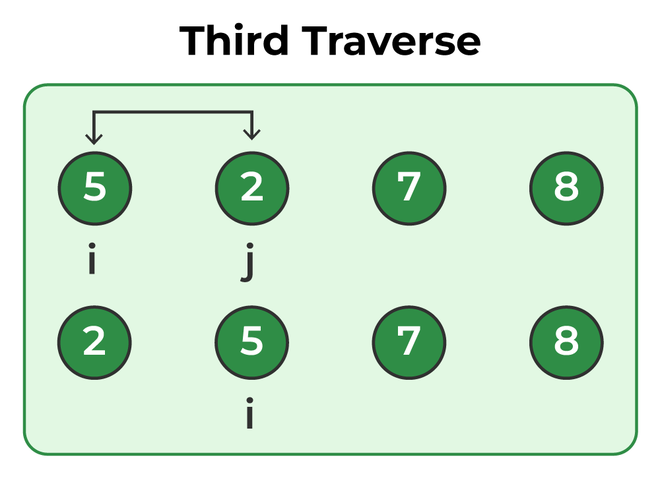
**Step 2:**

*Repeat the previous Step for pos=1 to pos=3*



**Step 3:**

*Follow Step 1 for pos=1 to pos=2*



**Step 4:**

*As now pos=1 to pos=1 means the array is now fully sorted*

