Java Conditional Statement Exercises

======================================

1. Write a Java program to get a number from the user and print whether it is positive or negative

Test Data

Input number: 35

Expected Output :

Number is positive

2. Write a Java program to solve quadratic equations (use if, else if and else).

Test Data

Input a: 1

Input b: 5

Input c: 1

Expected Output :

The roots are -0.20871215252208009 and -4.7912878474779195

3. Take three numbers from the user and print the greatest number.

Test Data

Input the 1st number: 25

Input the 2nd number: 78

Input the 3rd number: 87

Expected Output :

The greatest: 87

4. Write a Java program that reads a floating-point number and prints "zero" if the number is zero. Otherwise, print "positive" or "negative". Add "small" if the absolute value of the number is less than 1, or "large" if it exceeds 1,000,000.

Test Data

Input a number: 25

Expected Output :

Input value: 25

Positive number

5. Write a Java program that keeps a number from the user and generates an integer between 1 and 7 and displays the name of the weekday.

Test Data

Input number: 3

Expected Output :

Wednesday

6. Write a Java program that reads in two floating-point numbers and tests whether they are the same up to three decimal places.

Test Data

Input floating-point number: 1256

Input floating-point another number: 3254

Expected Output :

They are different

7. Write a Java program to find the number of days in a month.

Test Data

Input a month number: 2

Input a year: 2016

Expected Output :

February 2016 has 29 days

8. Write a Java program that takes the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.

Test Data

Input an alphabet: p

Expected Output :

Input letter is Consonant

9. Write a Java program that takes a year from user and print whether that year is a leap year or not.

Test Data

Input the year: 2016

Expected Output :

2016 is a leap year

10. Write a program in Java to display the first 10 natural numbers.

Expected Output :

The first 10 natural numbers are:

1

2

3

4

5

6

7

8

9

10

11. Write a program in Java to display n terms of natural numbers and their sum.

Test Data

Input the number: 2

Expected Output :

Input number:

2

The first n natural numbers are :

2

1

2

The Sum of Natural Number upto n terms :

23

12. Write a program in Java to input 5 numbers from keyboard and find their sum and average.

Test Data

Input the 5 numbers : 1 2 3 4 5

Expected Output :

Input the 5 numbers :

1

2

3

4

5

The sum of 5 no is : 15

The Average is : 3.0

13. Write a program in Java to display the cube of the number upto given an integer.

Test Data

Input number of terms : 4

Expected Output :

Number is : 1 and cube of 1 is : 1

Number is : 2 and cube of 2 is : 8

Number is : 3 and cube of 3 is : 27

Number is : 4 and cube of 4 is : 64

14. Write a program in Java to display the multiplication table of a given integer.

Test Data

Input the number (Table to be calculated) : Input number of terms : 5

Expected Output :

5 X 0 = 0

5 X 1 = 5

5 X 2 = 10

5 X 3 = 15

5 X 4 = 20

5 X 5 = 25

15. Write a program in Java to display the n terms of odd natural number and their sum.

Test Data

Input number of terms is: 5

Expected Output :

The odd numbers are :

1

3

5

7

9

The Sum of odd Natural Number upto 5 terms is: 25

16. Write a program in Java to display the pattern like right angle triangle with a number.

Test Data

Input number of rows : 10

Expected Output :

1

12

123

1234

12345

123456

1234567

12345678

123456789

12345678910

17. Write a program in Java to make such a pattern like right angle triangle with a number which will repeat a number in a row.The pattern is as follows :

1

22

333

4444

18. Write a program in Java to make such a pattern like right angle triangle with number increased by 1.The pattern like

1

2 3

4 5 6

7 8 9 10

19. Write a program in Java to make such a pattern like a pyramid with a number which will repeat the number in the same row

1

2 2

3 3 3

4 4 4 4

20. Write a program in Java to print the Floyd's Triangle.

Test Data

Input number of rows : 5

Expected Output :

1

01

101

0101

10101

21. Write a program in Java to display the pattern like a diamond

Test Data

Input number of rows (half of the diamond) : 7

Expected Output :

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

22. Write a Java program to display Pascal's triangle.

Test Data

Input number of rows: 5

Expected Output :

Input number of rows: 5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

23. Write a java program to generate a following \*'s triangle.

Test Data

Input the number: 6

Expected Output :

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

24. Write a java program to generate a following @'s triangle.

Test Data

Input the number: 6

Expected Output :

@

@@

@@@

@@@@

@@@@@

@@@@@@

25. Write a Java program to display the number rhombus structure.

Test Data

Input the number: 7

Expected Output :

1

212

32123

4321234

543212345

65432123456

7654321234567

65432123456

543212345

4321234

32123

212

26. Write a Java program to display the following character rhombus structure.

Test Data

Input the number: 7

Expected Output :

A

ABA

ABCBA

ABCDCBA

ABCDEDCBA

ABCDEFEDCBA

ABCDEFGFEDCBA

ABCDEFEDCBA

ABCDEDCBA

ABCDCBA

ABCBA

ABA

A

27. Write a Java program that reads an integer and check whether it is negative, zero, or positive.

Test Data

Input a number: 7

Expected Output :

Number is positive

28. Write a Java program that reads a floating-point number. If the number is zero it prints "zero", otherwise, print "positive" or "negative". Add "small" if the absolute value of the number is less than 1, or "large" if it exceeds 1,000,000.

Test Data

Input a number: -2534

Expected Output :

Negative

29. Write a Java program that reads an positive integer and count the number of digits the number (less than ten billion) has

Test Data

Input an integer number less than ten billion: 125463

Expected Output :

Number of digits in the number: 6

30. Write a Java program that accepts three numbers and prints "All numbers are equal" if all three numbers are equal, "All numbers are different" if all three numbers are different and "Neither all are equal or different" otherwise.

Test Data

Input first number: 2564

Input second number: 3526

Input third number: 2456

Expected Output :

All numbers are different

31. Write a program that accepts three numbers from the user and prints "increasing" if the numbers are in increasing order, "decreasing" if the numbers are in decreasing order, and "Neither increasing or decreasing order" otherwise

Test Data

Input first number: 1524

Input second number: 2345

Input third number: 3321

Expected Output :

Increasing order

Java Array Exercises

==========================

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

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

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

Expected Output :

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

- - - - - - - - - -

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

20. Write a Java program to convert an array to ArrayList.

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

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

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

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

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

26. 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.

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

28. 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.

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

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

31. 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.

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

33. 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.

34. 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.

35. 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.

36. 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.

37. 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]

]

38. Write a Java program to get the majority element from an 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.

39. 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.

40. 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.

41. Write a Java program to find smallest and second smallest elements of a given array.

42. 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.

43. Write a Java program to find all combination of four elements of an given array whose sum is equal to a given value.

44. Write a Java program to count the number of possible triangles from an given unsorted array of positive integers.

45. Write a Java program to cyclically rotate a given array clockwise by one.

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

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

48. Write a Java program to arrange the elements of an given array of integers where all negative integers appear before all the positive integers.

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

50. Write a Java program to sort an array of positive integers of an 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.

51. 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.

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

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

Java Method Exercises

=========================

1. Write a Java method to find the smallest number among three numbers.

Test Data:

Input the first number: 25

Input the Second number: 37

Input the third number: 29

Expected Output:

The smallest value is 25.0

2. Write a Java method to compute the average of three numbers.

Test Data:

Input the first number: 25

Input the second number: 45

Input the third number: 65

Expected Output:

The average value is 45.0

3. Write a Java method to display the middle character of a string.

Note: a) If the length of the string is odd there will be two middle characters.

b) If the length of the string is even there will be one middle character.

Test Data:

Input a string: 350

Expected Output:

The middle character in the string: 5

4. Write a Java method to count all vowels in a string.

Test Data:

Input the string: w3resource

Expected Output:

Number of Vowels in the string: 4

5. Write a Java method to count all words in a string.

Test Data:

Input the string: The quick brown fox jumps over the lazy dog.

Expected Output:

Number of words in the string: 9

6. Write a Java method to compute the sum of the digits in an integer.

Test Data:

Input an integer: 25

Expected Output:

The sum is 7

7. Write a Java method to display the first 50 pentagonal numbers.

Note: A pentagonal number is a figurate number that extends the concept of triangular and square numbers to the pentagon, but, unlike the first two, the patterns involved in the construction of pentagonal numbers are not rotationally symmetrical.

Expected Output:

1 5 12 22 35 51 70 92 117 145

176 210 247 287 330 376 425 477 532 590

651 715 782 852 925 1001 1080 1162 1247 1335

1426 1520 1617 1717 1820 1926 2035 2147 2262 2380

2501 2625 2752 2882 3015 3151 3290 3432 3577 3725

8. Write a Java method to compute the future investment value at a given interest rate for a specified number of years.

Sample data (Monthly compounded) and Output:

Input the investment amount: 1000

Input the rate of interest: 10

Input number of years: 5

Expected Output:

Years FutureValue

1 1104.71

2 1220.39

3 1348.18

4 1489.35

5 1645.31

9. Write a Java method to print characters between two characters (i.e. A to P ).

Note: Prints 20 characters per line

Expected Output:

( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 : ;

< = > ? @ A B C D E F G H I J K L M N O

P Q R S T U V W X Y Z [ \ ] ^ \_ ` a b c

d e f g h i j k l m n o p q r s t u v w

x y z

10. Write a Java method to check whether a year (integer) entered by the user is a leap year or not.

Expected Output:

Input a year: 2017

false

11. Write a Java method to check whether a string is a valid password.

Password rules:

A password must have at least ten characters.

A password consists of only letters and digits.

A password must contain at least two digits.

Expected Output:

1. A password must have at least eight characters.

2. A password consists of only letters and digits.

3. A password must contain at least two digits

Input a password (You are agreeing to the above Terms and Conditions.): abcd1234

Password is valid: abcd1234

12. Write a Java method (takes a number n as input) to displays an n-by-n matrix.

Expected Output:

Input a number: 10

1 0 0 1 1 0 0 0 1 1

0 0 1 0 1 0 1 0 0 0

0 1 0 1 0 0 0 0 0 1

1 1 1 0 0 0 0 1 1 1

1 1 0 1 1 1 0 1 0 0

1 0 0 0 1 1 0 0 0 0

0 0 1 0 0 0 0 1 1 1

1 1 0 1 0 1 0 0 1 0

0 0 1 0 0 0 0 1 1 0

1 1 1 0 0 1 1 1 1 0

13. Write Java methods to calculate the area of a triangle.

Expected Output:

Input Side-1: 10

Input Side-2: 15

Input Side-3: 20

The area of the triangle is 72.6184377413890

14. Write a Java method to create the area of a pentagon.

Expected Output:

Input the number of sides: 5

Input the side: 6

The area of the pentagon is 61.93718642120281

15. Write a Java method to display the current date and time.

Expected Output:

Current date and time: Wednesday January 25, 2017 7:47:43

16. Write a Java method to find all twin prime numbers less than 100.

Expected Output:

(3, 5)

(5, 7)

(11, 13)

(17, 19)

(29, 31)

(41, 43)

(59, 61)

(71, 73)

Java Input-Output Exercises

==============================

1. Write a Java program to get a list of all file/directory names from the given.

2. Write a Java program to get specific files by extensions from a specified folder.

3. Write a Java program to check if a file or directory specified by pathname exists or not.

4. Write a Java program to check if a file or directory has read and write permission.

5. Write a Java program to check if given pathname is a directory or a file.

6. Write a Java program to compare two files lexicographically.

According to Wikipedia:

In mathematics, the lexicographic or lexicographical order (also known as lexical order, dictionary order, alphabetical order or lexicographic(al) product) is a generalization of the way the alphabetical order of words is based on the alphabetical order of their component letters. This generalization consists primarily in defining a total order over the sequences (often called words in computer science) of elements of a finite totally ordered set, often called alphabet.

7. Write a Java program to get last modified time of a file.

8. Write Java program to read input from java console.

9. Write a Java program to get file size in bytes, kb, mb.

10. Write a Java program to read contents from a file into byte array.

11. Write a Java program to read a file content line by line.

12. Write a Java program to read a plain text file.

13. Write a java program to read a file line by line and store it into a variable.

14. Write a Java program to store text file content line by line into an array.

15. Write a Java program to write and read a plain text file.

16. Write a Java program to append text to an existing file.

17. Write a Java program to read first 3 lines from a file.

18. Write a Java program to find the longest word in a text file.

Java Collection: ArrayList Exercises

=====================================

1. Write a Java program to create a new array list, add some colors (string) and print out the collection.

2. Write a Java program to iterate through all elements in a array list.

3. Write a Java program to insert an element into the array list at the first position.

4. Write a Java program to retrieve an element (at a specified index) from a given array list.

5. Write a Java program to update specific array element by given element.

6. Write a Java program to remove the third element from a array list.

7. Write a Java program to search an element in a array list.

8. Write a Java program to sort a given array list.

9. Write a Java program to copy one array list into another.

10. Write a Java program to shuffle elements in a array list.

11. Write a Java program to reverse elements in a array list.

12. Write a Java program to extract a portion of a array list.

13. Write a Java program to compare two array lists.

14. Write a Java program of swap two elements in an array list.

15. Write a Java program to join two array lists.

16. Write a Java program to clone an array list to another array list.

17. Write a Java program to empty an array list.

18. Write a Java program to test an array list is empty or not.

19. Write a Java program to trim the capacity of an array list the current list size.

20. Write a Java program to increase the size of an array list.

21. Write a Java program to replace the second element of a ArrayList with the specified element.

22. Write a Java program to print all the elements of a ArrayList using the position of the elements.

Java Collection: LinkedList Exercises

======================================

1. Write a Java program to append the specified element to the end of a linked list.

2. Write a Java program to iterate through all elements in a linked list.

3. Write a Java program to iterate through all elements in a linked list starting at the specified position.

4. Write a Java program to iterate a linked list in reverse order.

5. Write a Java program to insert the specified element at the specified position in the linked list.

6. Write a Java program to insert elements into the linked list at the first and last position.

7. Write a Java program to insert the specified element at the front of a linked list.

8. Write a Java program to insert the specified element at the end of a linked list.

9. Write a Java program to insert some elements at the specified position into a linked list.

10. Write a Java program to get the first and last occurrence of the specified elements in a linked list.

11. Write a Java program to display the elements and their positions in a linked list.

12. Write a Java program to remove a specified element from a linked list.

13. Write a Java program to remove first and last element from a linked list.

14. Write a Java program to remove all the elements from a linked list.

15. Write a Java program of swap two elements in an linked list.

16. Write a Java program to shuffle the elements in a linked list.

17. Write a Java program to join two linked lists.

18. Write a Java program to clone an linked list to another linked list.

19. Write a Java program to remove and return the first element of a linked list.

20. Write a Java program to retrieve but does not remove, the first element of a linked list.

21. Write a Java program to retrieve but does not remove, the last element of a linked list.

22. Write a Java program to check if a particular element exists in a linked list.

23. Write a Java program to convert a linked list to array list.

24. Write a Java program to compare two linked lists.

25. Write a Java program to test an linked list is empty or not.

26. Write a Java program to replace an element in a linked list.

Java Collection: HashSet Exercises

====================================

1. Write a Java program to append the specified element to the end of a hash set.

2. Write a Java program to iterate through all elements in a hash list.

3. Write a Java program to get the number of elements in a hash set.

4. Write a Java program to empty an hash set.

5. Write a Java program to test a hash set is empty or not.

6. Write a Java program to clone a hash set to another hash set.

7. Write a Java program to convert a hash set to an array.

8. Write a Java program to convert a hash set to a tree set.

9. Write a Java program to convert a hash set to a List/ArrayList.

10. Write a Java program to compare two hash set.

11. Write a Java program to compare two sets and retain elements which are same on both sets.

12. Write a Java program to remove all of the elements from a hash set.

Java Sorting Exercises

=============================

1. Write a Java program to sort an array of given integers using Quick sort Algorithm.

Quick sort is a comparison sort, meaning that it can sort items of any type for which a "less-than" relation (formally, a total order) is defined.

2. Write a Java program to sort an array of given integers using the Bubble sorting Algorithm.

According to Wikipedia "Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems even when compared to insertion sort. It can be practical if the input is usually in sort order but may occasionally have some out-of-order elements nearly in position."

3. Write a Java program to sort an array of given integers using Radix sort Algorithm.

According to Wikipedia "In computer science, radix sort is a non-comparative integer sorting algorithm that sorts data with integer keys by grouping keys by the individual digits which share the same significant position and value".

4. Write a Java program to sort an array of given integers using Merge sort Algorithm.

According to Wikipedia "Merge sort (also commonly spelled mergesort) is an O (n log n) comparison-based sorting algorithm. Most implementations produce a stable sort, which means that the implementation preserves the input order of equal elements in the sorted output."

5. Write a Java program to sort an array of given integers using Heap sort Algorithm.

In computer science, heapsort (invented by J. W. J. Williams in 1964) is a comparison-based sorting algorithm. Heapsort can be thought of as an improved selection sort: like that algorithm, it divides its input into a sorted and an unsorted region, and it interactively shrinks the unsorted region by extracting the largest element and moving that to the sorted region. The improvement consists of the use of a heap data structure rather than a linear-time search to find the maximum. Although somewhat slower in practice on most machines than a well-implemented quicksort, it has the advantage of a more favorable worst-case O(n log n) runtime. Heapsort is an in-place algorithm, but it is not a stable sort.

6. Write a Java program to sort an array of given integers using Selection Sort Algorithm.

According to Wikipedia "In computer science, selection sort is a sorting algorithm, specifically an in-place comparison sort. It has O(n2) time complexity, making it inefficient on large lists, and generally performs worse than the similar insertion sort".

7. Write a Java program to sort an array of given integers using Insertion sort Algorithm.

Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time. It is much less efficient on large lists than other algorithms such as quicksort, heapsort, or merge sort.

8. Write a Java program to sort an array of positive integers using the Bead Sort Algorithm.

According to Wikipedia "Bead sort, also called gravity sort, is a natural sorting algorithm, developed by Joshua J. Arulanandham, Cristian S. Calude and Michael J. Dinneen in 2002, and published in The Bulletin of the European Association for Theoretical Computer Science. Both digital and analog hardware implementations of bead sort can achieve a sorting time of O(n); however, the implementation of this algorithm tends to be significantly slower in software and can only be used to sort lists of positive integers. Also, it would seem that even in the best case, the algorithm requires O(n2) space".

9. Write a Java program to sort an array of positive integers using the BogoSort Sort Algorithm.

In computer science, bogosort is a particularly ineffective sorting algorithm based on the generate and test paradigm. The algorithm successively generates permutations of its input until it finds one that is sorted. It is not useful for sorting but may be used for educational purposes, to contrast it with other more realistic algorithms.

10. Write a Java program to sort an array of positive integers using the Cocktail sort Algorithm.

Cocktail shaker sort (also known as bidirectional bubble sort, cocktail sort, shaker sort, ripple sort, shuffle sort, or shuttle sort ) is a variation of bubble sort that is both a stable sorting algorithm and a comparison sort. The algorithm differs from a bubble sort in that it sorts in both directions on each pass through the list. This sorting algorithm is only marginally more difficult to implement than a bubble sort and solves the problem of turtles in bubble sorts. It provides only marginal performance improvements, and does not improve asymptotic performance; like the bubble sort, it is not of practical interest, though it finds some use in education.

11. Write a Java program to sort an array of given integers using the CombSort Algorithm.

The Comb Sort is a variant of the Bubble Sort. Like the Shell sort, the Comb Sort increases the gap used in comparisons and exchanges. Some implementations use the insertion sort once the gap is less than a certain amount. The basic idea is to eliminate turtles, or small values near the end of the list, since in a bubble sort these slow the sorting down tremendously. Rabbits, large values around the beginning of the list do not pose a problem in bubble sort.

12. Write a Java program to sort an array of given integers using CountingSort Algorithm.

According to Wikipedia "In computer science, counting sort is an algorithm for sorting a collection of objects according to keys that are small integers; that is, it is an integer sorting algorithm. It operates by counting the number of objects that have each distinct key value, and using arithmetic on those counts to determine the positions of each key value in the output sequence. Its running time is linear in the number of items and the difference between the maximum and minimum key values, so it is only suitable for direct use in situations where the variation in keys is not significantly greater than the number of items. However, it is often used as a subroutine in another sorting algorithm, radix sort, that can handle larger keys more efficiently".

13. Write a Java program to sort an array of given integers using Gnome sort Algorithm.

Gnome sort is a sorting algorithm originally proposed by Dr. Hamid Sarbazi-Azad (Professor of Computer Engineering at Sharif University of Technology) in 2000 and called "stupid sort" (not to be confused with bogosort), and then later on described by Dick Grune and named "gnome sort".

The algorithm always finds the first place where two adjacent elements are in the wrong order, and swaps them. It takes advantage of the fact that performing a swap can introduce a new out-of-order adjacent pair only next to the two swapped elements.

14. Write a Java program to sort an array of given integers using Pancake sort Algorithm.

Pancake sorting is the colloquial term for the mathematical problem of sorting a disordered stack of pancakes in order of size when a spatula can be inserted at any point in the stack and used to flip all pancakes above it. A pancake number is the minimum number of flips required for a given number of pancakes. The problem was first discussed by American geometer Jacob E. Goodman. It is a variation of the sorting problem in which the only allowed operation is to reverse the elements of some prefix of the sequence.

15. Write a Java program to sort an array of given integers using Permutation sort Algorithm.

Implement a permutation sort, which proceeds by generating the possible permutations of the input array/list until discovering the sorted one.

16. Write a Java program to sort an array of given integers Shell Sort Algorithm.

According to Wikipedia "Shell sort or Shell's method, is an in-place comparison sort. It can be seen as either a generalization of sorting by exchange (bubble sort) or sorting by insertion (insertion sort). The method starts by sorting pairs of elements far apart from each other, then progressively reducing the gap between elements to be compared. Starting with far apart elements can move some out-of-place elements into position faster than a simple nearest neighbor exchange."

17. Write a Java program to sort an array of given non-negative integers using Sleep Sort Algorithm.

Sleep sort works by starting a separate task for each item to be sorted, where each task sleeps for an interval corresponding to the item's sort key, then emits the item. Items are then collected sequentially in time.

18. Write a Java program to sort an array of given non-negative integers using Stooge Sort Algorithm.

Stooge sort is a recursive sorting algorithm with a time complexity of O(nlog 3 / log 1.5 ) = O(n2.7095...). The running time of the algorithm is thus slower compared to efficient sorting algorithms, such as Merge sort, and is even slower than Bubble sort.