

**Note:** There are **20** questions, write program for ALL of them.

1.	Write a Java program to sort a list of elements using the bubble sort algorithm. Bubble Sort works by repeatedly swapping the adjacent elements if they are in wrong order.
2.	Write a program in Java to find the second smallest element in an array. Input the size of array : 5 Input 5 elements in the array (value must be <9999) : element - 0 : 0 element - 1 : 9 element - 2 : 4 element - 3 : 6 element - 4 : 5 Expected Output : The Second smallest element in the array is : 4
3.	Write a program in Java for multiplication of two square Matrices. Test Data : Input the rows and columns of first matrix : 2 2 Input the rows and columns of second matrix : 2 2 Input elements in the first matrix : element - [0],[0] : 1 element - [0],[1] : 2 element - [1],[0] : 3 element - [1],[1] : 4 Input elements in the second matrix : element - [0],[0] : 5 element - [0],[1] : 6 element - [1],[0] : 7 element - [1],[1] : 8 Expected Output: The First matrix is : 1 2 3 4 The Second matrix is : 5 6 7 8 The multiplication of two matrix is : 19 22 43 50
4.	Write a program in Java to find sum of right diagonals of a matrix.

	<p>Test Data :</p> <p>Input the size of the square matrix : 2</p> <p>Input elements in the first matrix :</p> <p>element - [0],[0] : 1</p> <p>element - [0],[1] : 2</p> <p>element - [1],[0] : 3</p> <p>element - [1],[1] : 4</p> <p>Expected Output :</p> <p>The matrix is :</p> <p>1 2</p> <p>3 4</p> <p>Addition of the right Diagonal elements is :5</p>
5.	<p>Write a program in Java to calculate determinant of a 3 x 3 matrix.</p> <p>Test Data :</p> <p>Input elements in the first matrix :</p> <p>element - [0],[0] : 1</p> <p>element - [0],[1] : 0</p> <p>element - [0],[2] : -1</p> <p>element - [1],[0] : 0</p> <p>element - [1],[1] : 0</p> <p>element - [1],[2] : 1</p> <p>element - [2],[0] : -1</p> <p>element - [2],[1] : -1</p> <p>element - [2],[2] : 0</p> <p>Expected Output :</p> <p>The matrix is :</p> <p>1 0 -1</p> <p>0 0 1</p> <p>-1 -1 0</p> <p>The Determinant of the matrix is: 1</p>
6.	<p>Write a Java program to check whether a given substring is present in the given string. Also find the number of times a given word 'search' appears in the given string.</p> <p>Test Data:</p> <p>Input the string : This is a test string.</p> <p>Input the substring to be search : test</p> <p>Expected Output:</p> <p>Result: The substring is exist in the string.</p> <p>test has been found 1 time.</p> <p>Input the string : This is a test string. I am in love with this string.</p> <p>Input the substring to be search : string</p> <p>Expected Output:</p> <p>Result: The substring is exist in the string.</p> <p>string has been found 2 times.</p> <p>Input the string : This is a test string. I am in love with this string.</p> <p>Input the substring to be search : no</p> <p>Expected Output:</p> <p>Result: The substring is not exist in the string.</p> <p>no has not been found yet.</p>

7.	<p>Write a program in Java to find the largest and smallest word in a string.</p> <p>Test Data :</p> <p>Input the string : It is a string with smallest and largest word.</p> <p>Expected Output :</p> <p>The largest word is 'smallest' and the smallest word is 'a' in the string : 'It is a string with smallest and largest word.'.</p>
8.	<p>Write a Java program to print square root (<math>\sqrt{\phantom{x}}</math>), Ohm (<math>\Omega</math>), Plus Minus (<math>\pm</math>) and not equal (<math>\neq</math>).</p>
9.	<p>Write a program in Java to find the sum of the series <math>1!/1+2!/2+3!/3+4!/4+5!/5</math> using the function.</p> <p>Expected Output :</p> <p>The sum of the series is : 34</p>
10.	<p>Write a program in Java to check armstrong and perfect numbers using the function.</p> <p>Test Data :</p> <p>Input any number: 371</p> <p>Expected Output :</p> <p>The 371 is an Armstrong number. The 371 is not a Perfect number.</p>
11.	<p>Write a Java method to check whether a string is a valid password.</p> <p>Password rules:</p> <p>A password must have at least ten characters. A password consists of only letters and digits. A password must contain at least two digits.</p> <p>Expected Output:</p> <ol style="list-style-type: none"> <li>1. A password must have at least eight characters.</li> <li>2. A password consists of only letters and digits.</li> <li>3. A password must contain at least two digits</li> </ol> <p>Input a password (You are agreeing to the above Terms and Conditions.): abcd1234 Password is valid: abcd1234</p>
12.	<p>Write a Java method to find all twin prime numbers less than 100.</p> <p>Expected Output:</p> <p>(3, 5) (5, 7) (11, 13) (17, 19) (29, 31) (41, 43) (59, 61) (71, 73)</p>
13.	<p>Write a Java Program using recursion to Find Factorial of a Number.</p>

14.	<p>Write a program in Java to check whether two given strings are an anagram.  Test Data :  Input the first String : spare  Input the second String : pears  Expected Output :    spare and pears are Anagram.</p>
15.	<p>Consider the following definition of the class MyClass:</p> <pre> public class MyClass {     private static int count = 0;     private int x;      public MyClass(int i)     {         x = i;     }     public void incrementCount()     {         count++;     }      public void printX()     {         System.out.println("Value of x : " + x);     }      public static void printCount()     {         System.out.println("Value of count : " + count);     } }  public class MyClassDemo {     public static void main(String[] args)     {         MyClass myObject1 = new MyClass(5);         MyClass myObject2 = new MyClass(7);     } } </pre> <p>What is the output of the following Java code? (Assume that following statements are written inside main)</p> <pre> myObject1.printX(); myObject1.incrementCount(); MyClass.incrementCount(); myObject1.printCount(); myObject2.printCount(); myObject2.printX(); myObject1.setX(14); myObject1.incrementCount(); </pre>

	<pre>myObject1.printX(); myObject1.printCount(); myObject2.printCount();</pre>
16.	<p>Write a Java class Clock for dealing with the day time represented by hours, minutes, and seconds. Your class must have the following features:</p> <ul style="list-style-type: none"> <li>• Three instance variables for the hours (range 0 - 23), minutes (range 0 - 59), and seconds (range 0 - 59).</li> <li>• Three constructors: <ul style="list-style-type: none"> <li>◦ default (with no parameters passed; it should initialize the represented time to 12:0:0)</li> <li>◦ a constructor with three parameters: hours, minutes, and seconds.</li> <li>◦ a constructor with one parameter: the value of time in seconds since midnight (it should be converted into the time value in hours, minutes, and seconds)</li> </ul> </li> <li>• Instance methods: <ul style="list-style-type: none"> <li>◦ a <i>set</i>-method method setClock() with one parameter <i>seconds</i> since midnight (to be converted into the time value in hours, minutes, and seconds as above).</li> <li>◦ <i>get</i>-methods getHours(), getMinutes(), getSeconds() with no parameters that return the corresponding values.</li> <li>◦ <i>set</i>-methods setHours(), setMinutes(), setSeconds() with one parameter each that set up the corresponding instance variables.</li> <li>◦ method tick() with no parameters that increments the time stored in a Clock object by one second.</li> <li>◦ method addClock() accepting an object of type Clock as a parameter. The method should add the time represented by the parameter class to the time represented in the current class.</li> <li>◦ Add an instance method toString() with no parameters to your class. toString() must return a String representation of the Clock object in the form "(hh:mm:ss)", for example "(03:02:34)".</li> <li>◦ Add an instance method tickDown() which decrements the time stored in a Clock object by one second.</li> <li>◦ Add an instance method subtractClock() that takes one Clock parameter and returns the difference between the time represented in the current Clock object and the one represented by the Clock parameter. Difference of time should be returned as an clock object.</li> </ul> </li> </ul> <p>Write a separate class ClockDemo with a main() method. The program should:</p> <ul style="list-style-type: none"> <li>• instantiate a Clock object firstClock using one integer <i>seconds</i> since midnight obtained from the keyboard.</li> <li>• tick the clock ten times by applying its <i>tick()</i> method and print out the time after each tick.</li> <li>• Extend your code by appending to it instructions instantiating a Clock object secondClock by using three integers (hours, minutes, seconds) read from the</li> </ul>

	<p>keyboard.</p> <ul style="list-style-type: none"> <li>• Then tick the clock ten times, printing the time after each tick.</li> <li>• Add the secondClock time in firstClock by calling method addClock.</li> <li>• Print both clock objects calling toString method</li> </ul> <p>Create a reference thirdClock that should reference to object of difference of firstClock and secondClock by calling the method subtractClock().</p>
17.	<p>Write a Java class Complex for dealing with complex number. Your class must have the following features:</p> <ul style="list-style-type: none"> <li>• Instance variables : <ul style="list-style-type: none"> <li>◦ <b>realPart</b> for the real part of type double</li> <li>◦ <b>imaginaryPart</b> for imaginary part of type double.</li> </ul> </li> <li>• Constructor: <ul style="list-style-type: none"> <li>◦ <b>public Complex ()</b>: A default constructor, it should initialize the number to 0, 0)</li> <li>◦ <b>public Complex (double realPart, double imaginaryPart)</b>: A constructor with parameters, it creates the complex object by setting the two fields to the passed values.</li> </ul> </li> <li>• Instance methods: <ul style="list-style-type: none"> <li>◦ <b>public Complex add (Complex otherNumber)</b>: This method will find the sum of the current complex number and the passed complex number. The methods returns a new Complex number which is the sum of the two.</li> <li>◦ <b>public Complex subtract (Complex otherNumber)</b>: This method will find the difference of the current complex number and the passed complex number. The methods returns a new Complex number which is the difference of the two.</li> <li>◦ <b>public Complex multiply (Complex otherNumber)</b>: This method will find the product of the current complex number and the passed complex number. The methods returns a new Complex number which is the product of the two.</li> <li>◦ <b>public Complex divide (Complex otherNumber)</b>: This method will find the ... of the current complex number and the passed complex number. The methods returns a new Complex number which is the ... of the two.</li> <li>◦ <b>public void setRealPart (double realPart)</b>: Used to set the real part of this complex number.</li> <li>◦ <b>public void setImaginaryPart (double realPart)</b>: Used to set the imaginary part of this complex number.</li> <li>◦ <b>public double getRealPart()</b>: This method returns the real part of the complex number</li> <li>◦ <b>public double getImaginaryPart()</b>: This method returns the imaginary part of the complex number</li> <li>◦ <b>public String toString()</b>: This method allows the complex number to be easily printed out to the screen</li> </ul> <p>Write a separate class <b>ComplexDemo</b> with a main() method and test the Complex class methods.</p> </li></ul>

18.	<p>Write a Java class Author with following features:</p> <ul style="list-style-type: none"> <li>• Instance variables : <ul style="list-style-type: none"> <li>◦ <b>firstName</b> for the author's first name of type String.</li> <li>◦ <b>lastName</b> for the author's last name of type String.</li> </ul> </li> <li>• Constructor: <ul style="list-style-type: none"> <li>◦ <b>public Author (String firstName, String lastName)</b>: A constructor with parameters, it creates the Author object by setting the two fields to the passed values.</li> </ul> </li> <li>• Instance methods: <ul style="list-style-type: none"> <li>◦ <b>public void setFirstName (String firstName)</b>: Used to set the first name of author.</li> <li>◦ <b>public void setLastName (String lastName)</b>: Used to set the last name of author.</li> <li>◦ <b>public double getFirstName()</b>: This method returns the first name of the author.</li> <li>◦ <b>public double getLastName()</b>: This method returns the last name of the author.</li> <li>◦ <b>public String toString()</b>: This method printed out author's name to the screen</li> </ul> </li> </ul> <p>Write a Java class Book with following features:</p> <ul style="list-style-type: none"> <li>• Instance variables : <ul style="list-style-type: none"> <li>◦ <b>title</b> for the title of book of type String.</li> <li>◦ <b>author</b> for the author's name of type String.</li> <li>◦ <b>price</b> for the book price of type double.</li> </ul> </li> <li>• Constructor: <ul style="list-style-type: none"> <li>◦ <b>public Book (String title, Author name, double price)</b>: A constructor with parameters, it creates the Author object by setting the the fields to the passed values.</li> </ul> </li> <li>• Instance methods: <ul style="list-style-type: none"> <li>◦ <b>public void setTitle(String title)</b>: Used to set the title of book.</li> <li>◦ <b>public void setAuthor(String author)</b>: Used to set the name of author of book.</li> <li>◦ <b>public void setPrice(double price)</b>: Used to set the price of book.</li> <li>◦ <b>public double getTitle()</b>: This method returns the title of book.</li> <li>◦ <b>public double getAuthor()</b>: This method returns the author's name of book.</li> <li>◦ <b>public String toString()</b>: This method printed out book's details to the screen</li> </ul> </li> </ul> <p>Write a separate class <b>BookDemo</b> with a main() method creates a Book titled "Developing Java Software" with authors Russel Winderand price 79.75. Prints the Book's string representation to standard output (using System.out.println).</p>
-----	--