

**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY  
(KMIT)**

**(Approved by AICTE, New Delhi and Affiliated to JNTUH)**

**Narayanaguda, Hyderabad – 500029**

**(Autonomous)**



**DEPARTMENT OF CSE (DATA SCIENCE)**

**LAB MANUAL**

**JAVA PROGRAMMING LAB**

**B.Tech II YEAR II SEM (KR20 REGULATIONS)**

**ACADEMIC YEAR 2021-22**



## **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

(Approved by AICTE & Govt of T.S and Affiliated to JNTUH)  
3-5-1026, Narayanaguda, Hyderabad-29. Ph: 040-23261407

### **Department Of CSE (DATA SCIENCE)**

#### **Vision of the Institution:**

To be the fountain head of latest technologies, producing highly skilled, globally competent engineers.

#### **Mission of the Institution:**

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students



## **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

(Approved by AICTE & Govt of T.S and Affiliated to JNTUH)  
3-5-1026, Narayanaguda, Hyderabad-29. Ph: 040-23261407

### **Department Of CSE (DATA SCIENCE)**

### **Vision & Mission of Department**

#### **Vision of the Department:**

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

#### **Mission of the Department:**

- To provide faculty with state-of-the-art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Approved by AICTE & Govt of T.S and Affiliated to JNTUH)  
3-5-1026, Naravanaguda, Hyderabad-29. Ph: 040-23261407



## Department Of CSE (DATA SCIENCE)

### PROGRAM OUTCOMES (POs)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



## **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

(Approved by AICTE & Govt of T.S and Affiliated to JNTUH)  
3-5-1026, Narayanaguda, Hyderabad-29. Ph: 040-23261407

### **Department Of CSE (DATA SCIENCE)**

#### **PROGRAM SPECIFIC OUTCOMES (PSOs)**

**PSO1:** An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftments.

**PSO2:** Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.



## **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

(Approved by AICTE & Govt of T.S and Affiliated to JNTUH)  
3-5-1026, Narayanaguda, Hyderabad-29. Ph: 040-23261407

### **Department of CSE (DATA SCIENCE)**

#### **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

**PEO1:** Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

**PEO2:** Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

**PEO3:** Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

**PEO4:** Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

## **COURSE OBJECTIVE:**

This lab complements the Java Programming course

- 1 To introduce the object-oriented programming concepts.
- 2 To understand object-oriented programming concepts, and apply them in solving problems.
- 3 To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- 4 To introduce the implementation of packages and interfaces
- 5 To introduce the concepts of exception handling and multithreading.
- 6 To introduce the design of Graphical User Interface using applets and swing controls.

## **COURSE OUTCOMES:**

At the end of the course the student is able to

<b>CO1:</b> Design application for solving real world problems.
<b>CO2:</b> Able to debug and test the application
<b>CO3:</b> Able to use JDBC application efficiently.
<b>CO4:</b> Able to use collection framework to suit the needs of the application.
<b>CO5:</b> Able to analyze the multithreaded applications and use it to optimize the code.

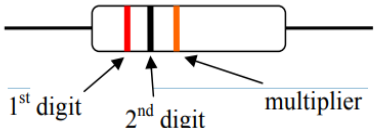
## **CO VS PO MAPPING:**

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12
CO1	2	1	2									
CO2				2	2							
CO3	1		2		2							
CO4		3	1									
CO5	2	1	1		2							

## **CO-PSO Matrix:**

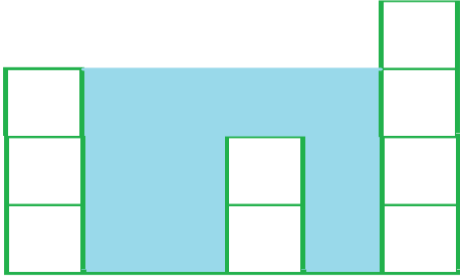

	PSO-1	PSO-2
CO1	1	1
CO2	2	3
CO3	1	2
CO4	2	3
CO5	2	3

## List Of Experiments

Exp No.	Experiment Name	Page No.
1	<p><b>1. a.</b> Given a number, check if the number (N) can be written as the form <math>(k+1)*k</math>. Write a java program to print those numbers in the given range.</p> <p><b>b.</b> Write a java program to check whether the given number is gapful or not. A number is gapful if it is at least 3 digits long and is divisible by the number formed by stringing the first and last numbers together. The smallest number that fits this description is 100. First digit is 1, last digit is 0, forming 10, which is a factor of 100. Therefore, 100 is gapful.</p> <p><b>c. Cricketer's Pension</b> Continuing our journey in mastering the conditional statements &amp; our interest with cricket. Let us help the Indian cricket's governing body (BCCI) to automate its plan of allotting pensions to former players. The rules are given below: If a player has played more than 10 test matches and 100 ODI's he receives Rs.50,000. If a player has played more than 10 test matches he receives Rs.25,000. If a player has played more than 100 ODI's he receives Rs.15,000. If a player has played for India he receives Rs.10,000. The amount is incremented by 1/4th for every 'man of the match' award. If a player has not played for India but played IPL he receives an amount of Rs.8,000. If a player has not played for India nor IPL he receives an amount of Rs.7,000.</p>	15
2	<p>A resistor is a circuit device designed to provide a specific resistance between its two ends. Resistance is expressed in ohms (<math>\Omega</math>) or kilo-ohms (<math>k\Omega</math>). Resistors are usually marked with coloured bands that encode their resistance, as shown in figure 1 below. The first two bands represent digits and the third is a power-of-ten multiplier.</p>  <p style="text-align: center;">Figure 1: Colour Bands of a resistor</p>	19



	<p>The table below shows the number value of each band colour. For example, if the first band is red (represents 2), the second is black (represents 0), and the third is orange (represents 3), the resistance is <math>20 \times 10^3 \Omega</math> or 20 k<math>\Omega</math>.</p> <table><tr><th>Colour</th><th>Number Value</th></tr><tr><td>Black</td><td>0</td></tr><tr><td>Brown</td><td>1</td></tr><tr><td>Red</td><td>2</td></tr><tr><td>Orange</td><td>3</td></tr><tr><td>Yellow</td><td>4</td></tr><tr><td>Green</td><td>5</td></tr><tr><td>Blue</td><td>6</td></tr><tr><td>Violet</td><td>7</td></tr><tr><td>Grey</td><td>8</td></tr><tr><td>White</td><td>9</td></tr></table> <p>Write a Resistor class containing the parameterized constructor, which takes in three strings representing the three band colours. Write the methods to calculate and set the resistance for the resistor.</p>	Colour	Number Value	Black	0	Brown	1	Red	2	Orange	3	Yellow	4	Green	5	Blue	6	Violet	7	Grey	8	White	9	
Colour	Number Value																							
Black	0																							
Brown	1																							
Red	2																							
Orange	3																							
Yellow	4																							
Green	5																							
Blue	6																							
Violet	7																							
Grey	8																							
White	9																							
3	<p><b>a.</b>Given an array <code>arr[]</code> of <b>N</b> integers, the task is to find the maximum difference between any two elements of the array.</p> <p><b>b.</b> Write a Java program to fill the below pattern into a square matrix: The matrix has to be filled with numbers starting from 1. It has to start fill first row last column, last row (reverse), first column (reverse) and so on. Please refer the following example input = 5 output =</p> <pre>1   2   3   4   5 16  17  18  19  6 15  24  25  20  7 14  23  22  21  8 13  12  11  10  9</pre> <p><b>c.</b> Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining. Examples: Input: <code>arr[] = {3, 0, 0, 2, 0, 4}</code> Output: 10</p>	22																						

	 <p>Bars for input {3, 0, 0, 2, 0, 4} Total trapped water = 3 + 3 + 1 + 3 = 10</p> <p>We can trap "3*2 units" of water between 3 and 2, "1 unit" on top of bar 2 and "3 units" between 2 and 4.</p>	
4	<p><b>a.</b> Given two strings text1 and text2, return the length of their longest common subsequence. A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters. (eg, "ace" is a subsequence of "abcde" while "aec" is not). A common subsequence of two strings is a subsequence that is common to both strings. If there is no common subsequence, return 0.</p> <p><b>b.</b> Given two strings s1 and s2, your task is to merge those strings to form a new merged string. A merge operation on two strings is described as follows: Append alternating characters from s1 and s2, respectively, to mergedString. Once all of the characters in one of the strings have been merged, append the remaining characters in the other string to mergedString.</p> <p><b>c.</b> You are given a string S of alphabet characters and the task is to find its matching decimal representation as on the shown keypad. Output the decimal representation corresponding to the string. For ex: if you are given "amazon" then its corresponding decimal representation will be 262966.</p> 	27
5	Define a MyRectangle class with four public data members representing the x- and y-coordinates of the bottom-left vertex and	32

	<p>top-right vertex of a rectangle, whose sides are parallel to the x- or y-axis.</p> <p>For example, the statement <code>new MyRectangle(20,80,30,90)</code> creates a rectangle with bottom-left vertex at position (20,80), and top-right vertex at (30,90).</p> <p>(a) Write an <code>area()</code> method, which computes the area of a rectangle.</p> <p>(b) Write the <code>overlap(MyRectangle rect)</code> method. This method returns a rectangle which is the overlapped region of two rectangles. In the event that there is no overlap, it should return a rectangle with both bottom-left vertex and top-right vertex at position (0,0).</p> <p>(c) Using the <code>overlap(MyRectangle rect)</code> method written above, write the <code>overlapAll(MyRectangle[] rectangles)</code> method which returns the overlapped region of all the rectangles in the array. You may assume that there is at least one element in the array. Your method should be efficient in that the moment it finds that the overlapped region is empty, it should return a rectangle with both vertices at (0,0) immediately.</p> <p>(d) Write <code>MySquare.java</code>, <code>MySquare</code> extends <code>MyRectangle</code>. A square is defined by its bottom-left vertex and the length of its side. Complete the <code>super( . . . )</code> statement in the constructor.</p> <p>(e) Below is output of <code>MySquare.java</code> program when the user enters: 10 30 5.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Class <code>MySquare</code>: [(10,30); (15,35)] Area = 25</p> </div> <p>Override the <code>toString()</code> method in <code>MyRectangle</code> in order to get such output.</p>	
6	<p><b>a.</b> Write a java program that loads names and phone numbers from the text file into <code>HashTable</code> where data is organized as one line per record and each field in record are separated by a tab(<code>\t</code>). It takes a name or phone number as input and prints the corresponding other value from hash table.</p> <p>You have created a web-based survey of favorite programming languages and are capturing the results into a text file named “logfile”. The structure of the text file is:</p> <pre> Total # Entries Vote for Entry 1 IP Address for Entry 1 Timestamp in seconds for Entry 1 Vote for Entry 2 IP Address for Entry 2 Timestamp in seconds for Entry 2 ... </pre> <p>For example, here is a sample logfile of six entries:</p>	35

	<pre> 6 PHP 137.229.156.12 1000002 C# 137.229.156.18 1000005 PHP 137.229.156.12 1000006 Prolog 156.213.38.31 1000010 PHP 128.120.56.214 1000020 PHP 137.229.156.12 1000022 </pre> <p>The logfile is ordered by increasing timestamp. You are concerned that some people are voting multiple times for the same item. To somewhat address this problem, throw out any new votes for the same item that come from the same IP address within 20 seconds.</p> <p>In the above example, the second and last votes for PHP would be thrown out because they are for the same item from the same IP address and occur within 20 seconds of other PHP votes from the same IP address. However, the PHP vote from 128.120.56.214 would be retained since there is not another PHP vote from this IP address.</p> <p>Write a Java program to count the votes from the logfile, throwing out duplicate votes using the rules above. Display the votes in a table, as shown below for the example:</p> <pre> PHP 2 C# 1 Prolog 1 </pre>	
7	<p><b>a.</b> Write a Java Class to implement a method Addition() that returns a new Array where each array element at the index k corresponds to the sum of elements of the array (src) starting at index 0 and including element at the index 'k'. For example, for array [2,3,5], the method will return array [2,5,10]. For an array of size '0' or a null parameter, the method will throw exception IllegalArgumentExceptionWith the message "Invalid Argument".</p> <p><b>b.</b> Write a Java Code to implement a multithreaded version of FizzBuzz with four threads.</p> <p>If the number is divisible by 3, output "fizz".  If the number is divisible by 5, output "buzz".  If the number is divisible by both 3 and 5, output "fizzbuzz".  If the number is not divisible by both 3 and 5 print the number</p> <p>For instance if n is 15, we will have the output as " 1, 2, fizz, 4, buzz, fizz, 7, 8, fizz,buzz, 11, fizz, 13, 14, fizzbuzz "</p> <p>1. Thread A will call fizz() to check for divisibility of 3 and outputs fizz .  2. Thread B will call buzz() to check for divisibility of 5 and outputs</p>	40

	<p>buzz .</p> <p>3. Thread C will call fizzbuzz() to check for divisibility of 3 and 5 and outputs fizzbuzz .</p> <p>4. Thread D will call number() which should only output the numbers.</p>																									
8	<p><b>a.</b> Write a java program to store the employee details in an ArrayList and display the employee details in ascending order of their experience. Create 'Employee' class with two instance variables Employee name and Employee experience (no. of years).</p> <p><b>b.</b> Write a program to find the most common words in the list of words given in sorted order based on occurrence from largest to smallest. If any of words are having same occurrence then consider the smallest character order comes first.</p> <p>Input format: First line contains the list of words and next line contains a number (k) which represent the top most words to display.</p> <p>Output format: display the k top most words.</p>	44																								
9	<p><b>a.</b>Write a java Program to write a method fCount which takes a string as a parameter. The Method fCount should return the Map which has the frequency count of the given word. For example if the string passed is “hello” the map should return {h-1,e-1,l-2,o-1}. The order of the characters should be same as in the string.</p> <p><b>b.</b> When working with HashMaps, sometimes cases arise where we wish to determine if two HashMaps have any key/value pairs in common. For example, we might have the following two Hashmaps (named hashmap1 and hashmap2, respectively) that map from String to String (i.e., their type is HashMap) and we want to count how many key/value pairs they have in common.</p> <div><div><p><b>hashmap1</b></p><table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Alice</td><td>Healthy</td></tr><tr><td>Mary</td><td>Ecstatic</td></tr><tr><td>Bob</td><td>Happy</td></tr><tr><td>Chuck</td><td>Fine</td></tr><tr><td>Felix</td><td>Sick</td></tr></tbody></table></div><div><p><b>hashmap2</b></p><table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Mary</td><td>Ecstatic</td></tr><tr><td>Felix</td><td>Healthy</td></tr><tr><td>Ricardo</td><td>Superb</td></tr><tr><td>Tam</td><td>Fine</td></tr><tr><td>Bob</td><td>Happy</td></tr></tbody></table></div></div> <p>In the example above, these two HashMaps have two key/value pairs in common, namely: "Mary"/"Ecstatic" and "Bob"/"Happy". Note that although the key "Felix" is in both HashMaps, the associated value with this key is different in the two maps (hence this does not count as a key/value pair that is common to both HashMaps). Similarly, just having the same value without the same key (such as the value "Fine" which is mapped to by different keys in the two different HashMaps) would also not count as a common key/value pair between the two HashMaps.</p>	Key	Value	Alice	Healthy	Mary	Ecstatic	Bob	Happy	Chuck	Fine	Felix	Sick	Key	Value	Mary	Ecstatic	Felix	Healthy	Ricardo	Superb	Tam	Fine	Bob	Happy	47
Key	Value																									
Alice	Healthy																									
Mary	Ecstatic																									
Bob	Happy																									
Chuck	Fine																									
Felix	Sick																									
Key	Value																									
Mary	Ecstatic																									
Felix	Healthy																									
Ricardo	Superb																									
Tam	Fine																									
Bob	Happy																									

	<p>Your job is to write a method:</p> <pre>public int commonKeyValuePairs(HashMap&lt;String,String&gt; map1,                                 HashMap&lt;String,String&gt; map2)</pre> <p>that is passed two objects of type <code>HashMap&lt;String,String&gt;</code> and returns the number of common key/value pairs between the two HashMaps.</p>	
10	<p><b>a.</b> Demonstrate a JDBC Program on Employee Schema given below  Read a department number from the user and display those employee names in ascending order who are working in the department.  Note - Display records based on employee names in ascending order  The DB Credentials  Name of the DB - test;  Name of the table- emp;  JDBC_DRIVER = "com.mysql.jdbc.Driver";  DB_URL = "jdbc:mysql://localhost/test";  Username-student  Password-student  emp schema;</p> <pre>+-----+-----+-----+-----+-----+   Field   Type   Null   Key   Default   Extra   +-----+-----+-----+-----+-----+   empno   int(4)   NO   PRI   NULL       ename   varchar(50)   NO     NULL       job   varchar(50)   NO     NULL       mgr   int(4)   YES     NULL       hiredate   date   YES     NULL       sal   decimal(10,2)   YES     NULL       comm   decimal(10,2)   YES     NULL       deptno   int(2)   YES   MUL   NULL     +-----+-----+-----+-----+-----+</pre> <p><b>b.</b> Demonstrate a JDBC program to display all the employee names in ascending order who are working in "Dallas" location using <code>createStatement()</code>  Schema Given below  emp schema;</p> <pre>+-----+-----+-----+-----+-----+   Field   Type   Null   Key   Default   Extra   +-----+-----+-----+-----+-----+   empno   int(4)   NO   PRI   NULL       ename   varchar(50)   NO     NULL       job   varchar(50)   NO     NULL       mgr   int(4)   YES     NULL       hiredate   date   YES     NULL       sal   decimal(10,2)   YES     NULL       comm   decimal(10,2)   YES     NULL       deptno   int(2)   YES   MUL   NULL     +-----+-----+-----+-----+-----+</pre>	51

	<pre> +-----+-----+-----+-----+-----+ dept schema +-----+-----+-----+-----+-----+   Field   Type   Null   Key   Default   Extra   +-----+-----+-----+-----+-----+   deptno   int(2)   NO   PRI   NULL       dname   varchar(50)   NO     NULL       location   varchar(50)   NO     NULL     +-----+-----+-----+-----+-----+  c. Demonstrate JDBC program to read three values (dno,dname,dloc) from the user and insert those records into the dept table using Prepared Statement The Reading of input should be first deptno followed by deptname followed by deptlocation Dept Schema +-----+-----+-----+-----+-----+   Field   Type   Null   Key   Default   Extra   +-----+-----+-----+-----+-----+   deptno   int(2)   NO   PRI   NULL       dname   varchar(50)   NO     NULL       location   varchar(50)   NO     NULL     +-----+-----+-----+-----+-----+ </pre>	
--	---	--

## EXPERIMENT 1

**1. a. Given a number, check if the number (N) can be written as the form  $(k+1)*k$ . Write a java program to print those numbers in the given range.**

```
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                if(i==(j+1)*j)
                    System.out.println(i);
            }
        }
    }
}
```

### EXPECTED OUTPUT:

**Sample Input:**

**10**

**Sample Output:**

**2**

**6**

**b. Write a java program to check whether the given number is gapful or not.**

A number is gapful if it is at least 3 digits long and is divisible by the number formed by stringing the first and last numbers together. The smallest number that fits this description is 100. First digit is 1, last digit is 0, forming 10, which is a factor of 100. Therefore, 100 is gapful.

```
import java.util.*;
class Gapfull
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        System.out.println(gapful(a));
    }

    public static int gapful(int n) {
```



```

        for (int i = 0; n - i > 100; i++) {
            boolean low = isGapful(n - i);
            boolean high = isGapful(n + i);
            if (low || high) {
                return low ? n - i : n + i;
            }
        }
        return 100;
    }

    private static boolean isGapful(int n) {
        String s = "" + n;
        int div = Integer.parseInt("" + s.charAt(0) + s.charAt(s.length() - 1));
        return n % div == 0;
    }
}

```

### EXPECTED OUTPUT:

case=1  
 input=25  
 output=100

case=2  
 input=100  
 output=100

case=3  
 input=103  
 output=105

**c. Cricketer's Pension** Continuing our journey in mastering the conditional statements & our interest with cricket.

Let us help the Indian cricket's governing body (BCCI) to automate its plan of allotting pensions to former players.

The rules are given below:

If a player has played more than 10 test matches and 100 ODI's he receives Rs.50,000.

If a player has played more than 10 test matches he receives Rs.25,000.

If a player has played more than 100 ODI's he receives Rs.15,000.

If a player has played for India he receives Rs.10,000.

The amount is incremented by 1/4th for every 'man of the match' award.

If a player has not played for India but played IPL he receives an amount of Rs.8,000.

If a player has not played for India nor IPL he receives an amount of Rs.7,000.

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("has the player played for india[y/n]?");
        String ch=sc.next();
        System.out.println("enter the no of test matches he has played:");
        int test_matches=sc.nextInt();
        System.out.println("enter the no of ODI matches he has played:");
        int ODI_matches=sc.nextInt();
        System.out.println("no of man of the matches he has won:");
        int man_of_the_matches=sc.nextInt();
        float amount=0.0f;
        if(test_matches>10 &&ODI_matches>100){
            amount+=50000;
        }
        else if(test_matches>10){
            amount+=25000;
        }
        else if(ODI_matches>100){
            amount+=15000;
        }
        else if(ch.equals('y')){
            amount+=10000;
        }
        amount=amount+(amount*(man_of_the_matches*0.25f));
        System.out.println(amount);
    }
}
```

}

}

**EXPECTED OUTPUT:**

**Has the player played for india[y/n]?:**

**y**

**Enter the no of test matches he has played:**

**20**

**Enter the no of ODI matches he has played:**

**120**

**No of man of the matches he has won:**

**50**

**675000.0**

## EXPERIMENT 2

2. A resistor is a circuit device designed to provide a specific resistance between its two ends. Resistance is expressed in ohms ( $\Omega$ ) or kilo-ohms ( $k\Omega$ ). Resistors are usually marked with coloured bands that encode their resistance, as shown in figure 1 below. The first two bands represent digits and the third is a power-of-ten multiplier.

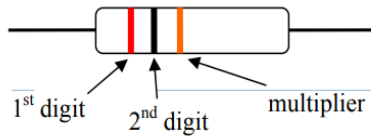


Figure 1: Colour Bands of a resistor

The table below shows the number value of each band colour. For example, if the first band is red (represents 2), the second is black (represents 0), and the third is orange (represents 3), the resistance is  $20 \times 10^3 \Omega$  or 20 k $\Omega$ .

Colour	Number Value
Black	0
Brown	1
Red	2
Orange	3
Yellow	4
Green	5
Blue	6
Violet	7
Grey	8
White	9

Write a Resistor class containing the parameterized constructor, which takes in three strings representing the three band colours. Write the methods to calculate and set the resistance for the resistor.

```
import java.util.*;

class Resistor{
    HashMap<String,Integer> color;
    // data members
    private String band1;
    private String band2;
    private String band3;
    private double value; // Resistor value (Ohms)

    // default constructor
```

```

public Resistor()
{
    band1=band2=band3="";
    value=0.0;
}
// Constructor taking three strings as parameters
// and calculating the resistance
public Resistor(String b1, String b2, String b3)
{
    // *** to be completed ***
    band1=b1;
    band2=b2;
    band3=b3;

    color=new HashMap<>();
    color.put("black",0);
    color.put("brown",1);
    color.put("red",2);
    color.put("orange",3);
    color.put("yellow",4);
    color.put("green",5);
    color.put("blue",6);
    color.put("violet",7);
    color.put("grey",8);
    color.put("white",9);

}

public String getResistance(){
    int firstdig=color.get(band1);
    int seconddig=color.get(band2);
    int thirddig=(int)Math.pow(10,color.get(band3));
    String value1=Integer.toString(firstdig)+Integer.toString(seconddig);

```

```
        value=Integer.parseInt(value1)*thirddig;
        value=value/1000;
        value1=value+"Kilo_ohms";
        return value1;
    }
}

public class Testdemo{
    public static void main(String args[]){
        Resistor r1=new Resistor("red","black","orange");
        System.out.println(r1.getResistance());
    }
}
```

#### **EXPECTED OUTPUT**

20.0Kilo\_ohms

### EXPERIMENT 3

**3. a. Given an array arr[] of N integers, the task is to find the maximum difference between any two elements of the array.**

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++){
            a[i]=sc.nextInt();
        }
        int diff=0;
        for(int i=0;i<n;i++){
            for(int j=i+1;j<n;j++){
                if(Math.abs(a[i]-a[j])>diff){
                    diff=Math.abs(a[i]-a[j]);
                }
            }
        }
        System.out.println(diff);
    }
}
```

#### EXPECTED OUTPUT:

**Sample Input:**

**5**

**10 20 30 40 50**

**Sample Output:**

**40**

**b.** Write a Java program to fill the below pattern into a square matrix:

The matrix has to be filled with numbers starting from 1. It has to start fill first row last column, last row (reverse), first column (reverse) and so on. Please refer the following example

input = 5

output =

```
1   2   3   4   5
16  17  18  19   6
15  24  25  20   7
14  23  22  21   8
13  12  11  10   9
```

```
import java.util.*;
```

```
public class Main
```

```
{
```

```
    public static void main(String[] args) {
```

```
        Scanner sc=new Scanner(System.in);
```

```
        int n=sc.nextInt();
```

```
        int rowBegin = 0;
```

```
        int rowEnd = n-1;
```

```
        int colBegin = 0;
```

```
        int colEnd = n- 1;
```

```
        int[][] a=new int[n][n];
```

```
        while (rowBegin<= rowEnd&&colBegin<= colEnd) {
```

```
            // Traverse Right
```

```
            for (int j = colBegin; j <= colEnd; j ++ ) {
```

```
                a[rowBegin][j]=sc.nextInt();
```

```
            }
```

```
        rowBegin++;
```

```
        // Traverse Down
```

```
        for (int j = rowBegin; j <= rowEnd; j ++ ) {
```

```
            a[j][colEnd]=sc.nextInt();
```



```

    }
colEnd--;
    if (rowBegin<= rowEnd) {
        // Traverse Left
        for (int j = colEnd; j >= colBegin; j --) {
            a[rowEnd][j]=sc.nextInt();
        }
    }
rowEnd--;

    if (colBegin<= colEnd) {
        // Traver Up
        for (int j = rowEnd; j >= rowBegin; j --) {
            a[j][colBegin]=sc.nextInt();
        }
    }
colBegin ++;
}

for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
System.out.print(a[i][j]+" ");
}
System.out.println();
}
}

```

### **EXPECTED OUTPUT:**

**Sample Input:**

**3**  
**2 4 5 3 6 8 1 9 7**

**Sample Output:**

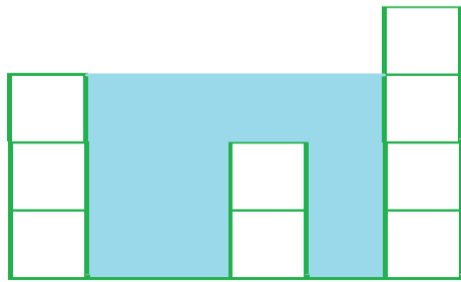
2 4 5  
9 7 3  
1 8 6

c. Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

Examples:

Input: arr[] = {3, 0, 0, 2, 0, 4}

Output: 10



Bars for input {3, 0, 0, 2, 0, 4}  
Total trapped water = 3 + 3 + 1 + 3 = 10

We can trap "3\*2 units" of water between 3 and 2, "1 unit" on top of bar 2 and "3 units" between 2 and 4.

```
class Rainwater{
```

```
// Function to return the maximum water that can be stored
```

```
public static int maxWater(int[] arr, int n)
```

```
{
```

```
    // To store the maximum water that can be stored
```

```
    int res = 0;
```

```
    // For every element of the array except first and last element
```

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

```
    {
```

```
        // Find maximum element on its left
```

```
        int left = arr[i];
```

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

```
        {
```

```
            left = Math.max(left, arr[j]);
```

```
        }
```

```
        // Find maximum element on its right
```

```
        int right = arr[i];
```

```

        for(int j = i + 1; j < n; j++)
        {
            right = Math.max(right, arr[j]);
        }

        // Update maximum water value
        res += Math.min(left, right) - arr[i];
    }
    return res;
}

public static void main(String[] args)
{
    int[] arr = { 3,0,0,2,0,4 };
    int n = arr.length;

    System.out.print(maxWater(arr,n));
}
}

```

**EXPECTED OUTPUT:**

**10**

## EXPERIMENT 4

**4.a. Given two strings text1 and text2, return the length of their longest common subsequence. A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters. (eg, "ace" is a subsequence of "abcde" while "aec" is not). A common subsequence of two strings is a subsequence that is common to both strings. If there is no common subsequence, return 0.**

```
class LCS
{
    public static int longestCommonSubsequence(String text1, String text2)
    {
        int [][]dp=new int[text1.length()+1][text2.length()+1];
        for(int i=1;i<=text1.length();i++)
        {
            for(int j=1;j<=text2.length();j++)
            {
                if(text1.charAt(i-1)==text2.charAt(j-1))
                    dp[i][j]=dp[i-1][j-1]+1;
            }
            else
            {
                dp[i][j]=Math.max(dp[i-1][j],dp[i][j-1]);
            }
        }
        return dp[text1.length()][text2.length()];
    }
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println(longestCommonSubsequence(sc.next(),sc.next()));
    }
}
```

**EXPECTED OUTPUT:**

**Sample Input:**

abcdefghijkl

caeghiklp

**Sample Output: 6**

**b.** Given two strings s1 and s2, your task is to merge those strings to form a new merged string. A merge operation on two strings is described as follows:

Append alternating characters from s1 and s2, respectively, to mergedString. Once all of the characters in one of the strings have been merged, append the remaining characters in the other string to mergedString.

```
import java.io.*;
import java.util.*;
class MergedStr{
    public static String mergeAlternately(String word1, String word2) {
        StringBuilder sb = new StringBuilder();
        int index1 = 0, index2 = 0;
        while (index1 < word1.length() && index2 < word2.length()) {
            sb.append(word1.charAt(index1));
            sb.append(word2.charAt(index2));
            index1++;
            index2++;
        }
        while (index1 < word1.length()) {
            sb.append(word1.charAt(index1));
            index1++;
        }
        while (index2 < word2.length()) {
            sb.append(word2.charAt(index2));
            index2++;
        }
        return sb.toString();
    }
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println(mergeAlternately(sc.next(),sc.next()));
    }
}
```

```
}  
}
```

**EXPECTED OUTPUT:**

**Sample Input:**

**abc**

**def**

**Sample Output:**

**abcdef**

c. You are given a string S of alphabet characters and the task is to find its matching decimal representation as on the shown keypad. Output the decimal representation corresponding to the string.

For ex: if you are given “amazon” then its corresponding decimal representation will be 262966.

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
*	0	□

```
import java.io.*;  
import java.util.*;  
class Keypad  
{  
public static void main(String args[])  
{  
Scanner sc=new Scanner(System.in);  
String str;  
str=sc.next();  
char[] ch = new char[str.length()];  
  
for (int i = 0; i<str.length(); i++)  
{
```

```
ch[i] = str.charAt(i);

    switch (ch[i])
    {
        case 'a':
        case 'b':
        case 'c':
System.out.print("2");
        break;
        case 'd':
        case 'e':
        case 'f':
System.out.print("3");
        break;
        case 'g':
        case 'h':
        case 'i':
System.out.print("4");
        break;
        case 'j':
        case 'k':
        case 'l':
System.out.print("5");
        break;
        case 'm':
        case 'n':
        case 'o':
System.out.print("6");
        break;
        case 'p':
        case 'q':
        case 'r':
        case 's':
```

```
System.out.print("7");
    break;
    case 't':
    case 'u':
    case 'v':
System.out.print("8");
    break;
    case 'w':
    case 'x':
    case 'y':
    case 'z':
System.out.print("9");
    break;
    default:
System.exit(0);
    break;
}}}}
```

### **EXPECTED OUTPUT:**

#### **Sample Input:**

**amazon**

#### **Sample Output:**

**262966**



## EXPERIMENT 5

5. Define a **MyRectangle** class with four public data members representing the x- and y-coordinates of the bottom-left vertex and top-right vertex of a rectangle, whose sides are parallel to the x- or y-axis.

For example, the statement `new MyRectangle(20,80,30,90)` creates a rectangle with bottom-left vertex at position (20,80), and top-right vertex at (30,90).

- (a) Write an `area()` method, which computes the area of a rectangle.
- (b) Write the `overlap(MyRectangle rect)` method. This method returns a rectangle which is the overlapped region of two rectangles. In the event that there is no overlap, it should return a rectangle with both bottom-left vertex and top-right vertex at position (0,0).
- (c) Using the `overlap(MyRectangle rect)` method written above, write the `overlapAll(MyRectangle[] rectangles)` method which returns the overlapped region of all the rectangles in the array. You may assume that there is at least one element in the array. Your method should be efficient in that the moment it finds that the overlapped region is empty, it should return a rectangle with both vertices at (0,0) immediately.
- (d) Write `MySquare.java`, `MySquare` extends `MyRectangle`.

A square is defined by its bottom-left vertex and the length of its side. Complete the `super( . . )` statement in the constructor.

(e) Below is output of `MySquare.java` program when the user enters: 10 30 5.

**Class MySquare: [(10,30); (15,35)]**  
**Area = 25**

Override the `toString()` method in `MyRectangle` in order to get such output.

```
class MyRectangle
{
    public int v1x;
    public int v1y;
    public int v2x;
    public int v2y;

    public MyRectangle(int ver1x,int ver1y,int ver2x,int ver2y){
        v1x=ver1x;
        v1y=ver1y;
        v2x=ver2x;
        v2y=ver2y;
    }
    public int area(){
        int area =Math.abs((v1x-v2x)*(v2y-v1y));
        return area;
    }
}
```

```

public MyRectangleoverlap(MyRectangle rect){

    int x1=Math.max(v1x ,rect.v1x);
    int y1=Math.max(v1y ,rect.v1y);
    int x2=Math.min(v2x ,rect.v2x);
    int y2=Math.min(v2y ,rect.v2y);

    if(x1>x2 || y1>y2){
        MyRectangle m1=new MyRectangle(0,0,0,0);
        System.out.println("why");
        return m1;
    }
    else{
        MyRectangle m2=new MyRectangle(x1,y1,x2,y2);

        return m2;

    }

}
}

}

class MySquare extends MyRectangle {
    public int xcor;
    public int ycor;
    public int length;
    public MySquare(int xcord,intycord,intleng){
        xcor=xcord;
        ycor=ycord;
        length=leng;

    }
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int x=sc.nextInt();
        int y=sc.nextInt();
        int len=sc.nextInt();

        MySquare square =new MySquare(x,y,len);
        System.out.println(square.area());

    }
    public int area(){
        int area = len*len;

```

```
        return area;
    }
    public void toString(){
        System.out.println("Class MySquare:[(x,y);(x+len,y+len)]");
        System.out.println(square.area());
    }
}
```

**EXPECTED OUTPUT:**

**Sample Input: 10 30 5**

**Sample Output:25**

## EXPERIMENT 6

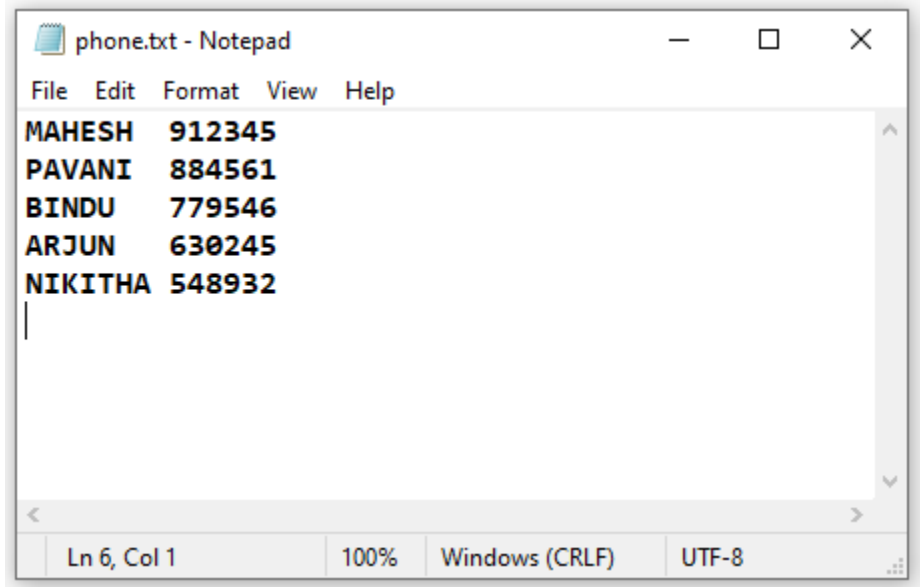
**6. a.** Write a java program that loads names and phone numbers from the text file into Hash Table where data is organized as one line per record and each field in record are separated by a tab(\t). It takes a name or phone number as input and prints the corresponding other value from hash table.

```
import java.util.*;
import java.io.*;

public class PhoneRecord
{ public static void main(String[] args) throws Exception
{ Scanner sc = new Scanner(System.in);
  FileReader fr = new FileReader("phone.txt");
  BufferedReader br = new BufferedReader(fr);
  Hashtable ht = new Hashtable();
  String line, name, mobile;
  while((line = br.readLine()) != null)
  { String s[] = line.split("\\s+");
    ht.put(s[0], s[1]); }
  System.out.println(ht);
  System.out.print("Enter name: ");
  name = sc.next();
  if (ht.containsKey(name))
  { System.out.println("Mobile number is " + ht.get(name)); }
  else { System.out.println("Not Found"); }
  System.out.print("Enter mobile number: ");
  mobile = sc.next();
  if (ht.containsValue(mobile))
  {
    Set<Map.Entry> s1 = ht.entrySet();
    for (Map.Entry e : s1)
    { if (mobile.equals(e.getValue()))
      { System.out.println("Name is " + e.getKey()); } } }
    else { System.out.println("Not Found"); }
  br.close(); } }
```

## Expected Output:

### Phone.txt



## EXPECTED OUTPUT

```
{ NIKITHA=548932, BINDU=779546, MAHESH=912345, ARJUN=630245, PAVANI=884561 }  
Enter name: MAHESH  
Mobile number is 912345  
Enter mobile number: 779546  
Name is BINDU
```

**b. You have created a web-based survey of favorite programming languages and are capturing the results into a text file named “logfile”. The structure of the text file is:**

```
Total # Entries  
Vote for Entry 1  
IP Address for Entry 1  
Timestamp in seconds for Entry 1  
Vote for Entry 2  
IP Address for Entry 2  
Timestamp in seconds for Entry 2  
...
```

**For example, here is a sample logfile of six entries:**

```

6
PHP
137.229.156.12
1000002
C#
137.229.156.18
1000005
PHP
137.229.156.12
1000006
Prolog
156.213.38.31
1000010
PHP
128.120.56.214
1000020
PHP
137.229.156.12
1000022

```

**The logfile is ordered by increasing timestamp. You are concerned that some people are voting multiple times for the same item. To somewhat address this problem, throw out any new votes for the same item that come from the same IP address within 20 seconds.**

**In the above example, the second and last votes for PHP would be thrown out because they are for the same item from the same IP address and occur within 20 seconds of other PHP votes from the same IP address. However, the PHP vote from 128.120.56.214 would be retained since there is not another PHP vote from this IP address.**

**Write a Java program to count the votes from the logfile, throwing out duplicate votes using the rules above. Display the votes in a table, as shown below for the example:**

**PHP 2**

**C# 1**

**Prolog 1**

```

import java.util.*;
import java.io.*;
import java.util.concurrent.CopyOnWriteArrayList;
class Data{
    String plang,ip,timestamp;
    void setData( Stringplang, String ip,String timestamp)
    {
        this.plang=plang;
        this.ip=ip;
        this.timestamp=timestamp;
    }
    String getPlang()
    {

```

```

return this.plang;
}
String getIP()
{
return this.ip;
}
String getTimestamp()
{
return this.timestamp;
}
}
public class DuplicateVotess
{
public static void main(String[] args) throws Exception
{
Scanner sc = new Scanner(System.in);
FileReader fr = new FileReader("logfile.txt");
BufferedReader br = new BufferedReader(fr);
String line;
String[] data;
//ArrayList<Data> al = new ArrayList<Data>();
List<Data> al = new CopyOnWriteArrayList<Data>();
while((line = br.readLine()) != null) {
data = line.split("\t");
Data d = new Data();
d.setData(data[0], data[1], data[2]);
al.add(d);
}

for(Data i : al)
{
System.out.println(i.getPlang() + ", " + i.getIP() + ", " + i.getTimestamp());
}
}
}

```

```

    }

    for (Data d: al) {
        if (d.getPlang().equals(d.plang)) {
            if (d.getIP().equals(d.ip))
            {
                al.remove(d);
            }
        }
    }

    for(Data i : al){
        System.out.println(i.getPlang()+" "+i.getIP()+" "+i.getTimestamp());

    }

} }

```

**EXPECTED OUTPUT:**

**PHP 2**

**C# 1**

**Prolog 1**



## EXPERIMENT 7

7. a. Write a Java Class to implement a method Addition() that returns a new Array where each array element at the index k corresponds to the sum of elements of the array (src) starting at index 0 and including element at the index 'k'. For example, for array [2,3,5], the method will return array [2,5,10]. For an array of size '0' or a null parameter, the method will throw exception IllegalArgumentExceptionWith the message "Invalid Argument".

```
import java.util.Arrays;
import java.util.Scanner;
class RunningSum7a
{
    public static void main(String[] args)
    {
        System.out.println("Enter Array Size: ");
        Scanner sc=new Scanner (System.in);
        int n=sc.nextInt();
        int arr[];
        try
        {
            if (n>0)
                arr=new int[n];
            else
                throw new IllegalArgumentException() ;
            System.out.println("Enter Array Elements: ");
            for (int i=0;i<n ;i++ )
            {
                arr[i]=sc.nextInt();
            }

            RunningSumrs=new RunningSum();
            int resultarr[]=rs.addition(arr);
            System.out.println(Arrays.toString(resultarr));

        }
        catch (IllegalArgumentException e)
        {
            System.out.println("Invalid Argument");
        }

    }
}

int[] addition(int[] a) {
    int sum = 0;
    for (int i = 0; i<a.length; i++) {
        a[i] += sum;
    }
}
```

```

        sum = a[i];
    }
    return a;
}
}

```

**Expected output:**

**Sample Input & Output:**

**Case 1:**

**Enter Array Size:**

5

**Enter Array Elements:**

3 5 7 10 12

[3, 8, 15, 25, 37]

**Case 2:**

**Enter Array Size:**

0

**Invalid Argument**

**b. Write a Java Code to implement a multithreaded version of FizzBuzz with four threads.**

**If the number is divisible by 3, output "fizz".**

**If the number is divisible by 5, output "buzz".**

**If the number is divisible by both 3 and 5, output "fizzbuzz".**

**If the number is not divisible by both 3 and 5 print the number**

**For instance if n is 15 , we will have the output as “ 1, 2, fizz, 4, buzz, fizz, 7, 8, fizz,buzz, 11, fizz, 13, 14, fizzbuzz “**

**1. Thread A will call fizz() to check for divisibility of 3 and outputs fizz .**

**2. Thread B will call buzz() to check for divisibility of 5 and outputs buzz .**

**3. Thread C will call fizzbuzz() to check for divisibility of 3 and 5 and outputs fizzbuzz .**

**4. Thread D will call number() which should only output the numbers.**

```

import java.util.Scanner;
class ThreadFizzBuzz
{
    public static void main(String[] args) throws Exception
    {

```

```

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter n value: ");
        int n=sc.nextInt();
        for(int i=1;i<=n ;i++)
        {
            if (i%15 == 0)
            {
                C t1=new C();
                t1.start();
                t1.join();
            }
            else if (i%5==0)
            {
                B t2=new B();
                t2.start();
                t2.join();
            }
            else if (i%3 == 0)
            {
                A t3=new A();
                t3.start();
                t3.join();
            }
            else
                System.out.print(i+", ");
        }
    }
}

class A extends Thread
{
    public void run()
    {
        System.out.print("fizz, ");
    }
}

class B extends Thread
{
    public void run()
    {
        System.out.print("buzz" +", ");
    }
}

```

```
}  
  
class C extends Thread  
{  
    public void run()  
    {  
        System.out.print("fizzbuzz" + ", ");  
    }  
}
```

**Expected Output:**

**Sample Input:**

**Enter n value:**

**20**

**Sample Output:**

**1, 2, fizz, 4, buzz, fizz, 7, 8, fizz, buzz, 11, fizz, 13, 14, fizzbuzz, 16, 17, fizz, 19, buzz,**

## EXPERIMENT 8

**8.a. Write a java program to store the employee details in an ArrayList and display the employee details in ascending order of their experience. Create 'Employee' class with two instance variables Employee name and Employee experience (no. of years).**

```
import java.util.*;
class EmpSorting8a
{
    public static void main(String[] args)
    {
        ArrayList<Employee>ar = new ArrayList<Employee>();
        ar.add(new Employee("John",5));
        ar.add(new Employee("Nic",3));
        ar.add(new Employee("Roma",9));
        System.out.println("Unsorted Emp List");
        for (int i = 0; i<ar.size(); i++)
            System.out.println(ar.get(i).toString());

        Collections.sort(ar, new Sortbyexp());

        System.out.println("Sorted by Experience");
        for (int i = 0; i<ar.size(); i++)
            System.out.println(ar.get(i).toString());
    }
}
class Employee
{
    String empname;
    int exp;
    Employee(String empname, int exp)
    {
        this.empname= empname;
        this.exp=exp;
    }
    public String toString()
    {
        return this.empname + " " + this.exp ;
    }
}
class Sortbyexp implements Comparator<Employee> {
    public int compare(Employee a, Employee b)
    {
        return a.exp - b.exp;
    }
}
```

**EXPECTED OUTPUT:****Unsorted Emp List****John 5****Nic 3****Roma 9****Sorted by Experience****Nic 3****John 5****Roma 9**

**b.** Write a program to find the most common words in the list of words given in sorted order based on occurrence from largest to smallest. If any of words are having same occurrence then consider the smallest character order comes first.

Input format: First line contains the list of words and next line contains a number (k) which represent the top most words to display.

Output format: display the k top most words.

```
import java.util.*;
class CommonWordSort8b{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        String s=sc.nextLine();
        String arr[]=s.split(" ");
        Map <String,Integer>ch=new LinkedHashMap<String,Integer>();

        for(int i=0;i<arr.length;i++){
            String x=arr[i];
            if(ch.containsKey(x)){
                int c=ch.get(x);
                c++;
                ch.put(x,c);
            }
            else{
                ch.put(x,1);
            }
        }

        ArrayList<Integer> al=new ArrayList<Integer>();
        Set<String> ss=ch.keySet();
        for(String i:ss){
            if(al.contains(ch.get(i))){ }
            else{
                al.add(ch.get(i));
            }
        }
    }
}
```

```

Collections.sort(al);
for(int i=al.size()-1;i>=0;i--){
    int temp= al.get(i);
        ArrayList<String>tal=new ArrayList<>();
        for(Map.Entry<String, Integer> l1 : ch.entrySet())
        {
            if(l1.getValue() == temp)
                tal.add(l1.getKey());
        }
        Collections.sort(tal);
        for( String s1:tal)
            System.out.print(s1+" ");
    }
}

```

**Expected output:**  
lets do one to do how to do  
do to how lets one

## EXPERIMENT 9

**9. a. Write a java Program to write a method fCount which takes a string as a parameter. The Method fCount should return the Map which has the frequency count of the given word. For example if the string passed is “hello” the map should return {h-1,e-1,l-2,o-1}. The order of the characters should be same as in the string.**

```
import java.util.*;
class Test9A
{
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a string : ");
        String str = input.next();

        Map<Character, Integer> result = new HashMap<Character, Integer>();
        result = fCount(str);

        for(int i = 0; i<str.length(); i++)
        {
            if(result.get(str.charAt(i)) != 0)
            {
                System.out.print(str.charAt(i)+"-"+result.get(str.charAt(i)) + " ");
                result.put(str.charAt(i), 0);
            }
        }

        input.close();
    }
    public static Map<Character, Integer>fCount(String str)
    {
        Map<Character, Integer> map = new HashMap<Character, Integer>();

        for(int i = 0; i<str.length(); i++)
        {
            if(map.containsKey(str.charAt(i)))
            {
                map.put(str.charAt(i), map.get(str.charAt(i)) + 1);
            }
            else
            {
                map.put(str.charAt(i), 1);
            }
        }

        return map;
    }
}
```



```
}  
}
```

## EXPECTED OUTPUT

Run 1 :

Enter a string : hello

h-1 e-1 l-2 o-1

Run 2 :

Enter a string :kmit

k-1 m-1 i-1 t-1

b. When working with HashMaps, sometimes cases arise where we wish to determine if two HashMaps have any key/value pairs in common. For example, we might have the following two Hashmaps (named hashmap1 and hashmap2, respectively) that map from String to String (i.e., their type is HashMap) and we want to count how many key/value pairs they have in common.

hashmap1		hashmap2	
Key	Value	Key	Value
Alice	Healthy	Mary	Ecstatic
Mary	Ecstatic	Felix	Healthy
Bob	Happy	Ricardo	Superb
Chuck	Fine	Tam	Fine
Felix	Sick	Bob	Happy

In the example above, these two HashMaps have two key/value pairs in common, namely: "Mary"/"Ecstatic" and "Bob"/"Happy". Note that although the key "Felix" is in both HashMaps, the associated value with this key is different in the two maps (hence this does not count as a key/value pair that is common to both HashMaps). Similarly, just having the same value without the same key (such as the value "Fine" which is mapped to by different keys in the two different HashMaps) would also not count as a common key/value pair between the two HashMaps.

Your job is to write a method:

```
public int commonKeyValuePairs(HashMap<String,String> map1,  
                                HashMap<String,String> map2)
```

that is passed two objects of type `HashMap<String,String>` and returns the number of common key/value pairs between the two HashMaps.

```
import java.util.*;  
class Test9B  
{
```

```

public static void main(String args[])
{
    Scanner input = new Scanner(System.in);

    HashMap<String, String>mapOne = new HashMap<String, String>();
    HashMap<String, String>mapTwo = new HashMap<String, String>();

    System.out.print("Enter how many pairs in HashMap 1 : ");
    int mapOneSize = input.nextInt();
    input.nextLine();

    System.out.println("Enter your " + mapOneSize + " pair of String ");
    for(int i = 0; i<mapOneSize; i++)
    {
        System.out.println("Enter Pair " + (i+1));
        String pairs[] = input.nextLine().split(" ");
        mapOne.put(pairs[0],pairs[1]);
    }

    System.out.print("Enter how many pairs in HashMap 2 : ");
    int mapTwoSize = input.nextInt();
    input.nextLine();
    System.out.println("Enter your " + mapTwoSize + " pair of String ");
    for(int i = 0; i<mapTwoSize; i++)
    {
        System.out.println("Enter Pair " + (i+1));
        String pairs[] = input.nextLine().split(" ");
        mapTwo.put(pairs[0],pairs[1]);
    }
    int result = commonKeyValuePairs(mapOne,mapTwo);

    System.out.println("The number of key/value pairs are : " + result);

    input.close();
}

public static int commonKeyValuePairs(HashMap<String, String> map1 ,
HashMap<String, String> map2)
{
    int count = 0;
    for(Map.Entry m1:map1.entrySet())
    {
        for(Map.Entry m2:map2.entrySet())
        {
            if(m1.getKey().equals(m2.getKey())&& m1.getValue().equals(m2.getValue()))
            {
                count++;
            }
        }
    }
}

```

```

        }
    }
}
return count;
}
}

```

### EXPECTED OUTPUT

Enter how many pairs in HashMap 1 : 5

Enter your 5 pair of String

Enter Pair 1

Alice Healthy

Enter Pair 2

Mary Ecstatic

Enter Pair 3

Bob Happy

Enter Pair 4

Chuck Fine

Enter Pair 5

Felix Sick

Enter how many pairs in HashMap 2 : 5

Enter your 5 pair of String

Enter Pair 1

Mary Ecstatic

Enter Pair 2

Felix Healthy

Enter Pair 3

Ricardo Superb

Enter Pair 4

Tam Fine

Enter Pair 5

Bob Happy

The number of key/value pairs are : 2

## EXPERIMENT 10

### 10. a.Demonstrate a JDBC Program on Employee Schema given below

Read a department number from the user and display those employee names in ascending order who are working in the department.

Note - Display records based on employee names in ascending order

The DB Credentials

Name of the DB - test;

Name of the table- emp;

JDBC\_DRIVER = "com.mysql.jdbc.Driver";

DB\_URL = "jdbc:mysql://localhost/test";

Username-student

Password-student

emp schema;

```
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | int(4) | NO | PRI | NULL | |
| ename | varchar(50) | NO | | NULL | |
| job | varchar(50) | NO | | NULL | |
| mgr | int(4) | YES | | NULL | |
| hiredate | date | YES | | NULL | |
| sal | decimal(10,2) | YES | | NULL | |
| comm | decimal(10,2) | YES | | NULL | |
| deptno | int(2) | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
```

Sample Database :

mysql> select \* from emp;

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | ename   | job       | mgr | hiredate | sal      | comm      | deptno |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 401 | Albert | Sr. Developer | 4001 | 2019-07-01 | 35000.00 | 3000.00 | 4 |
```

402	Robert	Data Analyst	4002	2019-08-01	30000.00	2000.00	4
403	Suresh	Developer	1002	2019-09-01	24000.00	2000.00	4
501	Nitesh	Developer	1001	2019-08-01	25000.00	2000.00	5
502	Mahesh	Developer	1001	2019-09-01	24000.00	2000.00	5
503	Akhilesh	Sr. Developer	1002	2019-08-01	45000.00	4000.00	5
1201	Venkatesh	Sr. Developer	101	2019-07-01	35000.00	3000.00	12
1202	Harish	Data Analyst	102	2019-08-01	30000.00	2000.00	12

+-----+-----+-----+-----+-----+-----+-----+-----+

**8 rows in set (0.00 sec)**

```
import java.sql.*;
import java.util.*;
class Test10A
{
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the Department Number : ");
        int deptNum = input.nextInt();
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/test","student","student");
            Statement stmt=con.createStatement();
            String sql = "select ename from emp where deptno = "+deptNum+" order
by enameasc";
            ResultSets=stmt.executeQuery(sql);
            ResultSetMetaDatarsmd = rs.getMetaData();
            System.out.println("");

            int numberOfColumns = rsmd.getColumnCount();
```

```

        for (int i = 1; i<= numberOfColumns; i++)
        {
            if (i> 1) System.out.print("  ");
            String columnName = rsmd.getColumnName(i);
            System.out.print(columnName);
        }
        System.out.println("");

        while (rs.next())
        {
            for (int i = 1; i<= numberOfColumns; i++)
            {
                if (i> 1)
                    System.out.print(", ");
                String columnValue = rs.getString(i);
                System.out.print(columnValue);
            }
            System.out.println("");
        }

        con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    input.close();
}
}

```

### **Expected Output:**

Enter the Department Number : 5

ename

Akhilesh

Mahesh

Nitesh

**b.** Demonstrate a JDBC program to display all the employee names in ascending order who are working in

"Dallas" location using createStatement()

Schema Given below

emp schema;

Field	Type	Null	Key	Default	Extra
empno	int(4)	NO	PRI	NULL	
ename	varchar(50)	NO		NULL	
job	varchar(50)	NO		NULL	
mgr	int(4)	YES		NULL	
hiredate	date	YES		NULL	
sal	decimal(10,2)	YES		NULL	
comm	decimal(10,2)	YES		NULL	
deptno	int(2)	YES	MUL	NULL	

dept schema

Field	Type	Null	Key	Default	Extra
deptno	int(2)	NO	PRI	NULL	
dname	varchar(50)	NO		NULL	
location	varchar(50)	NO		NULL	

Sample Database :

```
mysql> select * from emp;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| empno | ename   | job       | mgr | hiredate | sal    | comm    | deptno |
+-----+-----+-----+-----+-----+-----+-----+
| 401 | Albert  | Sr. Developer | 4001 | 2019-07-01 | 35000.00 | 3000.00 | 4 |
| 402 | Robert  | Data Analyst  | 4002 | 2019-08-01 | 30000.00 | 2000.00 | 4 |
| 403 | Suresh  | Developer     | 1002 | 2019-09-01 | 24000.00 | 2000.00 | 4 |
| 501 | Nitesh  | Developer     | 1001 | 2019-08-01 | 25000.00 | 2000.00 | 5 |
| 502 | Mahesh  | Developer     | 1001 | 2019-09-01 | 24000.00 | 2000.00 | 5 |
| 503 | Akhilesh | Sr. Developer | 1002 | 2019-08-01 | 45000.00 | 4000.00 | 5 |
| 1201 | Venkatesh | Sr. Developer | 101 | 2019-07-01 | 35000.00 | 3000.00 | 12 |
| 1202 | Harish   | Data Analyst  | 102 | 2019-08-01 | 30000.00 | 2000.00 | 12 |
+-----+-----+-----+-----+-----+-----+-----+
```

8 rows in set (0.00 sec)

```
mysql> select * from dept;
```

```
+-----+-----+-----+
| deptno | dname | location |
+-----+-----+-----+
| 4 | ECE | Collin |
| 5 | CSE | Dallas |
| 12 | IT | Denton |
+-----+-----+-----+
```

3 rows in set (0.00 sec)

```
import java.sql.*;
```

```
class Test10B
```

```
{
    public static void main(String args[])
    {
        try
        {
```



```

        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/test","student","student");
        Statement stmt=con.createStatement();
        String location = "Dallas";
        String sql = "select ename from emp e, dept d wheree.deptno=d.deptno
and location = 'Dallas'";
        ResultSets=stmt.executeQuery(sql);
        ResultSetMetaDatarsmd = rs.getMetaData();
        int columnsNumber = rsmd.getColumnCount();
        System.out.println("");

        int numberOfColumns = rsmd.getColumnCount();

        for (int i = 1; i<= numberOfColumns; i++)
        {
            if (i> 1) System.out.print("  ");
            String columnName = rsmd洗getColumnName(i);
            System.out.print(columnName);
        }
        System.out.println("");
        while (rs.next())
        {
            for (int i = 1; i<= numberOfColumns; i++)
            {
                if (i> 1)
                    System.out.print(", ");
                String columnValue = rs.getString(i);
                System.out.print(columnValue);
            }
            System.out.println("");
        }

```

```

        con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}

```

### Expected Output :

```

ename
Nitesh
Mahesh
Akhilesh

```

c. Demonstrate JDBC program to read three values (dno,dname,dloc) from the user and insert those records into the dept table using Prepared Statement

The Reading of input should be first deptno followed by deptname followed by deptlocation

Dept Schema

```

+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| deptno | int(2) | NO | PRI | NULL | |
| dname | varchar(50) | NO | | NULL | |
| location | varchar(50) | NO | | NULL | |
+-----+-----+-----+-----+-----+

```

Sample Database :

```

mysql> select * from dept;
+-----+-----+-----+
| deptno | dname | location |

```

```

+-----+-----+-----+
|  4 | ECE | Collin |
|  5 | CSE | Dallas |
| 12 | IT  | Denton |
+-----+-----+-----+
3 rows in set (0.00 sec)

```

```

import java.sql.*;
import java.util.*;
class Test10C
{
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the Department No. : ");
        int dno = input.nextInt();
        System.out.print("Enter the Department Name : ");
        String dname = input.next();
        System.out.print("Enter the Department Location : ");
        String dloc = input.next();
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/test","student","student");
            PreparedStatement pstmt = null;
            String query = "INSERT INTO dept(deptno, dname, location)" +
"VALUES (?, ?, ?)";

            pstmt = con.prepareStatement(query);
            pstmt.setInt(1, dno);
            pstmt.setString(2, dname);
            pstmt.setString(3, dloc);

```

```

        int status = pstmt.executeUpdate();
        if(status > 0)
        {
            System.out.println("Record is inserted successfully !!!");
        }

        con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    input.close();
}
}

```

### Expected Output:

Enter the Department No. : 3

Enter the Department Name : Mechanical

Enter the Department Location : Dallas

Record is inserted successfully !!!

After Inserted database would be as below.

mysql> select \* from dept;

```

+-----+-----+-----+
| deptno | dname   | location |
+-----+-----+-----+
| 3 | Mechanical | Dallas |
| 4 | ECE       | Collin  |
| 5 | CSE       | Dallas  |
| 12 | IT        | Denton  |
+-----+-----+-----+

```

4 rows in set (0.00 sec)