

WIP

- 1) Write program to find the factorial value of any number entered by the user using Javascript.

Input: The first line contains an integer T , the total number of test cases. Then T lines follow, each line contains an integer N .

Output: For each test case, display the factorial of the given number N in a new line.

Example: $1 \leq T \leq 1000$
 $0 \leq N \leq 20$

Input	Output
3	6
3	24
4	120
5	

Ans. Factorial of any Number.

```
<!doctype html>
<html>
<head>
<script>
function show() {
    var i, n, fact;
    fact = 1;
    n = Number(document.getElementById("num").value);
    for(i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
}
```

</script>

</head>

<body>

Enter Num: <input id="Num">

<button onclick="show()"> Factorial </button>

<input id="answer">

</body>

</html>

②

In a company an employee is paid as under if basic salary is less than Rs.1500, then HRA=10% of basic salary DA=90% of basic salary. If his salary is either equal to or above Rs.1500 then HRA=Rs,500 and DA=98% of basic salary. If the Employee's salary is input, write a program to find gross salary. using Java script event handlers.

NOTE: Gross salary = Basic salary + HRA + DA.

Input: The first line contains an integer T , total number of test cases. The follow T lines, each line contains an integer salary.

output: For each test-case, output the gross salary of the employee in a new line. your answer will be considered correct if the absolute error is less than

Input	output-
3	2406.00
1203	20383.16
10042	
1312	2624

Ans:

```

<html>
<head>
<title> Gross salary </title>
</head>
<body>
<script>
function show() {
    var n, ans;
    n = Number (document.getElementById("salary").value);
    if (n < 1500) {
        ans = n + ((10/100)*n) + ((90/100)*n);
    }
    else {
        ans = n + 500 + ((98/100)*n);
    }
    document.getElementById("ans").value = ans;
}
</script>
Enter salary: <input id="salary">
<button onclick="show()"> Gross salary </button>
<label id="ans"> 0 </label>
</body>
</html>

```

⑤ write a program to take a character (c) as input and check whether the given character is a vowel or a constant using Java.

NOTE: vowels are 'A', 'E', 'I', 'O', 'U'. Rest all alphabets are called constants.

Input: First line will contain the character c

output: print "vowel" if the given character is a vowel, otherwise print "constant".

constraint: c will be upper case English alphabet.

sample input

Z

O

sample output

constant

vowel

Ans: class char {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

char c = sc.next();

if (c == "A" || c == "E" || c == "I" || c == "O" ||

c == "U") {

System.out.println("vowel");

}

else if (c == "a" || c == "e" || c == "i" || c == "o" ||

c == "u") {

(3)

```
System.out.println("vowel");
```

```
}
```

```
else {
```

```
System.out.println("constant");
```

```
}
```

```
}
```

```
}
```

Output:

N

constant

A

vowel

(4)

Describe in detail Multithreading Concept. illustrate with the java code.

Multithreading in java allows concurrent execution of two (or) more parts of a program for maximum utilization of cpu. Each part of such program is called a thread, so, threads are light weight processes within a process.

Program:

```
class Multithreading Demo extends Thread {
```

```
public void run() {
```

```
try {
```

```
System.out.println("Thread" + Thread.currentThread().  
getId() + " is running");
```

```
}
```

```
catch (Exception e) {
```

```
System.out.println("Exception is caught");
```

```
}
```

```
}
```

```
}
```

```
Public class Multithread {
```

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

```
        Int n=4;
```

```
        for (int i=0; i<n; i++) {
```

```
            Multithreading Demo . object = new multithreading Demo();
```

```
            object . start();
```

```
        }
```

```
    }
```

```
}
```

output : Thread 4 is running

Thread 5 is running

Thread 6 is running

Thread 7 is running.

5

connect to the database using any of the JDBC driver

insert & display 2 records with Bus ticket reservation

information like ticket no (8 digit no), source, destination,

Date of journey, Depart no, Arrive on (Time 24 Hr (lock)),

set no, name, Age.

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

```
Public class Data {
```

```

Public static void main (String [] args) {
    Connection con=null;
    Statement st=null;

    try {
        Class.forName ("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection ("jdbc:mysql://localhost/
            bus", "root", "root");
        System.out.println ("Connection established");
        st=con.createStatement();
        st.executeUpdate ("Insert into booking values
            (131, 'hyderabad', 'chennai', '2021-01-12',
            '19:30', 4, 'person 1', 38));
        st.executeUpdate ("Insert into booking values (130, 'vijayawada',
            'hyderabad', '2021-01-20', '19:30', 5, 'person 2', 40));
        ResultSet rs=st.executeQuery ("select * from Booking");
        while (rs.next()) {
            System.out.println ("Ticket no:" + rs.getInt ("ticket
                number"));
            System.out.println ("Source:" + rs.getString ("source"));
            System.out.println ("Destination:" + rs.getString ("destination"));
        }
        con.close();
        st.close();
    }
}

```

}

catch (Exception e) {

e.printStackTrace();

}

}

}

output:

connection established

Ticket no: 131

source: hyderabad

destination: chennai

Ticket no: 130

source: vijayawada

destination: hyderabad