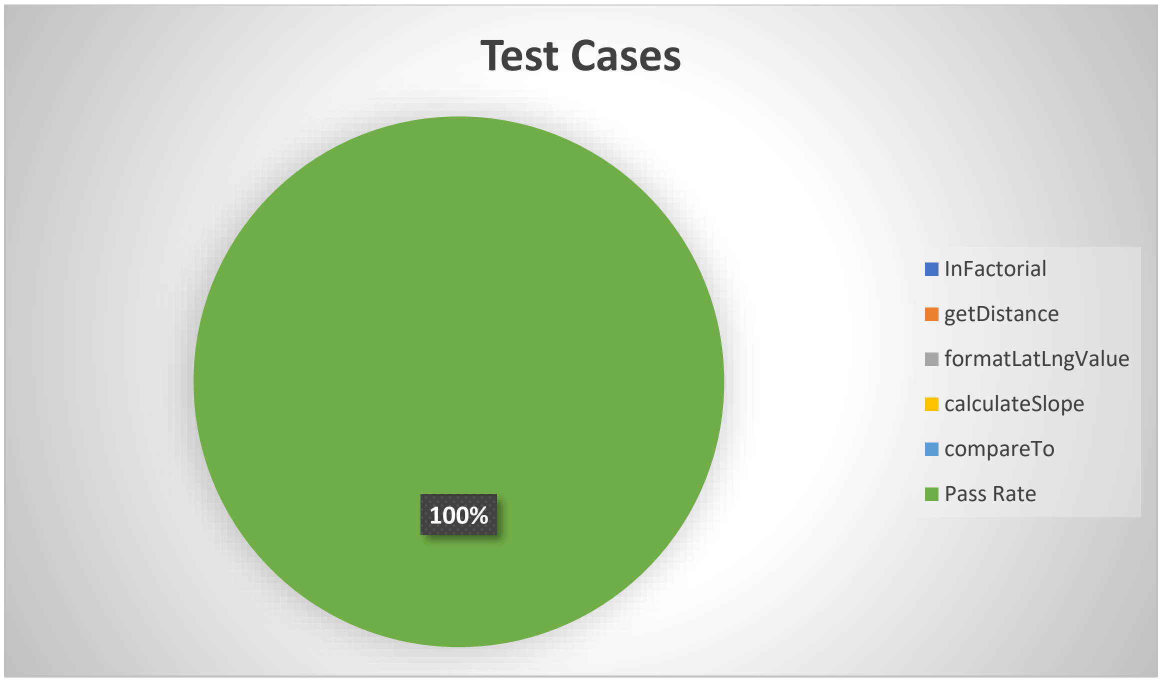
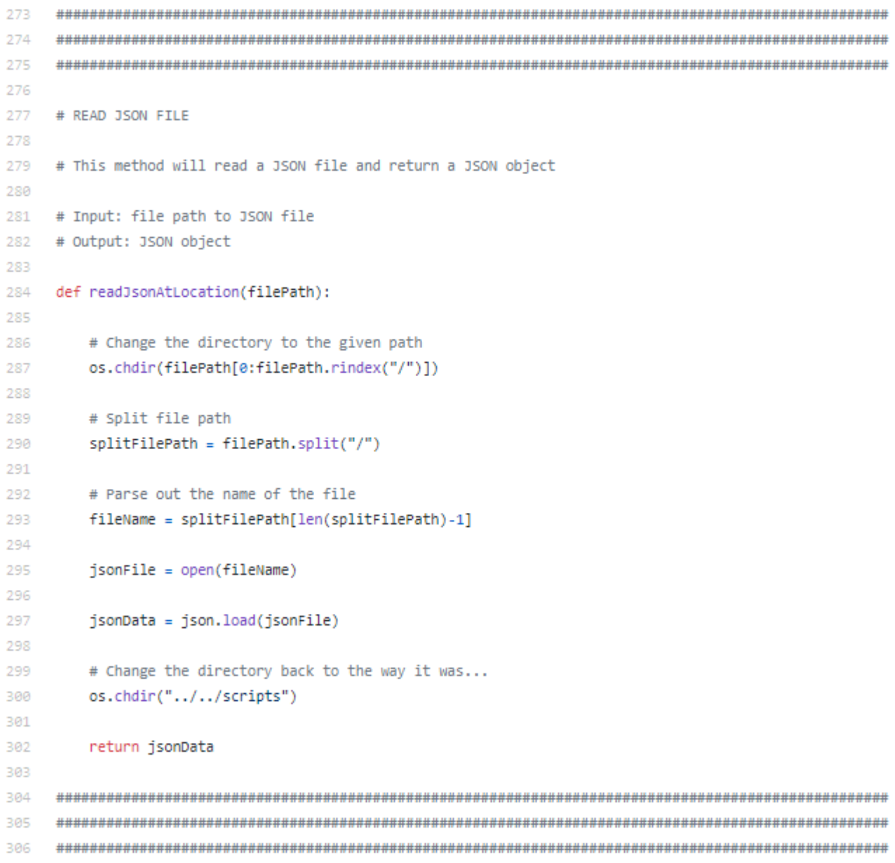


READ JSON FILE

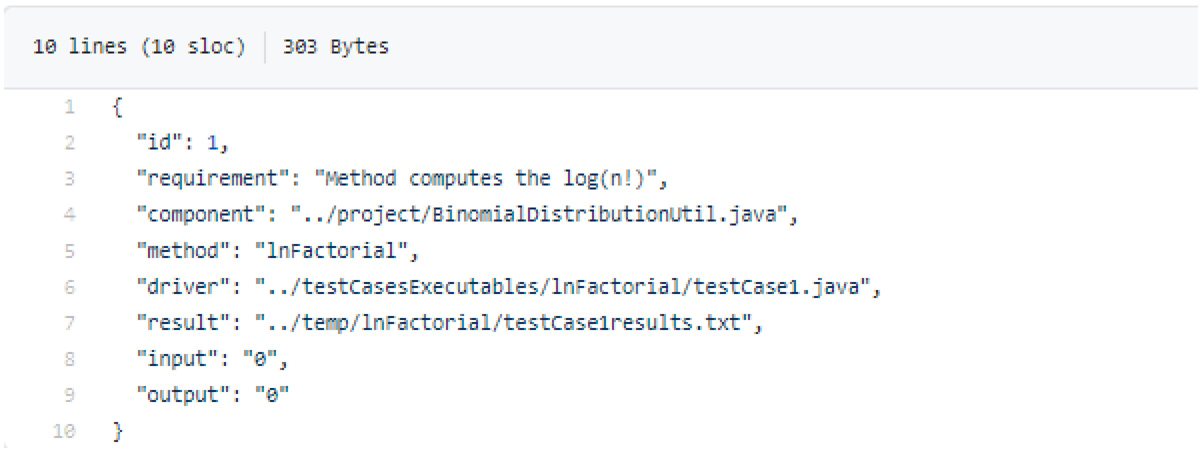


```
####################################################################################################
####################################################################################################
####################################################################################################

# READ JSON FILE

# This method will read a JSON file and return a JSON object

# Input: file path to JSON file
# Output: JSON object

def readJsonAtLocation(filePath):

    # Change the directory to the given path
    os.chdir(filePath[0:filePath.rindex("/")])

    # Split file path
    splitFilePath = filePath.split("/")

    # Parse out the name of the file
    fileName = splitFilePath[len(splitFilePath)-1]

    jsonFile = open(fileName)

    jsonData = json.load(jsonFile)

    # Change the directory back to the way it was...
    os.chdir("../../scripts")

    return jsonData

####################################################################################################
####################################################################################################
####################################################################################################
```

Team Term Project

The code provided above reads a JSON file which uses human-readable text to store and transmit data objects consisting of attribute value pairs and array data types.

LNFACTORIAL JSON FILE

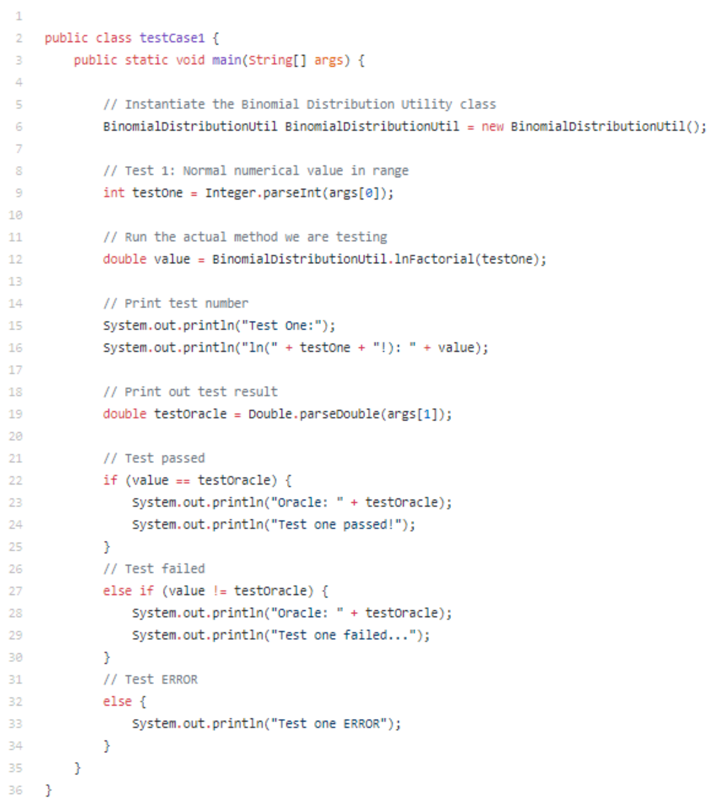


10 lines (10 sloc) | 303 Bytes

```json
{
  "id": 1,
  "requirement": "Method computes the log(n!)",
  "component": "../project/BinomialDistributionUtil.java",
  "method": "lnFactorial",
  "driver": "../testCasesExecutables/lnFactorial/testCase1.java",
  "result": "../temp/lnFactorial/testCase1results.txt",
  "input": "0",
  "output": "0"
}
```

It uses the JSON file to run and execute each test case.

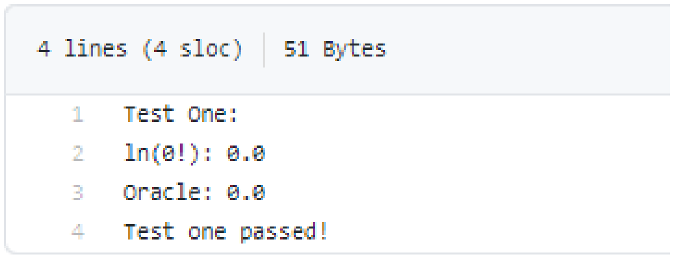
TESTCASE EXECUTABLE



```java

public class testCase1 {
    public static void main(String[] args) {

        // Instantiate the Binomial Distribution Utility class
        BinomialDistributionUtil BinomialDistributionUtil = new BinomialDistributionUtil();

        // Test 1: Normal numerical value in range
        int testOne = Integer.parseInt(args[0]);

        // Run the actual method we are testing
        double value = BinomialDistributionUtil.lnFactorial(testOne);

        // Print test number
        System.out.println("Test One:");
        System.out.println("ln(" + testOne + "!): " + value);

        // Print out test result
        double testOracle = Double.parseDouble(args[1]);

        // Test passed
        if (value == testOracle) {
            System.out.println("Oracle: " + testOracle);
            System.out.println("Test one passed!");
        }
        // Test failed
        else if (value != testOracle) {
            System.out.println("Oracle: " + testOracle);
            System.out.println("Test one failed...");
        }
        // Test ERROR
        else {
            System.out.println("Test one ERROR");
        }
    }
}
```

The results will be collected and compared with the expected results.

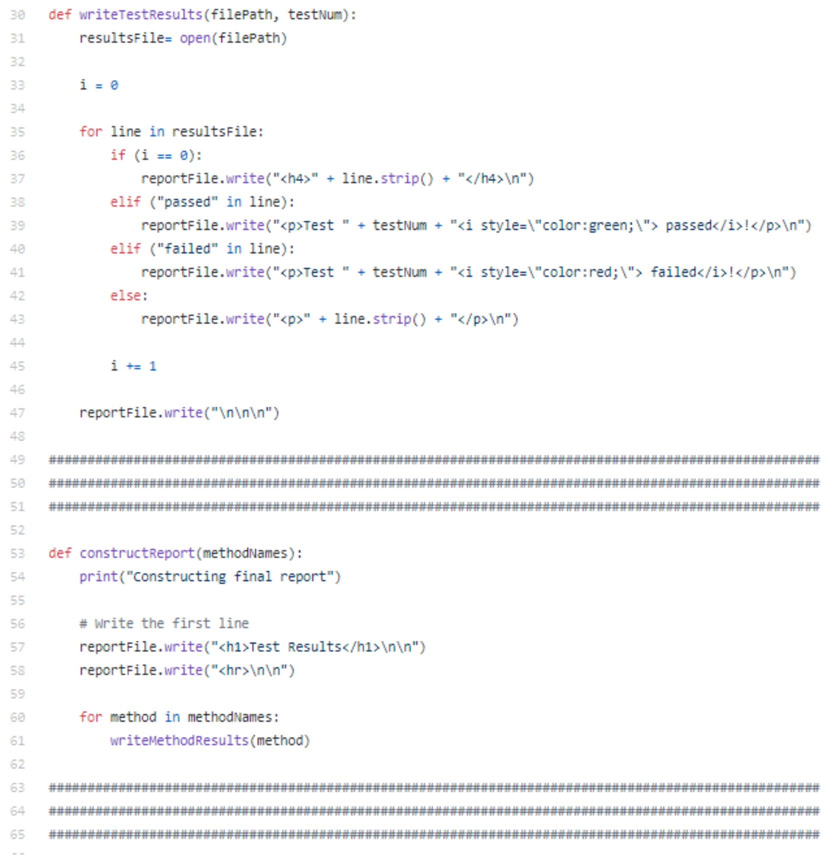
TESTCASE RSULTS



```
4 lines (4 sloc) | 51 Bytes

Test One:
ln(0!): 0.0
Oracle: 0.0
Test one passed!
```

After the test cases are ran the constructReport() method is called which combs though the temporary results files and constructs a final report as a HTML document.

constructReport() method

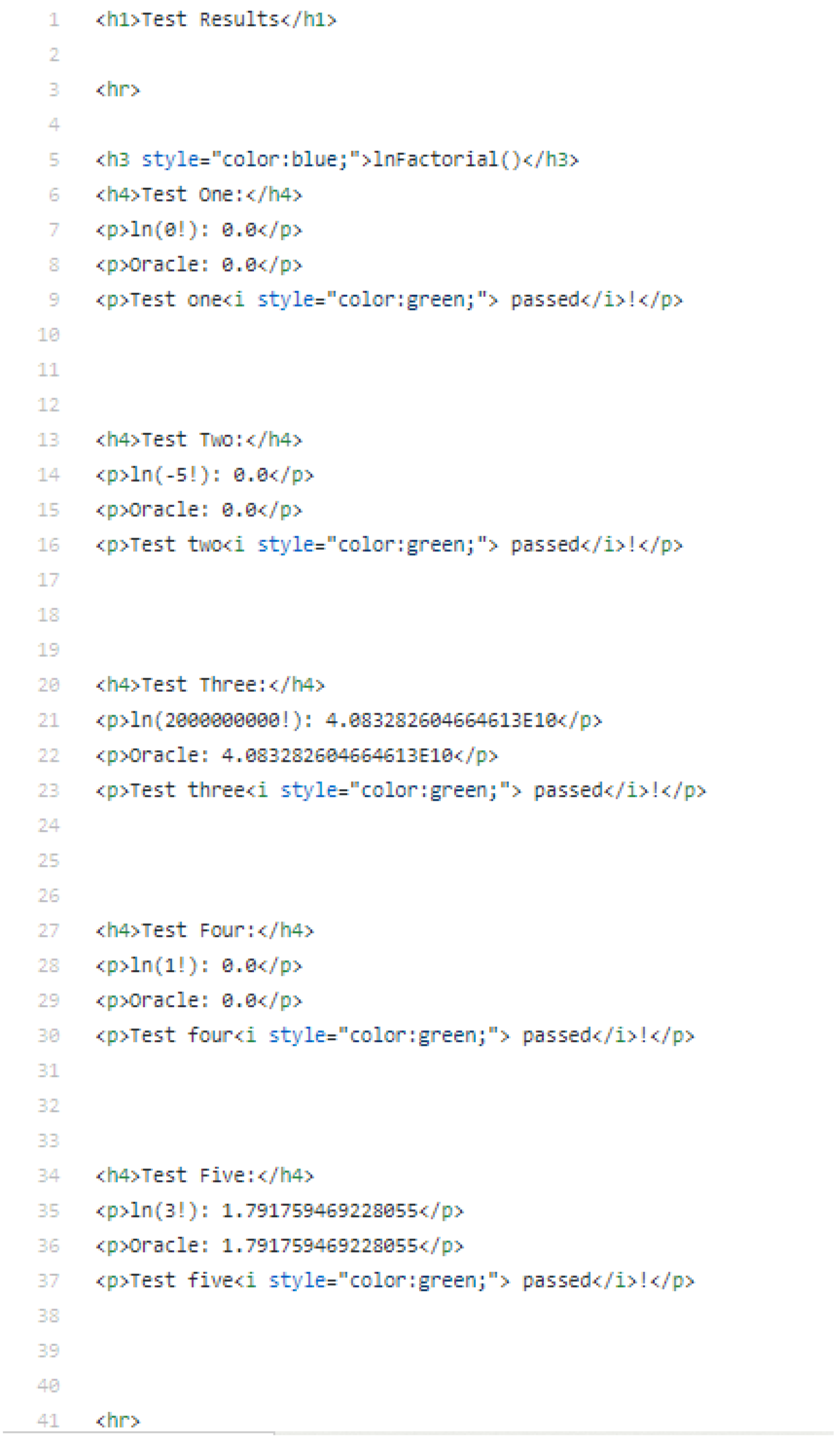


```
def writeTestResults(filePath, testNum):
    resultsFile= open(filePath)

    i = 0

    for line in resultsFile:
        if (i == 0):
            reportFile.write("<h4>" + line.strip() + "</h4>\n")
        elif ("passed" in line):
            reportFile.write("<p>Test " + testNum + "<i style=\"color:green;\"> passed</i>!</p>\n")
        elif ("failed" in line):
            reportFile.write("<p>Test " + testNum + "<i style=\"color:red;\"> failed</i>!</p>\n")
        else:
            reportFile.write("<p>" + line.strip() + "</p>\n")

        i += 1

    reportFile.write("\n\n\n")

####################################################################################################
####################################################################################################
####################################################################################################

def constructReport(methodNames):
    print("Constructing final report")

    # Write the first line
    reportFile.write("<h1>Test Results</h1>\n\n")
    reportFile.write("<hr>\n\n")

    for method in methodNames:
        writeMethodResults(method)

####################################################################################################
####################################################################################################
####################################################################################################
```

Team Term Project

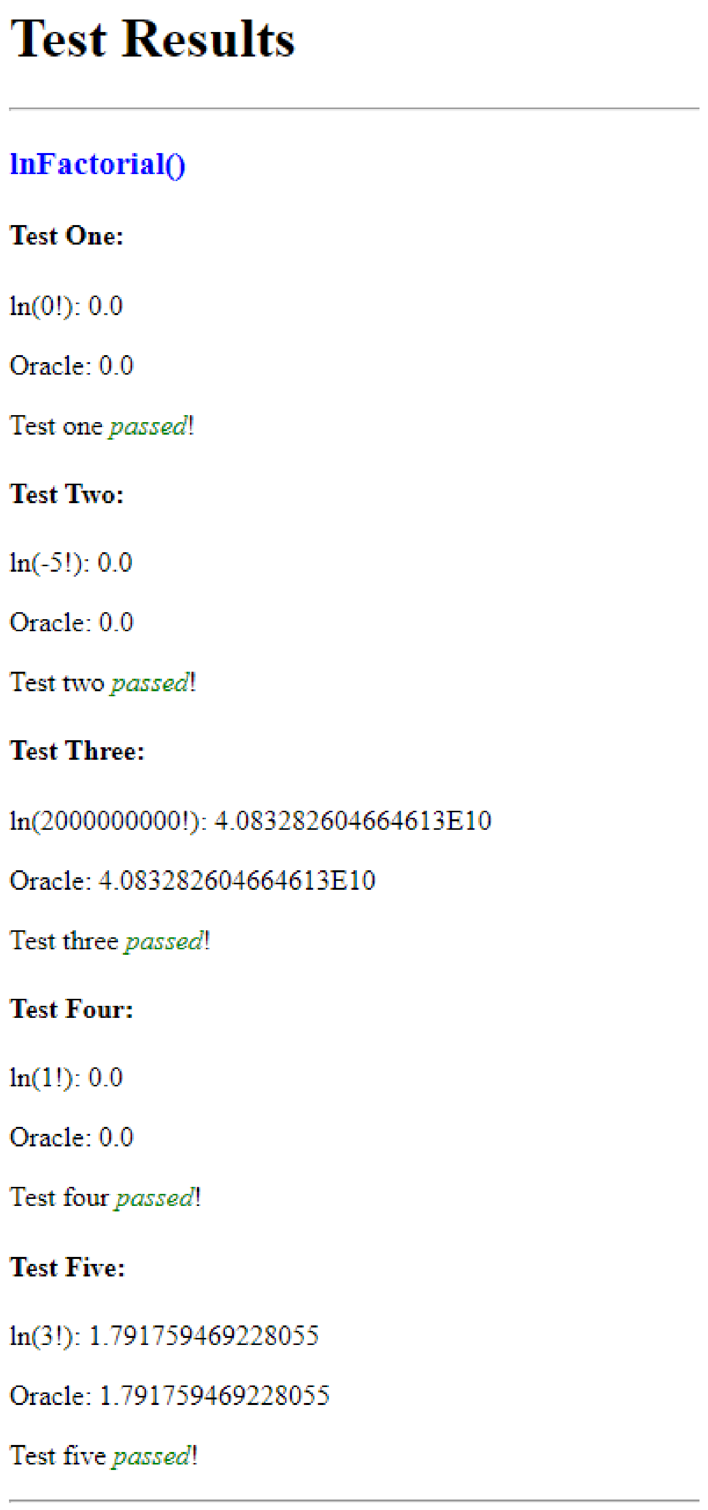
The following final report is constructed after all the test cases are ran

FINAL TEST REPORT



```html
<h1>Test Results</h1>

<hr>

<h3 style="color:blue;">lnFactorial()</h3>
<h4>Test One:</h4>
<p>ln(0!): 0.0</p>
<p>Oracle: 0.0</p>
<p>Test one<i style="color:green;"> passed</i>!</p>



<h4>Test Two:</h4>
<p>ln(-5!): 0.0</p>
<p>Oracle: 0.0</p>
<p>Test two<i style="color:green;"> passed</i>!</p>



<h4>Test Three:</h4>
<p>ln(2000000000!): 4.083282604664613E10</p>
<p>Oracle: 4.083282604664613E10</p>
<p>Test three<i style="color:green;"> passed</i>!</p>



<h4>Test Four:</h4>
<p>ln(1!): 0.0</p>
<p>Oracle: 0.0</p>
<p>Test four<i style="color:green;"> passed</i>!</p>



<h4>Test Five:</h4>
<p>ln(3!): 1.791759469228055</p>
<p>Oracle: 1.791759469228055</p>
<p>Test five<i style="color:green;"> passed</i>!</p>



<hr>
```

Final Output



Test Results

lnFactorial()

Test One:

ln(0!): 0.0

Oracle: 0.0

Test one *passed*!

Test Two:

ln(-5!): 0.0

Oracle: 0.0

Test two *passed*!

Test Three:

ln(2000000000!): 4.083282604664613E10

Oracle: 4.083282604664613E10

Test three *passed*!

Test Four:

ln(1!): 0.0

Oracle: 0.0

Test four *passed*!

Test Five:

ln(3!): 1.791759469228055

Oracle: 1.791759469228055

Test five *passed*!