**Long Project - 2**

**Group 27**

**Members:**

* Gayathri Balakumar
* Susindaran Elangovan
* Vidya Gopalan
* Saikrishna Kanukuntla

**Folder cs6301/g27 contents:**

1. Num.java – **[RUN]**

Store and perform arithmetic operations on arbitrarily large numbers.

**Methods**:

* + 1. add( Num, Num)
    2. subtract( Num, Num )
    3. product( Num, Num )
    4. power( Num, long )
    5. power( Num, Num )
    6. divide( Num, Num )
    7. mod( Num, Num )
    8. squareRoot( Num )
    9. factorial( Num )

**Run**: (Same pattern for all the runnable programs in this package)

javac cs6301/g27/Num.java

java cs6301.g27.Num [inputfile | base] [base]

**Example**:

java cs6301.g27.Num 1001

java cs6301.g27.Num input.txt

java cs6301.g27.Num input.txt 1001

1. Level3Driver.java – **[RUN]**

Accept input in the form of postfix arithmetic expressions and evaluate them using Num class.

1. Level4Driver.java – **[RUN]**

Accept input in the form of a program with statements of either infix/postfix expressions with line numbers or zero condition checking branching statements and execute the program while performing arithmetic operations using the Num class.

1. ArithmeticProgram.java

A program represented by a set of arithmetic expressions (both in infix and post-fix form)

and zero-check conditional branching statements, that could be executed in a top-down fashion.

1. ProgramStatement.java

Statement representing a line of an arithmetic program, containing all the necessary information to execute that line of the program.

1. PostfixExpression.java

Represents an arithmetic expression in the post-fix form.

1. Token.java

Represents a token in a mathematical expression.

1. ShuntingYard.java

Given an infix expression with each token separated by space, returns the postfix expression by using the Shunting Yard Algorithm.