CS1020 Lab #0
Exercise #3: Reading Techniques

http://www.comp.nus.edu.sg/~cs1020/3_ca/labs.html

Objective:

The objective of this exercise is to ensure that you know how to read inputs using any of the three methods which you will encounter in your take-home labs and sit-in labs.

Task statement:

There are various techniques of parsing inputs. In this exercise, you will implement 3 common techniques:

1. Given an integer N, you should read N lines, each line containing some data.
2. Read until some special character(s) (e.g. read until -1).
3. Read until the end of the file.

Write a program **Reading.java** that reads some input data in one of the following 3 formats.

Format 1: The first line of input contains the string "LIMIT". This means that the second line contains an integer N, which is the number of operations. The next N lines contain a string on each line, describing the operation to be performed.

Format 2: The first line of input contains the string "SENTINEL". This means that the subsequent lines contain a string on each line, describing the operation to be performed. The inputs end with a line containing the string "-1".

Format 3: The first line of input contains the string "EOF". This means that the subsequent lines contain a string on each line, describing the operation to be performed. You are to read until the end of file (in interactive input on UNIX, until the user enters **Ctrl-d**).

The string that describes the operation to be performed contains an operation (ADD, SUB, or MUL) followed by two integers *x* and *y*:

- ADD *x y* : Perform *x* + *y*
- SUB *x y* : Perform *x* – *y*
- MUL *x y* : Perform *x* * *y*

For each of the operations read, your program is to print out its result.

You may write additional method(s) to make your program more modular.

Example #1:

Input

```
LIMIT
2
ADD 14 32
MUL -6 20
```

Output

```
46
-120
```

Example #2:

Input

```
SENTINEL
MUL 15 4
ADD -17 30
-1
```

Output

```
60
13
```

Example #3:

Input

```
EOF
SUB 1234 5678
SUB -2 -300
SUB 99999 0
```

Output

```
-4444
298
99999
```