

Prescreen Question - Spreadsheet

For this assignment, you are to create a console spreadsheet evaluator. Your spreadsheet is to take input in the form of a CSV (comma separated value) file and also output the result in CSV format. Your spreadsheet will perform only the four basic mathematical operations (+, -, *, /). It should be able to reference other cells for data. The calculations are to be done in postfix form (RPN). For cells that require an evaluation, they are to be preceded with an equal sign, as is typical of spreadsheets today. All column names are single letters from A..Z. That is to say, only 25 columns will be allowed. All row names are numbers. The number of rows will not exceed one million. Input and output numerical values will be integers or floating point. All floating point values will have either a trailing or leading zero if necessary (3.0 or 0.3). All input values will be non-negative.

Your program will be expected to evaluate the given examples and other inputs with different numbers and matrix sizes. Do not assume the input matrix is rectangular.

You are **NOT** expected to be able to evaluate the following:

multi-operator operations	=1 2 3 + *
multi-cell operations	=A1 B1 +
negative inputs	=1 -1 +
single-operand operators	=5 !
random strings and other malformed input	=FOO 2 +
multi-cell indirection	A1: 2, B1: =A1, C1: =B1
Unreferenced cells	=Z1 (where Z1 doesn't have a value)

You may use any language of your choice. Aside from the correctness of your answer, other aspects of your result will be considered including structure, readability, adherence to conventions, and running time/space.

RPN Primer

See <http://www.alcula.com/calculators/rpn/> as an example of how a RPN calculator functions.

Notes about your submission

1. Please test your program against the examples below.
2. Please provide instructions to build and run your program from the shell.
3. The code will be evaluated by how clear the code is, if it works, and how resistant it is to edge cases.

Examples

The following examples will illustrate how your results will be evaluated. The column alignment is purely for demonstration purposes. You should not concern yourself with this in your program.

1. Basic File Parsing

```
1, 2, 3      1, 2, 3
4, 5, 6  -> 4, 5, 6
7, 8, 9      7, 8, 9
```

2. Single Operation Basic Arithmetic

```
1,      2,      3,      4      1,      2,      3,      4
=5 1 +,  =6 2 +,  =7 3 -,  =8 2 -  -> 6,      8,      4,      6
=9 3.5 *, =10 5 *, =11 2 /, =12 4 /    31.5, 50, 5.5, 3
```

3. Single Cell Referencing

Note: Cell references can be forward pointing. That is to say, cell B5 can contain "=Z100",

```
1,      2,      3      1, 2, 3
=A1, =B1, 6      -> 1, 2, 6
=B3, 8,      =C2    8, 8, 6
```

Handing In

You will hand in the source code for this program and any build scripts. In the comments at the top, you should include any building and running instructions.

Your program should be runnable on a typical Linux system as

```
$ ./kontagent_ss <CSV_FILE> >& target_file
```