Programming task: Spreadsheet

Dessert Labs

This task is designed to help us understand how you solve problems, how you translate your ideas into code, and how you model and structure a non-trivial programming project.

You can complete the task in a programming language of your choice, but please check with us if that's not one of Java, Javascript, Python, or C++.

Task description

Write a program to which parses a spreadsheet-like CSV file and evaluates each cell by these rules:

1. Each cell is an expression in postfix notation (see [Wikipedia](#)).
2. Each token in the expression will be separated by one or more spaces.
3. One cell can refer to another cell with the {LETTER}{NUMBER} notation (e.g. "A2", "B4"– letters refer to columns, numbers to rows).
4. Expressions may include the basic arithmetic operators +, -, *, /

Your program should output another CSV file of the same dimensions containing the results of evaluating each cell to its final value. If any cell is an invalid expression, then **for that cell only** print #ERR.

For example, for the following CSV input:

```
b1 b2 +,2 b2 3 * -,3 ,+
a1      ,5          , ,7 2 /
c2 3 * ,1 2         ,  ,5 1 2 + 4 * + 3 -
```

...output something like:

```
-8,-13,3,#ERR
-8,5,0,3.5
0,#ERR,0,14
```

There are other error conditions that your implementation should detect and handle in a manner you think appropriate.

Solution guide

- We are looking for well-written and structured code; don't rush. Expect to spend 3-4 hours.
- Use only what is available in the standard and most commonly used extension libraries for your platform.
- Where some detail of the task is unspecified, use your best judgement. Assumptions or limitations in your implementation are fine but please document them.

Deliverables

We want to see:

- Your source code and test input data
- A **short** report (½ page max) outlining your approach, any highlights or limitations of your implementation, and any trade-offs or design decisions you think worth noting.

We will chat about your implementation in a subsequent technical interview.