Homework on Lex

Christian Rinderknecht

25 October 2005

Due date: 15 November 2005

Write an integer postfix calculator in Lex.

For example, expressions such as `1 2 +` and `1 2 3 4 /*-` should be evaluated respectively to `3`, i.e. $1+2$, and `-0.5`, i.e. $1-(2\times 3/4)$. The rule is that once an operator is found, its arguments must have been entered before. Another correct expression is `1 2 + 3 *`, which means, in infix notation: $(1+2)\times 3$.

As a shorthand, it is possible to write `1-` instead of `0 1 -`.

White spaces only serve to separate numbers, but are otherwise optional. The end of line denotes the end of an expression.

The input file may have several lines (each of one must be a postfix arithmetic expression), therefore the calculation for each line must be printed in the same order as the input lines. For example, if the input file contains

```
1  2+
1 2 3 4 / *-
1 2 + 3 *
```

the calculator should print

```
3
-.5
9
```

Notes

- You probably need the C function `double atof(char*)`, which converts character strings to double precision floating-point numbers.
- Error messages must be printed and the position in the input file must be given (line number and column number) to help fixing the problem.
- Modern implementations of Lex are called Flex. You can find its source code or binary versions for Windows and Linux on the Web, as well as documentation (manuals or tutorials).