

## Exercises for chapter: *lex*

1. Write an integer postfix calculator in *lex*: expression such as `1 2 +` and `1 2 3 4/*-` should be evaluated to `3` and `-.5` respectively. White space only serves to separate number, but is otherwise optional; the line end denotes the end of an expression. You will probably need the C function `int atoi(char*)` which converts strings to ints.
2. It is possible to have `] and -` in a character range. The `] character has to be first, and - has to be either first or last. Why?`
3. Write regular expressions that match from the beginning of the line to the first letter `'a'`; to the last letter `'a'`. Also expressions that match from the first and last `'a'` to the end of the line. Include representative input and output in your answer.
4. Write a *lex* parser that analyzes text the way the `TEX` input processor does with the normal category code values. It should print its output with
  - `<space>` denoting any space that is not ignored or skipped, and
  - `<cs: command>` for recognizing a control sequence `\command`;
  - open and close braces should also be marked as `<{>`, `<}>`.

Here is some sample input:

```
this is {a line} of text.  
handle \control sequences \andsuch  
with \arg{uments}.  
    Aha!  
this line has %a comment
```

```
x  
y%  
z
```

```
\comm%  
and
```