CSCC24 Summer 2019 – Assignment 4
Due: Tuesday, August 6, midnight
This assignment may be done in pairs.
This assignment is worth 10% of the course grade.

In this assignment, you will implement in Haskell a parser for a toy language.
As usual, you should also aim for reasonably efficient algorithms and reasonably lucid code.

Turbo Parser

The Turbo language of the previous assignment is given the following EBNF grammar. This grammar is incomplete and contains ambiguity (under `<expr>`); these are resolved informally right after.

```
<stmt>  ::= <assign>
          | <pen-cmd>
          | <turn>
          | <forward>
          | <for-loop>
          | <seq>
<assign>   ::= <var> "=" <expr>
<pen-cmd>  ::= "penup" | "pendown"
<turn>     ::= "turn" <expr>
<forward>  ::= "forward" <expr>
<for-loop> ::= "for" <var> "=" <expr> "to" <expr> <seq>
<seq>      ::= "{" { <stmt> ";" } "}"

<expr> ::= <literal>
         | <var>
         | "(" <expr> ")"
         | "-" <expr>
         | <expr> <op> <expr>

<op> ::= "+" | "-" | "*" | "/"
```

- The start symbol is `<stmt>`.
- `<literal>` is for real number literals: One or more digits, optionally followed by "." and one or more digits. (Changed July 31.) (Unary prefix minus is handled separately.)
- `<var>` is for variable names: A letter followed by zero or more letters or digits. However, the following are reserved words and cannot be variable names: `pendown`, `penup`, `turn`, `forward`, `for`, `to`. (Changed July 31.)
- Ambiguity under `<expr>` is resolved by: Unary prefix "`-`" has the highest precedence, "`*`" and "`/`" have the same middle precedence, "`+`" and infix "`-`" have the same lowest precedence. The 4 binary infix operators associate to the left.

- There may be whitespaces around literals, variable names, and terminal strings, e.g.,

```
for  i=0 to  60  {
  forward s;
  s   =    s  *  0.99  ;turn   i*0.8;
  }
```

Implement a parser for Turbo; the name and type is
`mainParser :: Parser Stmt`
in file 'TurboParser.hs'. The correspondence from the syntax to the abstract syntax tree type `Stmt` should be mostly self-evident.

End of questions.