Programming Project

Write a lexical analyzer program to scan postfix expressions (input file), from the grammar I gave you (attached below), and recognize the lexemes as tokens. The program (can be written in C++ or Python if you wish) should identify the lexemes, identify their respective tokens and produce a table of lexemes and tokens.

Required:

1. A complete program listing
2. A complete description of your program (what it does, how it operates, input requirements, what does the output represent, etc.)
3. Test runs (run your program and print the screen results)
4. Also in your description, you should explain why you chose the test run data (input files) that you chose. That is, what did you test for?

Grammar

Postfix Expressions (Reverse Polish Notation) is the language generated by the grammar $G = (V, T, < Post >, P)$:

with production rules, P, as follows:

$$\begin{array}{lcl}
< Post > & \rightarrow & < const > \mid < variable > \mid < Post >< delim >< Post >< delim >< operator > \\
< letter > & \rightarrow & A \mid B \mid C \mid ... \mid Z \mid a \mid b \mid ... \mid z \\
< digit > & \rightarrow & 0 \mid 1 \mid 2 \mid ... \mid 8 \mid 9 \\
< operator > & \rightarrow & + \mid - \mid * \mid / \\
< decpt > & \rightarrow & . \\
< delim > & \rightarrow & \_ \\
< const > & \rightarrow & < digit >< digit^* > \big| < digit^* > . < digit >< digit^* > \\
< variable > & \rightarrow & < letter >< letter^* >
\end{array}$$