Program exercise - Lex

• Use Lex to generate a scanner for Micro/Ex
• Micro/Ex is an extension of Micro.
• Comment =>  %% ……. \n  – Please skip it.
• Tokens of Micro/Ex
  1. BEGIN
  2. END
  3. READ
  4. WRITE
  5. ID
  6. Integer Literal
     • Not prefixed with “+” and “-”
Program exercise - Lex

7. Float Point Literal
   • 12.345, 12.5, 0.1, 123.
   – Not prefixed with “+” and “-”

8. Exponential Float Point Literal
   • 0.123E12, 1.23e-3
   – Not prefixed with “+” and “-”

9. String Literal “this is a string”

10. Left parenthesis: (

11. Right parenthesis: )

12. Semicolon ;
Program exercise - Lex

13. Comma ,
14. Assign Operation :=
15. Plus Operation +
16. Minus Operation –
17. Multiplication Operation *
18. Division /
19. Not Equal !=
20. Greater than >
Program exercise - Lex

21. Less than <
22. Greater or equal >=
23. Less or equal <=
24. Equal ==
25. IF
26. THEN
27. ELSE
28. ENDIF
Program exercise - Lex

29. FOR
30. TO
31. ENDFOR
32. WHILE
33. ENDWHILE
34. DECLARE
35. AS
36. INTEGER
37. REAL
38. ScanEof
Program exercise - Lex

• Your program should report the number and the value of the scanned tokens sequentially.

• Also, it should signal lexical error when it scans an illegal token.
Program exercise - Lex

• Please use your scanner to process two files:
  – Please use the “exr_lex_test_data.txt” in the web site
  – Please write a Micro/Ex program which contains a lexical error and use your scanner to test it.
• Script file should contain the follows:
  – Source code of your lex program
  – Your Micro/EX program which contains lexical errors
  – The execution results for processing exr_lex_test_data.txt and your Micro/EX program
執行輸出參考格式

• 類似即可不用相同

Token number =1, value is “begin”
Token number =33, value is “declare”
Token number =5, value is “A”
Token number =13, value is “,”
.
.
.
.
.
.
.
.
.
.
.

Token number =37, value is “EOF”
End of the execution