Provided Files:

- A shell file
- i1.txt - A sample input file • i2.txt - A sample input file • i3.txt - A sample input file • i4.txt - A sample input file
- i5.txt - A sample input file
- w1.txt - The required output for i1.txt • w2.txt • w3.txt - The required output for i3.txt • w4.txt - The required output for i4.txt
- w5.txt - The required output for i5.txt

Description: Implement a lexical analyzer in Java for a small language, called **AC** for **A**dding **C**alculator, that accommodates two forms of numerical data types, allows computation and printing of numerical values, and offers a small set of variable names to hold the results of computations. Below are the language **AC**'s tokens and lexemes:

Token IDs	**Token Names**	**Lexemes in Regular Expression**
0	FLOATDCL	f
1	INTDCL	i
2	PRINT	p
3	ID	a+b+c+d+e+g+h+j+k+l+m+n+o+q+r+s+t+u+v+w+x+y+z
4	ASSIGN	=
5	PLUS	+
6	MINUS	-
7	INUM	$(0+1+2+3+4+5+6+7+8+9)^+$
8	FNUM	$(0+1+2+3+4+5+6+7+8+9)^+.(0+1+2+3+4+5+6+7+8+9)^+$

The lexical analyzer shall read a source program written in **AC** as a text file and prints out a stream of lexemes, where each lexeme represents an instance of some symbols in the **AC**'s alphabet. For example, strings such as 5 and 3.2 are recognized as lexemes of token types `INUM` and `FNUM`, respectively. Reserved keywords such as `f` and `p` are distinguished from identifiers such as `a` and `b`. Lexemes do not include any white space characters such as blanks (' '), tabs ('\t'), and new lines ('\n'), which serve only to format the source code. However, a

white space character can be used as a delimiter. The lexical analyzer should terminate right after it reads any undefined or invalid symbol (see i5.txt and w5.txt).

Given the input file, your program should produce exactly same required output.

You are not to use any member methods of class java.util.Scanner except hasNextLine and nextLine. For example, you CANNOT use hasNextInt, nextInt, hasNextFloat, nextFloat, hasNextDouble, and nextDouble for this assignment. You are NOT allowed to use any classes in Java package java.util.regex.