<stmt> : <assign> | <binop> | <declare>
<assign> : <id> ‘=’ {<term> | <str>}
<binop> : {<term> (<literal> | <exp>) <term>} | {<str> ‘+’ <str>}
<declare> : <var> {<id> | <assign>}
<term> : <id> | <num> | <binop>
<literal> : ‘+’, ‘-’, ‘*’, ‘/’
<id> : ID
<num> : NUM
<exp> : EXP
<str> : STR
<var> : ‘VAR ‘

x=0;	Invalid	Lexical error	Semicolon token is not defined so it’s considered as an illegal character.
VARx=y2**	Valid		
VAR VAR = x/2	Invalid	Syntactic error	All tokens are valid. VAR is reserved for <var>. And according to the grammar <declare>, what stands after <var> should be either <id> or <assign>.
VARx=VARy=3	Invalid	Syntactic error	All tokens are valid. According to the grammar <assign>, what stands after <id> = should be either <term> or <str>. But VAR appears after =, so this is syntactic error. My parser can only deal with one statement per line.
x = "string" + "123"	Invalid	Lexical error	“123” is not valid because STR type should contains only upper/lower-case letters, spaces, periods
"VAR" X = "str"	Invalid	Syntactic error	All tokens are valid. According to the grammar, there is no grammar like <str> <id> = <str>.
i = (2*2)3**	Invalid	Lexical error	Neither left and right parenthesis is defined tokens so they are considered as illegal characters.
VARx=1+2*3/56**	Valid		
VARx=1+2*3/56=y**	Invalid	Syntactic error	All tokens are valid. Look at <assign>, <term>, and <binop> , we can’t have another equal sign after <binop>.
x=1*2/34**	Valid		