PLC Test 2

Name: Ethan Tang

a.)

Tokens	Regular Expression
INT_LIT	(/d+)
VAR_IDENT	([A-Za-z_]{6, 8})
INT_TYPE_BYTE	(#b)
INT_TYPE_WORD	(#w)
INT_TYPE_DWORD	(#d)
INT_TYPE_QWORD	(#q)
IF_STMT	(\$i)
FOR_LOOP	(\$f)
WHILE_LOOP	(\$w)
BEGIN	(!b)
END	(!e)

b.)

Order of Operation: BDSMAO

Brackets first

Division and Subtraction (left to right)

Multiplication and Addition (left to right)

Order(Exponents)

<program> → !b <stmtlist> !e	Not Left Recursive
<stmtlist> → <stmt> | <stmt>; <stmtlist>	Not Left Recursive
<stmt> → <if_stmt> | <while> | <declr> | <for> | <assign>	Not Left Recursive
<if_stmt> → `$i``(`<bool_expr>`)`<stmt>	Not Left Recursive
<while> → `$w``(`<bool_expr>`)`<stmt>	Not Left Recursive
<for> → `$f` `(` INT_TYPE `>>` INT_LIT `)``:`<stmt>	Not Left Recursive
<declr> → INT_TYPE VAR_IDEN	Not Left Recursive
<assign> → VAR_IDEN `=` <expr>	Not Left Recursive
<expr> → <term> { (`*`|`+`) <term>}	Not Left Recursive
<term> → <factor> { (`/`|`-`|`%`) <factor>}	Not Left Recursive
<factor> → VAR_IDEN | INT_LIT | `(` <expr> `)`	Not Left Recursive
<bool_expr> → <brel> { (`!=`|`==`) <brel>}	Not Left Recursive
<brel> → <bexpr> { (`>=`|`<=`|`<`|`>`) <bexpr>}	Not Left Recursive
<bexpr> → <bterm> { (`*`|`+`) <bterm>}	Not Left Recursive
<bterm> → <bfactor> { (`/`|`-`|`%`) <bfactor>}	Not Left Recursive
<bfactor> → VAR_IDEN | INT_LIT | `(` <expr> `)`	Not Left Recursive

c.)

None of the grammar rules' LHS is appearing at the beginning of the RHS so this language's rule sets are not left recursive.

This grammar rule set is also not ambiguous because the LHS of the rules does not appear in its RHS hence there is only one derivation tree.

This language conforms to the LL grammar because it is not left recursive and unambiguous.

d.)

This language is unambiguous as explained in part c.

e.)

Code written in Lexeme.java

f.)

Code written in Syntax.java

g.)

First file test1.in has 5 lexical errors. They are represented by error code 444.

Error 1: #, invalid because # had to be followed by b, w, d, or q to represent a valid number type.

Error 2: q, invalid identifier. Identifier has to consist of 6 to 8 letters or underscores

Error 3: $d, invalid keyword. $d lexeme does not exist

Error 4: notvr, invalid identifier. Identifier has to consist of 6 to 8 letters or underscores

Error 5: ><, invalid relational operator. >< does not exist as a lexeme.

Second file test2.in has 5 syntax errors. They are scanned by Syntax.java. Any errors detected will force the program to end early.

Error 1: Line 2, missing integer type on variable assignment.

Error 2: Line 3, missing semicolon at the end of the statement.

Error 3: Line 4, missing UNTIL token after the first identifier inside the for loop condition.

Error 4: Line 5, unidentified relational operator >> inside the while loop condition.

Error 5: line 6, missing identifier on variable declaration.

Third file test3.in and fourth file test4.in have no error.

h.) Rules of the language:

```
LR(1) grammar ('' is ε):
(0) P -> !b SL !e
(1) SL -> S
(2) SL -> S ; SL
(3) S -> IS
(4) S -> W
(5) S -> D
(6) S -> A
(7) S -> F
(8) IS -> $i ( BE ) S
(9) W -> $w ( BE ) S
(10) F -> $f ( INTID >> INTLIT ) : S
(11) D -> INTID VARID
(12) A -> VARID = E
(13) E -> T
(14) E -> T * T
(15) E -> T + T
(16) T -> F
(17) T -> F / F
(18) T -> F - F
(19) T -> F % F
(20) F -> VARID
(21) F -> INTLIT
(22) F -> ( E )
(23) BE -> BREL
(24) BE -> BREL != BREL
(25) BE -> BREL == BREL
(26) BREL -> BEXPR
(27) BREL -> BEXPR >= BEXPR
(28) BREL -> BEXPR <= BEXPR
(29) BREL -> BEXPR > BEXPR
(30) BREL -> BEXPR < BEXPR
(31) BEXPR -> BTERM
(32) BEXPR -> BTERM * BTERM
(33) BEXPR -> BTERM + BTERM
(34) BTERM -> BFAC
(35) BTERM -> BFAC / BFAC
(36) BTERM -> BFAC % BFAC
(37) BTERM -> BFAC - BFAC
(38) BFAC -> VARID
(39) BFAC -> INTLIT
(40) BFAC -> ( E )
```

FIRST table

Nonterminal	FIRST
P	{!b}
SL	{$i,$w,INTID,VARID,$f,INTLIT,(}
S	{$i,$w,INTID,VARID,$f,INTLIT,(}
IS	{$i}
W	{$w}
F	{$f,VARID,INTLIT,(}
D	{INTID}
A	{VARID}
E	{$f,VARID,INTLIT,(}
T	{$f,VARID,INTLIT,(}
BE	{VARID,INTLIT,(}
BREL	{VARID,INTLIT,(}
BEXPR	{VARID,INTLIT,(}
BTERM	{VARID,INTLIT,(}
BFAC	{VARID,INTLIT,(}

Parse Table of this language:

LR table

| State | ACTION | | |

| | | |

| | | |

| | | |

| | | |

Code Sample 1: !b $w (VARID >= INTLIT) VARID = VARID + INTLIT !e PASS

Input (tokens): !b $w (VARID >= INTLIT) VARID = VARID + INTLIT !e

Maximum number of steps: 100

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	!b $w (VARID >= INTLIT) VARID = VARID + INTLIT !e $	s1	!b
2	0 !b 1	$w (VARID >= INTLIT) VARID = VARID + INTLIT !e $	s10	
3	0 !b 1 $w 10	(VARID >= INTLIT) VARID = VARID + INTLIT !e $	s19	
4	0 !b 1 $w 10 (19	VARID >= INTLIT) VARID = VARID + INTLIT !e $	s36	
5	0 !b 1 $w 10 (19 VARID 36	>= INTLIT) VARID = VARID + INTLIT !e $	r_{38}	
6	0 !b 1 $w 10 (19 BFAC	>= INTLIT) VARID = VARID + INTLIT !e $	35	
7	0 !b 1 $w 10 (19 BFAC 35	>= INTLIT) VARID = VARID + INTLIT !e $	r_{34}	
8	0 !b 1 $w 10 (19 BTERM	>= INTLIT) VARID = VARID + INTLIT !e $	34	
9	0 !b 1 $w 10 (19 BTERM 34	>= INTLIT) VARID = VARID + INTLIT !e $	r_{31}	
10	0 !b 1 $w 10 (19 BEXPR	>= INTLIT) VARID = VARID + INTLIT !e $	33	
11	0 !b 1 $w 10 (19 BEXPR 33	>= INTLIT) VARID = VARID + INTLIT !e $	s59	
12	0 !b 1 $w 10 (19 BEXPR 33 >= 59	INTLIT) VARID = VARID + INTLIT !e $	s107	
13	0 !b 1 $w 10 (19 BEXPR 33 >= 59 INTLIT 107	) VARID = VARID + INTLIT !e $	r_{39}	
14	0 !b 1 $w 10 (19 BEXPR 33 >= 59 BFAC	) VARID = VARID + INTLIT !e $	105	
15	0 !b 1 $w 10 (19 BEXPR 33 >= 59 BFAC 105	) VARID = VARID + INTLIT !e $	r_{34}	
16	0 !b 1 $w 10 (19 BEXPR 33 >= 59 BTERM	) VARID = VARID + INTLIT !e $	104	
17	0 !b 1 $w 10 (19 BEXPR 33 >= 59 BTERM 104	) VARID = VARID + INTLIT !e $	r_{31}	
18	0 !b 1 $w 10 (19 BEXPR 33 >= 59 BEXPR	) VARID = VARID + INTLIT !e $	103	
19	0 !b 1 $w 10 (19 BEXPR 33 >= 59 BEXPR 103	) VARID = VARID + INTLIT !e $	r_{27}	
20	0 !b 1 $w 10 (19 BREL	) VARID = VARID + INTLIT !e $	32	
21	0 !b 1 $w 10 (19 BREL 32	) VARID = VARID + INTLIT !e $	r_{23}	
22	0 !b 1 $w 10 (19 BE	) VARID = VARID + INTLIT !e $	39	
23	0 !b 1 $w 10 (19 BE 39	) VARID = VARID + INTLIT !e $	s69	
24	0 !b 1 $w 10 (19 BE 39) 69	VARID = VARID + INTLIT !e $	s12	
25	0 !b 1 $w 10 (19 BE 39) 69 VARID 12	= VARID + INTLIT !e $	s21	
26	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21	VARID + INTLIT !e $	s44	
27	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 VARID 44	+ INTLIT !e $	r_{20}	
28	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 F	+ INTLIT !e $	42	
29	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 F 42	+ INTLIT !e $	r_{16}	
30	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T	+ INTLIT !e $	41	
31	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41	+ INTLIT !e $	s71	
32	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41 + 71	INTLIT !e $	s130	
33	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41 + 71 INTLIT 130	!e $	r_{21}	
34	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41 + 71 F	!e $	127	
35	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41 + 71 F 127	!e $	r_{16}	
36	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41 + 71 T	!e $	132	
37	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 T 41 + 71 T 132	!e $	r_{15}	
38	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 E	!e $	40	
39	0 !b 1 $w 10 (19 BE 39) 69 VARID 12 = 21 E 40	!e $	r_{12}	
40	0 !b 1 $w 10 (19 BE 39) 69 A	!e $	7	
41	0 !b 1 $w 10 (19 BE 39) 69 A 7	!e $	r_{6}	
42	0 !b 1 $w 10 (19 BE 39) 69 S	!e $	125	
43	0 !b 1 $w 10 (19 BE 39) 69 S 125	!e $	r_{9}	
44	0 !b 1 W	!e $	5	
45	0 !b 1 W 5	!e $	r_{4}	
46	0 !b 1 S	!e $	3	
47	0 !b 1 S 3	!e $	r_{1}	
48	0 !b 1 SL	!e $	2	
49	0 !b 1 SL 2	!e $	s16	
50	0 !b 1 SL 2 !e 16	$	acc	

Code Sample 2: !b $i (VARID >= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e

PASS

Input (tokens): !b $i (VARID >= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e

Maximum number of steps: 100

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	!b $i (VARID >= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	s1	!b
2	0 !b 1	$i (VARID >= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	s9	
3	0 !b 1 $i 9	(VARID >= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	s18	
4	0 !b 1 $i 9 (18	VARID >= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	s36	
5	0 !b 1 $i 9 (18 VARID 36	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	r_{38}	
6	0 !b 1 $i 9 (18 BFAC	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	35	
7	0 !b 1 $i 9 (18 BFAC 35	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	r_{34}	
8	0 !b 1 $i 9 (18 BTERM	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	34	
9	0 !b 1 $i 9 (18 BTERM 34	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	r_{31}	
10	0 !b 1 $i 9 (18 BEXPR	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	33	
11	0 !b 1 $i 9 (18 BEXPR 33	>= INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	s59	
12	0 !b 1 $i 9 (18 BEXPR 33 >= 59	INTLIT) VARID = VARID + (INTLIT % INTLIT) !e $	s107	
13	0 !b 1 $i 9 (18 BEXPR 33 >= 59 INTLIT 107	) VARID = VARID + (INTLIT % INTLIT) !e $	r_{39}	
14	0 !b 1 $i 9 (18 BEXPR 33 >= 59 BFAC	) VARID = VARID + (INTLIT % INTLIT) !e $	105	
15	0 !b 1 $i 9 (18 BEXPR 33 >= 59 BFAC 105	) VARID = VARID + (INTLIT % INTLIT) !e $	r_{34}	
16	0 !b 1 $i 9 (18 BEXPR 33 >= 59 BTERM	) VARID = VARID + (INTLIT % INTLIT) !e $	104	
17	0 !b 1 $i 9 (18 BEXPR 33 >= 59 BTERM 104	) VARID = VARID + (INTLIT % INTLIT) !e $	r_{31}	
18	0 !b 1 $i 9 (18 BEXPR 33 >= 59 BEXPR	) VARID = VARID + (INTLIT % INTLIT) !e $	103	
19	0 !b 1 $i 9 (18 BEXPR 33 >= 59 BEXPR 103	) VARID = VARID + (INTLIT % INTLIT) !e $	r_{27}	
20	0 !b 1 $i 9 (18 BREL	) VARID = VARID + (INTLIT % INTLIT) !e $	32	
21	0 !b 1 $i 9 (18 BREL 32	) VARID = VARID + (INTLIT % INTLIT) !e $	r_{23}	
22	0 !b 1 $i 9 (18 BE	) VARID = VARID + (INTLIT % INTLIT) !e $	31	
23	0 !b 1 $i 9 (18 BE 31	) VARID = VARID + (INTLIT % INTLIT) !e $	s56	
24	0 !b 1 $i 9 (18 BE 31) 56	VARID = VARID + (INTLIT % INTLIT) !e $	s12	
25	0 !b 1 $i 9 (18 BE 31) 56 VARID 12	= VARID + (INTLIT % INTLIT) !e $	s21	
26	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21	VARID + (INTLIT % INTLIT) !e $	s44	
27	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 VARID 44	+ (INTLIT % INTLIT) !e $	r_{20}	
28	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 F	+ (INTLIT % INTLIT) !e $	42	
29	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 F 42	+ (INTLIT % INTLIT) !e $	r_{16}	
30	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T	+ (INTLIT % INTLIT) !e $	41	
31	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41	+ (INTLIT % INTLIT) !e $	s71	
32	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71	(INTLIT % INTLIT) !e $	s131	
33	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131	INTLIT % INTLIT) !e $	s28	
34	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 INTLIT 28	% INTLIT) !e $	r_{21}	
35	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 F	% INTLIT) !e $	25	
36	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 F 25	% INTLIT) !e $	s53	
37	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 F 25 % 53	INTLIT) !e $	s88	
38	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 F 25 % 53 INTLIT 88	) !e $	r_{21}	
39	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 F 25 % 53 F	) !e $	91	
40	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 F 25 % 53 F 91	) !e $	r_{19}	
41	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 T	) !e $	24	
42	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 T 24	) !e $	r_{13}	
43	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 E	) !e $	176	
44	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 E 176	) !e $	s241	
45	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 (131 E 176) 241	!e $	r_{22}	
46	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 F	!e $	127	
47	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 F 127	!e $	r_{16}	
48	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 T	!e $	132	
49	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 T 41 + 71 T 132	!e $	r_{15}	
50	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 E	!e $	40	
51	0 !b 1 $i 9 (18 BE 31) 56 VARID 12 = 21 E 40	!e $	r_{12}	
52	0 !b 1 $i 9 (18 BE 31) 56 A	!e $	7	
53	0 !b 1 $i 9 (18 BE 31) 56 A 7	!e $	r_{6}	
54	0 !b 1 $i 9 (18 BE 31) 56 S	!e $	94	
55	0 !b 1 $i 9 (18 BE 31) 56 S 94	!e $	r_{8}	
56	0 !b 1 IS	!e $	4	
57	0 !b 1 IS 4	!e $	r_{3}	
58	0 !b 1 S	!e $	3	
59	0 !b 1 S 3	!e $	r_{1}	
60	0 !b 1 SL	!e $	2	
61	0 !b 1 SL 2	!e $	s16	
62	0 !b 1 SL 2 !e 16	$	acc	

Code Sample 3: !b $i (VARID >< INTLIT) VARID = INTLIT !e FAIL

Input (tokens): !b $i (VARID >< INTLIT) VARID = INTLIT !e

Maximum number of steps: 100

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	!b $i (VARID >< INTLIT) VARID = INTLIT !e $	s1	
2	0 !b 1	$i (VARID >< INTLIT) VARID = INTLIT !e $	s9	
3	0 !b 1 $i 9	(VARID >< INTLIT) VARID = INTLIT !e $	s18	
4	0 !b 1 $i 9 (18	VARID >< INTLIT) VARID = INTLIT !e $	s36	
5	0 !b 1 $i 9 (18 VARID 36	>< INTLIT) VARID = INTLIT !e $		

Code Sample 4: !b $f (INTID >> INTLIT) VARID = INTLIT !e FAIL

Input (tokens): !b $f (INTID >> INTLIT) VARID = INTLIT !e

Maximum number of steps: 100

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	!b $f (INTID >> INTLIT) VARID = INTLIT !e $	s1	
2	0 !b 1	$f (INTID >> INTLIT) VARID = INTLIT !e $	s13	
3	0 !b 1 $f 13	(INTID >> INTLIT) VARID = INTLIT !e $	s22	
4	0 !b 1 $f 13 (22	INTID >> INTLIT) VARID = INTLIT !e $	s47	
5	0 !b 1 $f 13 (22 INTID 47	>> INTLIT) VARID = INTLIT !e $	s77	
6	0 !b 1 $f 13 (22 INTID 47 >> 77	INTLIT) VARID = INTLIT !e $	s142	
7	0 !b 1 $f 13 (22 INTID 47 >> 77 INTLIT 142	) VARID = INTLIT !e $	s180	
8	0 !b 1 $f 13 (22 INTID 47 >> 77 INTLIT 142) 180	VARID = INTLIT !e $		