<prog>
<identifier_start>
<identifier_body>
<dec-list>
<dec_enter>
<dec>
<type>
<stat-list_enter>
<stat-list>
<stat>
<write_enter>
<write>
<assign>
<expr_term_factor_enter>
<expr>
<term_enter>
<term>
<factor>
<number_start>
<number_body>
<sign>
<digit>
<letter>

Predictive Parsing Table With Error Messages

program

program <identifier_start> ; var <dec-list> begin <stat-list_enter> end.

E_integer is expected

var	begin	end.
E_program is expected	E_program is expected	E_program is expected
E_; is expected		
E_integer is expected	E_integer is expected	E_integer is expected
		lamda
		lamda

integer	display	;
E_program is expected	E_program is expected	E_program is expected
	E_; is expected	lamda
integer	E_integer is expected	E_integer is expected
	<stat> <stat-list>	
	<stat> <stat-list_enter>	
	<write_enter>	
	display (<write>	
		lamda
	E_; is expected	lamda
		lamda

:	,	"value="
E_program is expected	E_program is expected	E_program is expected
lamda	lamda	
lamda	, <dec_enter>	
E_integer is expected	E_integer is expected	E_integer is expected
	E_unexpected ,	"value=" , <identifier_start>) ;

(	)	=
E_program is expected	E_program is expected	E_program is expected
E_(is an invalid character	lamda	lamda
E_integer is expected	E_integer is expected	E_integer is expected
E_display is expected		
<term_enter> <expr>		
	lamda	
<factor> <term>		
	lamda	
(<expr_term_factor_enter>)		
	lamda	

+	-
E_program is expected	E_program is expected
lamda	lamda
E_integer is expected	E_integer is expected
<term_enter> <expr>	<term_enter> <expr>
+ <term_enter> <expr>	- <term_enter> <expr>
<factor> <term>	<factor> <term>
lamda	lamda
<number_start>	<number_start>
<sign> <digit> <number_body>	<sign> <digit> <number_body>
lamda	lamda
+	-

*	/	0
E_program is expected	E_program is expected	E_program is expected
lamda	lamda	<digit> <identifier_body>
E_integer is expected	E_integer is expected	E_integer is expected
		<term_enter> <expr>
		<factor> <term>
* <factor> <term>	/<factor> <term>	
		<number_start>
		<digit> <number_body>
lamda	lamda	<digit> <number_body>
		lamda
		0

1	2	3
E_program is expected	E_program is expected	E_program is expected
<digit> <identifier_body>	<digit> <identifier_body>	<digit> <identifier_body>
E_integer is expected	E_integer is expected	E_integer is expected
<term_enter> <expr>	<term_enter> <expr>	<term_enter> <expr>
<factor> <term>	<factor> <term>	<factor> <term>
<number_start>	<number_start>	<number_start>
<digit> <number_body>	<digit> <number_body>	<digit> <number_body>
<digit> <number_body>	<digit> <number_body>	<digit> <number_body>
lamda	lamda	lamda
1	2	3

4	5	6
E_program is expected	E_program is expected	E_program is expected
<digit> <identifier_body>	<digit> <identifier_body>	<digit> <identifier_body>
E_integer is expected	E_integer is expected	E_integer is expected
<term_enter> <expr>	<term_enter> <expr>	<term_enter> <expr>
<factor> <term>	<factor> <term>	<factor> <term>
<number_start>	<number_start>	<number_start>
<digit> <number_body>	<digit> <number_body>	<digit> <number_body>
<digit> <number_body>	<digit> <number_body>	<digit> <number_body>
lamda	lamda	lamda
4	5	6

7	8	9
E_program is expected	E_program is expected	E_program is expected
<digit> <identifier_body>	<digit> <identifier_body>	<digit> <identifier_body>
E_integer is expected	E_integer is expected	E_integer is expected
<term_enter> <expr>	<term_enter> <expr>	<term_enter> <expr>
<factor> <term>	<factor> <term>	<factor> <term>
<number_start>	<number_start>	<number_start>
<digit> <number_body>	<digit> <number_body>	<digit> <number_body>
<digit> <number_body>	<digit> <number_body>	<digit> <number_body>
lamda	lamda	lamda
7	8	9

p	q
E_program is expected	E_program is expected
<letter> <identifier_body>	<letter> <identifier_body>
<letter> <identifier_body>	<letter> <identifier_body>
<dec_enter> : <type> ;	<dec_enter> : <type> ;
<identifier_start> <dec>	<identifier_start> <dec>
E_integer is expected	E_integer is expected
<stat> <stat-list>	<stat> <stat-list>
<stat> <stat-list_enter>	<stat> <stat-list_enter>
<assign>	<assign>
<identifier_start>) ;	<identifier_start>) ;
<identifier_start> = <expr_term_factor_enter> ;	<identifier_start> = <expr_term_factor_enter> ;
<term_enter> <expr>	<term_enter> <expr>
<factor> <term>	<factor> <term>
<identifier_start>	<identifier_start>
E_can't use letters in number	E_can't use letters in number
p	q

r
E_program is expected
<letter> <identifier_body>
<letter> <identifier_body>
<dec_enter> : <type> ;
<identifier_start> <dec>
E_integer is expected
<stat> <stat-list>
<stat> <stat-list_enter>
<assign>
<identifier_start>) ;
<identifier_start> = <expr_term_factor_enter> ;
<term_enter> <expr>
<factor> <term>
<identifier_start>
E_can't use letters in number
r

s

E_program is expected

<letter> <identifier_body>

<letter> <identifier_body>

<dec_enter> : <type> ;

<identifier_start> <dec>

E_integer is expected

<stat> <stat-list>

<stat> <stat-list_enter>

<assign>

<identifier_start>) ;

<identifier_start> = <expr_term_factor_enter> ;

<term_enter> <expr>

<factor> <term>

<identifier_start>

E_can't use letters in number

s