	The Orignial given grammer	
	<prog>	->
EBNF	<identifier>	->
	<dec-list>	->
	<dec>	->
	<type>	->
	<stat-list>	->
	<stat>	->
	<write>	->
	<assign>	->
	<expr>	->
	<term>	->
	<factor>	->
	<factor>	->
EBNF	<number>	->
	<sign>	->
	<digit>	->
	<letter>	->

program <identifier> ; var <dec-list> begin <stat-list> end.
<letter>{<letter>|<digit>}
<dec> : <type> ;
<identifier> , <dec> | <identifier>
integer
<stat> | <stat> <stat-list>
<write> | <assign>
display ("value", <identifier>) ; | display (<identifier>) ;
<identifier> = <expr> ;
<expr> + <term> | <expr> - <term> | <term>
<term> * <factor> | <term> / <factor> | <factor>
(<expr>)
<identifier> | <number>
<sign> <digit> {<digit>}
+ | - | lamda
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
p | q | r | s

Left hand recursion

Left hand recursion

Left hand recursion

Left hand recursion

The Orignial Grammer in BNF		
<prog>	->	program <identifier_start> ; var <dec-list> begin <stat-list> end.
<identifier_start>	->	<letter><identifier_body>
<identifier_body>	->	<letter><identifier_body>
<identifier_body>	->	<digit><identifier_body>
<identifier_body>	->	lamda
<dec-list>	->	<dec> : <type> ;
<dec>	->	<identifier_start> , <dec>
<dec>	->	<idetifier_start>
<type>	->	integer
<stat-list>	->	<stat>
<stat-list>	->	<stat> <stat-list>
<stat>	->	<write>
<stat>	->	<assign>
<write>	->	display ("value", <identifier_start>) ;
<write>	->	display (<identifier_start>) ;
<assign>	->	<identifier_start> = <expr> ;
<expr>	->	<expr> + <term>
<expr>	->	<expr> - <term>
<expr>	->	<term>
<term>	->	<term> * <factor>
<term>	->	<term> / <factor>
<term>	->	<factor>
<factor>	->	(<expr>)
<factor>	->	<identifier_start>
<factor>	->	<number_start>
<number_start>	->	<sign> <digit> <number_body>
<number_body>	->	<digit><number_body>
<number_body>	->	lamda
<sign>	->	+
<sign>	->	-
<sign>	->	lamda
<digit>	->	0
<digit>	->	1
<digit>	->	2
<digit>	->	3
<digit>	->	4
<digit>	->	5
<digit>	->	6
<digit>	->	7
<digit>	->	8
<digit>	->	9
<letter>	->	p
<letter>	->	q

<letter>	->	r
<letter>	->	s

	Remove all left recursions	
	<prog>	->
	<identifier_start>	->
	<identifier_body>	->
	<identifier_body>	->
	<identifier_body>	->
	<dec-list>	->
INDETERMINATE	<dec>	->
INDETERMINATE	<dec>	->
	<type>	->
INDETERMINATE	<stat-list>	->
INDETERMINATE	<stat-list>	->
	<stat>	->
	<stat>	->
INDETERMINATE	<write>	->
INDETERMINATE	<write>	->
	<assign>	->
	<expr_term_factor_enter>	->
	<expr>	->
	<expr>	->
	<expr>	->
	<term_enter>	->
	<term>	->
	<term>	->
	<term>	->
	<factor>	->
	<factor>	->
	<factor>	->
	<number_start>	->
	<number_body>	->
	<number_body>	->
	<sign>	->
	<sign>	->
	<sign>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->
	<digit>	->

<letter>	->
<letter>	->
<letter>	->
<letter>	->

program <identifier_start> ; var <dec-list> begin <stat-list> end.
<letter> <identifier_body>
<letter> <identifier_body>
<digit> <identifier_body>
lamda
<dec> : <type> ;
<identifier_start> , <dec>
<identifier_start>
integer
<stat>
<stat> <stat-list>
<write>
<assign>
display ("value" , <identifier_start>) ;
display (<identifier_start>) ;
<identifier_start> = <expr_term_factor_enter> ;
<term_enter> <expr>
+ <term_enter> <expr>
- <term_enter> <expr>
lamda
<factor> <term>
* <factor> <term>
/<factor> <term>
lamda
(<expr_term_factor_enter>)
<identifier_start>
<number>
<sign> <digit> <number_body>
<digit> <number_body>
lamda
+
-
lamda
0
1
2
3
4
5
6
7
8
9

p
q
r
s

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

<digit>	->
<digit>	->
<digit>	->
<letter>	->
<letter>	->
<letter>	->
<letter>	->

program <identifier_start> ; var <dec-list> begin <stat-list_enter> end.
<letter> <identifier_body>
<letter> <identifier_body>
<digit> <identifier_body>
lamda
<dec_enter> : <type> ;
<identifier_start> <dec>
, <dec_enter>
lamda
integer
<stat> <stat-list>
lamda
<stat> <stat-list_enter>
<write_enter>
<assign>
display (<write>
"value" , <identifier_start>) ;
<identifier_start>) ;
<identifier_start> = <expr_term_factor_enter> ;
<term_enter> <expr>
+ <term_enter> <expr>
- <term_enter> <expr>
lamda
<factor> <term>
* <factor> <term>
/<factor> <term>
lamda
(<expr_term_factor_enter>)
<identifier_start>
<number_start>
<sign> <digit> <number_body>
<digit> <number_body>
lamda
+
-
lamda
0
1
2
3
4
5
6

7
8
9
p
q
r
s