Submitted By:
Naveen Venkat
2015A7PS0078P

Assumptions:

1. Only ID can be passed as the argument to a function
2. SIZE operator can be used before any ID in arithmetic expressions (not in boolean expressions)
3. *,/ have higher precedence over +,-

MODIFIED GRAMMAR RULES

1. <mainFunction> ===> MAIN SQO SQC <stmtsAndFunctionDefs> END
2. <stmtsAndFunctionDefs> ===> <stmtOrFunctionDef> <x1>
3. <x1> ===> _epsilon_
4. <x1> ===> <stmtsAndFunctionDefs>
5. <stmtOrFunctionDef> ===> <stmt>
6. <stmtOrFunctionDef> ===> <functionDef>
7. <stmt> ===> <funCallStmt>
8. <stmt> ===> <declarationStmt>
9. <stmt> ===> <ifStmt>
10. <stmt> ===> <ioStmt>
11. <stmt> ===> <assignmentStmt>
12. <functionDef> ===> FUNCTION SQO <parameter_list> SQC ASSIGNOP FUNID SQO <parameter_list> SQC <stmtsAndFunctionDefs> END
13. <parameter_list> ===> _epsilon_
14. <parameter_list> ===> <type> ID <other_params>
15. <other_params> ===> _epsilon_
16. <other_params> ===> COMMA <type> ID <other_params>
17. <type> ===> INT
18. <type> ===> REAL
19. <type> ===> STRING
20. <type> ===> MATRIX
21. <funCallStmt> ===> <functionCall> SEMICOLON
22. <functionCall> ===> FUNID OP <id_list> CL
23. <declarationStmt> ===> <type> <id_list> SEMICOLON
24. <ifStmt> ===> IF OP <booleanExpression> CL <stmtsAndFunctionDefs> <x2>
25. <x2> ===> ENDIF
26. <x2> ===> ELSE <stmtsAndFunctionDefs> ENDIF
27. <ioStmt> ===> READ OP ID CL SEMICOLON
28. <ioStmt> ===> PRINT OP ID CL SEMICOLON
29. <assignmentStmt> ===> <leftHandSide> ASSIGNOP <rightHandSide> SEMICOLON
30. <leftHandSide> ===> ID
31. <leftHandSide> ===> SQO <id_list> SQC
32. <rightHandSide> ===> <arithmeticExpression>
33. <rightHandSide> ===> <functionCall>
34. <rightHandSide> ===> <matrixExpression>
35. <rightHandSide> ===> <stringExpression>
36. <id_list> ===> ID <x3>
37. <x3> ===> _epsilon_

```
38. <x3> ===> COMMA <id_list>
39. <matrix> ===> SQO <rowlist> SQC
40. <rowlist> ===> <numlist> <x4>
41. <x4> ===> _epsilon_
42. <x4> ===> SEMICOLON <rowlist>
43. <numlist> ===> NUM <x5>
44. <x5> ===> _epsilon_
45. <x5> ===> COMMA <numlist>
46. <stringExpression> ===> <str_or_id> <x6>
47. <x6> ===> _epsilon_
48. <x6> ===> PLUS <stringExpression>
49. <str_or_id> ===> STR
50. <str_or_id> ===> ID
51. <matrix_or_id> ===> <matrix>
52. <matrix_or_id> ===> ID
53. <matrixExpression> ===> <matrix_or_id> <x7>
54. <x7> ===> _epsilon_
55. <x7> ===> <mat_op> <matrixExpression>
56. <mat_op> ===> PLUS
57. <mat_op> ===> MINUS
58. <arithmeticExpression> ===> <term> <x8>
59. <x8> ===> _epsilon_
60. <x8> ===> <term_op> <arithmeticExpression>
61. <term_op> ===> MINUS
62. <term_op> ===> PLUS
63. <term> ===> <factor> <x9>
64. <x9> ===> _epsilon_
65. <x9> ===> <factor_op> <term>
66. <factor_op> ===> MUL
67. <factor_op> ===> DIV
68. <factor> ===> <aVar>
69. <factor> ===> OP <arithmeticExpression> CL
70. <booleanExpression> ===> OP <booleanExpression> CL
<binLogicalOp> OP <booleanExpression> CL
71. <booleanExpression> ===> NOT OP <booleanExpression> CL
72. <booleanExpression> ===> <var> <relationalOp> <var>
73. <binLogicalOp> ===> AND
74. <binLogicalOp> ===> OR
75. <relationalOp> ===> LT
76. <relationalOp> ===> LE
77. <relationalOp> ===> EQ
78. <relationalOp> ===> GT
79. <relationalOp> ===> GE
80. <relationalOp> ===> NE
81. <matrixElement> ===> ID SQO NUM COMMA NUM SQC
82. <var> ===> ID
83. <var> ===> NUM
84. <var> ===> RNUM
85. <var> ===> <matrixElement>
86. <aVar> ===> <var>
87. <aVar> ===> <functionCall>
88. <aVar> ===> SIZE ID
```

FIRST SETS

Note: First set of a terminal is the terminal itself. So it has not been mentioned here.

```
<aVar> : { ID , NUM , RNUM , ID , FUNID , SIZE }
<arithmeticExpression> : { ID , NUM , RNUM , ID , FUNID , SIZE ,
OP }
<assignmentStmt> : { ID , SQO }
<binLogicalOp> : { AND , OR }
<booleanExpression> : { OP , NOT , ID , NUM , RNUM , ID }
<declarationStmt> : { INT , REAL , STRING , MATRIX }
<factor> : { ID , NUM , RNUM , ID , FUNID , SIZE , OP }
<factor_op> : { MUL , DIV }
<funCallStmt> : { FUNID }
<functionCall> : { FUNID }
<functionDef> : { FUNCTION }
<id_list> : { ID }
<ifStmt> : { IF }
<ioStmt> : { READ , PRINT }
<leftHandSide> : { ID , SQO }
<mainFunction> : { MAIN }
<mat_op> : { PLUS , MINUS }
<matrix> : { SQO }
<matrixElement> : { ID }
<matrixExpression> : { SQO , ID }
<matrix_or_id> : { SQO , ID }
<numlist> : { NUM }
<other_params> : { _epsilon_ , COMMA }
<parameter_list> : { _epsilon_ , INT , REAL , STRING , MATRIX }
<relationalOp> : { LT , LE , EQ , GT , GE , NE }
<rightHandSide> : {ID , NUM , RNUM , ID , FUNID , SIZE , OP ,
FUNID , SQO , ID , STR , ID }
<rowlist> : { NUM }
<stmt> : { FUNID , INT , REAL , STRING , MATRIX , IF , READ ,
PRINT , ID , SQO }
<stmtOrFunctionDef> : { FUNID , INT , REAL , STRING , MATRIX ,
IF , READ , PRINT , ID , SQO , FUNCTION }
<stmtsAndFunctionDefs> : { FUNID , INT , REAL , STRING , MATRIX ,
IF , READ , PRINT , ID , SQO , FUNCTION }
<str_or_id> : { STR , ID }
<stringExpression> : { STR , ID }
<term> : { ID , NUM , RNUM , ID , FUNID , SIZE , OP }
<term_op> : { MINUS , PLUS }
<type> : { INT , REAL , STRING , MATRIX }
<var> : { ID , NUM , RNUM , ID }
<x1> : { _epsilon_ , FUNID , INT , REAL , STRING , MATRIX , IF ,
READ , PRINT , ID , SQO , FUNCTION }
<x2> : { ENDIF , ELSE }
<x3> : { _epsilon_ , COMMA }
<x4> : { _epsilon_ , SEMICOLON }
<x5> : { _epsilon_ , COMMA }
<x6> : { _epsilon_ , PLUS }
```

```
<x7> : { _epsilon_ , PLUS , MINUS }
<x8> : { _epsilon_ , MINUS , PLUS }
<x9> : { _epsilon_ , MUL , DIV }
```

FOLLOW SETS

Note: Only those non terminals that derive null (_epsilon_) have been mentioned here

```
<stmtsAndFunctionDefs> : { END }
<stmtOrFunctionDef> : { END }
<other_params> : { SQC }
<parameter_list> : { SQC }
<x1> : { END }
<x2> : { END }
<id_list> : { SQC }
<x3> : { SQC }
<x4> : { SQC }
<rightHandSide> : { SEMICOLON }
<stringExpression> : { SEMICOLON }
<x5> : { SEMICOLON, SQC }
<x6> : { SEMICOLON }
<matrixExpression> : { SEMICOLON }
<x7> : { SEMICOLON }
<arithmeticExpression> : { SEMICOLON }
<x8> : { SEMICOLON }
<x9> : { MINUS, PLUS }
```