GROUP 56

Rishabh Garg (2014A7PS065P)
M Sharat Chandra (2014A7PS108P)

FIRSTS:

first(<program>): ε, DECLARE, DEF
first(<moduleDeclarations>): ε, DECLARE
first(<moduleDeclaration>): DECLARE
first(<moduleDefinitions>): DEF
first(<otherModules>): ε, MODULE
first(<driverModule>): DRIVER
first(<postDriver>): ε, DEF
first(<postModule>): DEF
first(<module>): MODULE
first(<ret>): ε, RETURNS
first(<input_plist>): ID
first(<input_plist_ex>): ε, COMMA
first(<output_plist>): ID
first(<output_plist_ex>): ε, COMMA
first(<dataType>): REAL, INTEGER, ARRAY, BOOLEAN
first(<type>): REAL, INTEGER, BOOLEAN
first(<moduleDef>): START
first(<statements>): ε, USE, SQBO, FOR, GET_VALUE, WHILE, ID, SWITCH, PRINT, DECLARE
first(<statement>): ε, USE, SQBO, FOR, GET_VALUE, WHILE, ID, SWITCH, PRINT, DECLARE
first(<ioStmt>): PRINT, GET_VALUE
first(<var>): FALSE, NUM, TRUE, ID, RNUM
first(<whichId>): ε, SQBO
first(<simpleStmt>): ε, USE, SQBO, ID
first(<assignmentStmt>): ID
first(<whichStmt>): ASSIGNOP, SQBO
first(<lvalueIDStmt>): ASSIGNOP
first(<lvalueARRStmt>): SQBO
first(<index>): NUM, ID
first(<moduleReuseStmt>): ε, USE, SQBO
first(<optional>): ε, SQBO
first(<idList>): ID
first(<idList_ex>): ε, COMMA
first(<expression>): FALSE, BO, NUM, TRUE, ID, RNUM

first(<expression1>): FALSE, BO, NUM, TRUE, ID, RNUM
first(<expression2>): FALSE, BO, NUM, TRUE, ID, RNUM
first(<expression3>): FALSE, BO, NUM, TRUE, ID, RNUM
first(<expression4>): FALSE, BO, NUM, TRUE, ID, RNUM
first(<op1>): PLUS, MINUS
first(<op2>): MUL, DIV
first(<relationalOp>): LE, GE, NE, LT, GT, EQ
first(<logicalOp>): AND, OR
first(<declareStmt>): DECLARE
first(<condionalStmt>): SWITCH
first(<caseStmt>): CASE
first(<value>): FALSE, NUM, TRUE
first(<default>): DEFAULT, ε
first(<iterativeStmt>): WHILE, FOR
first(<range>): NUM

FOLLOWS:

follow(<program>): $
follow(<moduleDeclarations>): DEF
follow(<moduleDeclaration>): DECLARE, DEF
follow(<moduleDefinitions>): $
follow(<otherModules>): DRIVER
follow(<driverModule>): $, DEF
follow(<postDriver>): $
follow(<postModule>): $, DEF
follow(<module>): DEF
follow(<ret>): START
follow(<input_plist>): SQBC
follow(<input_plist_ex>): SQBC
follow(<output_plist>): SQBC
follow(<output_plist_ex>): SQBC
follow(<dataType>): SQBC, SEMICOL, COMMA
follow(<type>): COMMA, SQBC, SEMICOL
follow(<moduleDef>): $, DEF
follow(<statements>): BREAK, END
follow(<statement>): USE, END, SQBO, FOR, GET_VALUE, SWITCH, DECLARE, BREAK, WHILE, PRINT, ID
follow(<ioStmt>): USE, END, FOR, WHILE, PRINT, ID, SQBO, GET_VALUE, DECLARE, BREAK, SWITCH
follow(<var>): GT, BC, GE, MUL, DIV, EQ, MINUS, AND, LE, SEMICOL, NE, LT, PLUS, OR

follow(<whichId>): GT, BC, GE, MUL, DIV, EQ, MINUS, AND, LE, SEMICOL, NE, LT, PLUS, OR
follow(<simpleStmt>): USE, END, FOR, WHILE, PRINT, ID, SQBO, GET_VALUE, DECLARE, BREAK, SWITCH
follow(<assignmentStmt>): USE, END, FOR, WHILE, PRINT, DECLARE, SQBO, GET_VALUE, ID, BREAK, SWITCH
follow(<whichStmt>): USE, END, FOR, WHILE, PRINT, ID, SQBO, GET_VALUE, DECLARE, BREAK, SWITCH
follow(<lvalueIDStmt>): USE, END, FOR, WHILE, PRINT, DECLARE, SQBO, GET_VALUE, ID, BREAK, SWITCH
follow(<lvalueARRStmt>): USE, END, FOR, WHILE, PRINT, DECLARE, SQBO, GET_VALUE, ID, BREAK, SWITCH
follow(<index>): SQBC
follow(<moduleReuseStmt>): USE, END, FOR, WHILE, PRINT, DECLARE, SQBO, GET_VALUE, ID, BREAK, SWITCH
follow(<optional>): USE
follow(<idList>): SQBC, SEMICOL, COLON
follow(<idList_ex>): SQBC, SEMICOL, COLON
follow(<expression>): SEMICOL, BC
follow(<expression1>): LE, BC, SEMICOL, NE, LT, GE, GT, EQ
follow(<expression2>): AND, LE, BC, SEMICOL, NE, LT, GE, GT, EQ, OR
follow(<expression3>): AND, GT, BC, SEMICOL, NE, LT, GE, LE, PLUS, EQ, MINUS, OR
follow(<expression4>): GT, BC, GE, MUL, DIV, EQ, MINUS, AND, LE, SEMICOL, NE, LT, PLUS, OR
follow(<op1>): FALSE, BO, NUM, TRUE, ID, RNUM
follow(<op2>): FALSE, BO, NUM, TRUE, ID, RNUM
follow(<relationalOp>): FALSE, BO, NUM, TRUE, ID, RNUM
follow(<logicalOp>): FALSE, BO, NUM, TRUE, ID, RNUM
follow(<declareStmt>): USE, END, FOR, WHILE, PRINT, ID, SQBO, GET_VALUE, DECLARE, BREAK, SWITCH
follow(<condionalStmt>): USE, END, FOR, WHILE, PRINT, ID, SQBO, GET_VALUE, DECLARE, BREAK, SWITCH
follow(<caseStmt>): USE, END, SQBO, FOR, DEFAULT, GET_VALUE, WHILE, DECLARE, BREAK, SWITCH, PRINT, ID
follow(<value>): COLON
follow(<default>): END
follow(<iterativeStmt>): USE, END, FOR, WHILE, PRINT, ID, SQBO, GET_VALUE, DECLARE, BREAK, SWITCH
follow(<range>): SQBC, BC

Note: $ corresponds to end-marker symbol.