GROUP-8

AYUSH JAIN- 2017A7PS0093P
BHARAT BHARGAVA- 2017A7PS0025P
SATVIK GOLECHHA- 2017A7PS0117P

FIRST:

<moduleDeclarations>	ε, DECLARE
<moduleDeclaration>	DECLARE
<otherModules>	ε, DEF
<driverModule>	DRIVERDEF
<module>	DEF
<ret>	RETURNS, ε
<input_plist_lr>	COMMA, ε
<input_plist>	ID
<output_plist_lr>	COMMA, ε
<output_plist>	ID
<datatype>	INTEGER, REAL, BOOLEAN, ARRAY
<type>	INTEGER, REAL, BOOLEAN
<moduleDef>	START
<statements>	ε, GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH
<ioStmt>	GET_VALUE, PRINT
<var>	ID, NUM, RNUM
<whichId>	SQBO, ε
<assignmentStmt>	ID
<lvalueIDStmt>	ASSIGNOP
<lvalueARRStmt>	SQBO
<index>	NUM, ID

<moduleReuseStmt>	USE, SQBO
<optional>	SQBO, ε
<idList_lr>	COMMA, ε
<idList>	ID
<opt_expr_lr>	ε, AND, OR
<one_more_opt>	ε, LT, LE, GT, GE, EQ, NE
<booleanConst>	true, false
<arithmeticExpr_lr>	ε, PLUS, MINUS
<term_lr>	ε, MUL, DIV
<factor>	BO, ID, NUM, RNUM, PLUS, MINUS
<op>	PLUS, MINUS, MUL, DIV
<op1>	PLUS, MINUS
<op2>	MUL, DIV
<logicalOp>	AND, OR
<relationalOp>	LT, LE, GT, GE, EQ, NE
<declareStmt>	DECLARE
<value>	NUM, TRUE, FALSE
<caseStmt>	CASE
<default>	DEFAULT, ε
<conditionalStmt>	SWITCH
<range>	NUM
<iterativeStmt>	FOR, WHILE
<rangeArr>	NUM, ID
<statement>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH
<printOpt>	true, false, ID, NUM, RNUM
<simpleStmt>	ID, USE, SQBO
<whichStmt>	ASSIGNOP, SQBO
<program>	DECLARE, DRIVERDEF, DEF

<term>	BO, ID, NUM, RNUM, PLUS, MINUS
<arithmeticExpr>	BO, ID, NUM, RNUM, PLUS, MINUS
<opt_expr>	BO, ID, NUM, RNUM, PLUS, MINUS, true, false
<expression>	BO, ID, NUM, RNUM, PLUS, MINUS, true, false

FOLLOW:

<program>	$
<moduleDeclarations>	DRIVERDEF, DEF
<moduleDeclaration>	DECLARE, DRIVERDEF, DEF
<otherModules>	DRIVERDEF, $
<driverModule>	DEF, $
<module>	DEF, DRIVERDEF, $
<ret>	START
<input_plist_lr>	SQBC
<input_plist>	SQBC
<output_plist_lr>	SQBC
<output_plist>	SQBC
<datatype>	
<rangeArr>	SQBC
<type>	COMMA, SQBC
<moduleDef>	DEF, DRIVERDEF, $
<statements>	END, BREAK
<statement>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<ioStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<var>	MUL, DIV, BC, PLUS, MINUS, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<whichId>	MUL, DIV, BC, PLUS, MINUS, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<printOpt>	BC
<simpleStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK

<assignmentStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<whichStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<lvalueIDStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<lvalueARRStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<index>	SQBC, RANGEOP
<moduleReuseStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<optional>	USE
<idList_lr>	COLON, SQBC, SEMICOL
<idList>	COLON, SQBC, SEMICOL
<expression>	BC, SEMICOL
<opt_expr_lr>	BC, SEMICOL
<opt_expr>	AND, OR, BC, SEMICOL
<one_more_opt>	AND, OR, BC, SEMICOL
<booleanConst>	AND, OR, BC, SEMICOL
<arithmeticExpr_lr>	BC, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<arithmeticExpr>	BC, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<term_lr>	PLUS, MINUS, BC, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<term>	PLUS, MINUS, BC, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<factor>	MUL, DIV, PLUS, MINUS, BC, LT, LE, GT, GE, EQ, NE, AND, OR, SEMICOL
<op>	
<op1>	BO, ID, NUM, RNUM, PLUS, MINUS
<op2>	BO, ID, NUM, RNUM, PLUS, MINUS
<logicalOp>	BO, ID, NUM, RNUM, PLUS, MINUS, true, false
<relationalOp>	BO, ID, NUM, RNUM, PLUS, MINUS

<declareStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<value>	COLON
<caseStmt>	ID, DEFAULT
<default>	END
<conditionalStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK
<range>	BC
<iterativeStmt>	GET_VALUE, PRINT, DECLARE, FOR, WHILE, ID, USE, SQBO, SWITCH, END, BREAK