FIRST AND FOLLOW SET

NONTERMINALS	FIRST SET	FOLLOW SET
<program>	DECLARE, DEF, DRIVERDEF	$
<moduleDeclarations>	DECLARE, ε	DEF, DRIVERDEF
<moduleDeclaration>	DECLARE	DEF, DRIVERDEF, DECLARE
<otherModule>	DEF, ε	DEF, $
<module>	DEF	DEF, DRIVERDEF, $
<driverModule>	DRIVERDEF	DEF, $
<ret>	RETURNS, ε	START
<input_plist>	ID	SQBC
<input_plistRec>	COMMA, ε	SQBC
<output_plist>	ID	SQBC
<output_plistRec>	COMMA, ε	SQBC
<type>	INTEGER, REAL, BOOLEAN	SQBC, COMMA, SEMICOL
<dataType>	INTEGER, REAL, BOOLEAN, ARRAY	COMMA, SQBC, SEMICOL
<moduleDef>	START	DEF, DRIVERDEF, $
<statements>	DECLARE, PRINT, USE, FOR, GET_VALUE, SWITCH, WHILE, ID, SEMICOL, SQBO, ε	BREAK, END
<statement>	DECLARE, PRINT, USE, FOR, GET_VALUE, SWITCH, WHILE, ID, SEMICOL, SQBO	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<ioStmt>	GET_VALUE, PRINT	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<whichId>	SQBO, ε	AND, OR, PLUS, MINUS, MUL, DIV, LT, LE, GT, GE, NE, EQ, SEMICOL, ASSIGNOP, BC
<index>	NUM, ID	SQBC

<simpleStmt>	ID, USE, SQBO	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<assignmentStmt>	ID	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<moduleReuseStmt>	SQBO, USE	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<optional>	SQBO, ε	USE
<idList>	ID	SEMICOL, SQBC, COLON
<idListRec>	COMMA, ε	SEMICOL, SQBC, COLON
<expression>	TRUE, FALSE, ID, NUM, RNUM, MINUS, BO	SEMICOL
<arithOrBoolExpr>	TRUE, FALSE, ID, NUM, RNUM, BO	SEMICOL, BC
<arithOrBoolExprRec>	AND, OR, ε	SEMICOL, BC
<anyTerm>	TRUE, FALSE, ID, NUM, RNUM, BO	AND, OR, SEMICOL, BC
<anyTermRec>	EMPTY, LT, LE, GT, GE, NE, EQ	AND, OR, SEMICOL, BC
<arithmeticExpr>	TRUE, FALSE, ID, NUM, RNUM, BO	AND, OR, LT, LE, GT, GE, NE, EQ, SEMICOL, BC
<arithmeticExprRec>	PLUS, MINUS, ε	AND, OR, LT, LE, GT, GE, NE, EQ, SEMICOL, BC
<term>	TRUE, FALSE, ID, NUM, RNUM, BO	AND, OR, PLUS, MINUS, LT, LE, GT, GE, NE, EQ, SEMICOL, BC
<termRec>	MUL, DIV, ε	AND, OR, PLUS, MINUS, LT, LE, GT, GE, NE, EQ, SEMICOL, BC
<factor>	TRUE, FALSE, ID, NUM, RNUM, BO	AND, OR, PLUS, MINUS, MUL, DIV, LT, LE, GT, GE, NE, EQ, SEMICOL, BC
<var>	TRUE, FALSE, ID, NUM, RNUM	AND, OR, PLUS, MINUS, MUL, DIV, LT, LE, GT, GE, NE,

		EQ, SEMICOL, BC
<pm>	PLUS, MINUS	TRUE, FALSE, ID, NUM, RNUM, BO
<md>	MUL, DIV	TRUE, FALSE, ID, NUM, RNUM, BO
<logicalOp>	AND, OR	TRUE, FALSE, ID, NUM, RNUM, BO
<relationalOp>	LT, LE, GT, GE, EQ, NE	TRUE, FALSE, ID, NUM, RNUM, BO
<declareStmt>	DECLARE	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<conditionalStmt>	SWITCH	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<caseStmts>	CASE	DEFAULT, END
<caseStmtsRec>	CASE, ε	DEFAULT, END
<value>	NUM, TRUE, FALSE	COLON
<default>	DEFAULT, ε	END
<iterativeStmt>	FOR, WHILE	DECLARE, PRINT, USE, FOR, END, GET_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO
<range>	NUM	BC, SQBC

The FIRST set of a terminal is a singleton set containing only that terminal, as **FIRST(<terminal>) = {<terminal>}**