**Group 48**

Shubham Tiwari 2016B4A70935P

Puneet Anand 2016B4A70487P

Mayank Jasoria 2016B4A70703P

Vibhav Oswal 2016B4A70594P

<program> --> <moduleDeclarations> <otherModules> <driverModule> <otherModules>

<moduleDeclarations> --> <moduleDeclaration> <moduleDeclarations>

| Ɛ

<moduleDeclaration> --> DECLARE MODULE ID SEMICOL

<otherModules> --> <module> <otherModules>

| Ɛ

<driverModule> --> DRIVERDEF DRIVER PROGRAM DRIVERENDDEF <moduleDef>

<module> --> DEF MODULE ID ENDDEF TAKES INPUT SQBO <input\_plist> SQBC SEMICOL <ret> <moduleDef>

<ret> --> RETURNS SQBO <output\_plist> SQBC SEMICOL

| Ɛ

<input\_plist> --> ID COLON <dataType> <input\_plistNew>

<input\_plistNew> --> COMMA ID COLON <dataType> <input\_plistNew>

| Ɛ

<output\_plist> --> ID COLON <type> <output\_plistNew>

<output\_plistNew> --> COMMA ID COLON <type> <output\_plistNew>

| Ɛ

<type> --> INTEGER

| REAL

| BOOLEAN

<dataType> --> <type>

| ARRAY SQBO <range> SQBC OF <type>

<moduleDef> --> START <statements> END

<statements> --> <statement> <statements>

| Ɛ

<statement> --> <ioStmt>

| <simpleStmt>

| <declareStmt>

| <condionalStmt>

| <iterativeStmt>

<ioStmt> --> GET\_VALUE BO ID <whichId> BC SEMICOL

| PRINT BO <expression> BC SEMICOL

<whichId> --> SQBO <index> SQBC

| Ɛ

<index> --> NUM

| ID

<simpleStmt> --> <assignmentStmt>

| <moduleReuseStmt>

<assignmentStmt> --> ID <whichId> ASSIGNOP <expression> SEMICOL

<moduleReuseStmt> --> <optional> USE MODULE ID WITH PARAMETERS <idList> SEMICOL

<optional> --> SQBO <idList> SQBC ASSIGNOP

| Ɛ

<idList> --> ID <idListNew>

<idListNew> --> COMMA ID <idListNew>

| Ɛ

<expression> --> <arithOrBoolExpr>

| MINUS BO <arithmeticExpr> BC

| PLUS BO <arithmeticExpr> BC

<arithOrBoolExpr> --> <RelopExpr> <arithOrBoolExprNew>

<arithOrBoolExprNew> --> <logicalOp> <RelopExpr> <arithOrBoolExprNew>

| Ɛ

<RelopExpr> --> <arithmeticExpr> <RelopExprNew>

<RelopExprNew> --> <relationalOp> <arithmeticExpr> <RelopExprNew>

| Ɛ

<arithmeticExpr> --> <term> <arithmeticExprNew>

<arithmeticExprNew> --> <pm> <term> <arithmeticExprNew>

| Ɛ

<term> --> <factor> <termNew>

<termNew> --> <md> <factor> <termNew>

| Ɛ

<factor> --> BO <arithOrBoolExpr> BC

| <varNew>

<varNew> --> <pm> <varNew>

| <var>

<var> --> ID <whichId>

| NUM

| RNUM

| TRUE

| FALSE

<pm> --> PLUS

| MINUS

<md> --> MUL

| DIV

<logicalOp> --> AND

| OR

<relationalOp> --> LT

| LE

| GT

| GE

| EQ

| NE

<declareStmt> --> DECLARE <idList> COLON <dataType> SEMICOL

<condionalStmt> --> SWITCH BO ID BC START <caseStmts> <default> END

<caseStmts> --> CASE <value> COLON <statements> BREAK SEMICOL <caseStmtsNew>

<caseStmtsNew> --> CASE <value> COLON <statements> BREAK SEMICOL <caseStmtsNew>

| Ɛ

<value> --> NUM

| TRUE

| FALSE

<default> --> DEFAULT COLON <statements> BREAK SEMICOL

| Ɛ

<iterativeStmt> --> FOR BO ID IN <range> BC START <statements> END

| WHILE BO <arithOrBoolExpr> BC START <statements> END

<range> --> NUM RANGEOP NUM

FIRST AND FOLLOW SET

|  |  |  |
| --- | --- | --- |
| **NONTERMINALS** | **FIRST SET** | **FOLLOW SET** |
| <program> | {DECLARE, DEF, DRIVERDEF} | {$} |
| <moduleDeclarations> | {DECLARE, Ɛ} | {DEF, DRIVERDEF} |
| <moduleDeclaration> | {DECLARE} | {DEF, DRIVERDEF, DECLARE} |
| <otherModule> | {DEF, Ɛ} | {DEF, $} |
| <module> | {DEF} | {DEF, DRIVERDEF, $} |
| <driverModule> | {DRIVERDEF} | {DEF, $} |
| <ret> | {RETURNS, Ɛ} | {START} |
| <input\_plist> | {ID} | {SQBC} |
| <input\_plistNew> | {COMMA, Ɛ} | {SQBC} |
| <output\_plist> | {ID} | {SQBC} |
| <output\_plistNew> | {COMMA, Ɛ} | {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, Ɛ} | {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, Ɛ} | {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, Ɛ} | {USE} |
| <idList> | {ID} | {SEMICOL, SQBC, COLON} |
| <idListNew> | {COMMA, Ɛ} | {SEMICOL, SQBC, COLON} |
| <expression> | {TRUE, FALSE, ID, NUM, RNUM, MINUS, PLUS, BO} | {SEMICOL, BC} |
| <arithOrBoolExpr> | {TRUE, FALSE, ID, NUM, RNUM, BO} | {SEMICOL, BC} |
| <arithOrBoolExprNew> | {AND, OR, Ɛ} | {SEMICOL, BC} |
| <RelopExpr> | {TRUE, FALSE, ID, NUM, RNUM, BO} | {AND, OR, SEMICOL, BC} |
| <RelopExprNew> | {Ɛ, 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} |
| <arithmeticExprNew> | {PLUS, MINUS, Ɛ} | {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} |
| <termNew> | {MUL, DIV, Ɛ} | {AND, OR, PLUS, MINUS, LT, LE, GT, GE, NE, EQ, SEMICOL, BC} |
| <factor> | {PLUS, MINUS, TRUE, FALSE, ID, NUM, RNUM, BO} | {AND, OR, PLUS, MINUS, MUL, DIV, LT, LE, GT, GE, NE, EQ, SEMICOL, BC} |
| <varNew> | {PLUS, MINUS, TRUE, FALSE, ID, NUM, RNUM} | {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, PLUS, MINUS} |
| <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} |
| <caseStmtsNew> | {CASE, Ɛ} | {DEFAULT, END} |
| <value> | {NUM, TRUE, FALSE} | {COLON} |
| <default> | {DEFAULT, Ɛ} | {END} |
| <iterativeStmt> | {FOR, WHILE} | {DECLARE, PRINT, USE, FOR, END, GET\_VALUE, SWITCH, BREAK, WHILE, ID, SEMICOL, SQBO} |
| <range> | {NUM} | {BC, SQBC} |