

GROUP No. 19

1. Animesh Bhargava - 2019B3A70545P
2. Ishvit Bhasin - 2019B5A70226P
3. Ritik Thakur - 2019B2A70878P
4. Salil Gupta - 2019B3A70154P
5. Utkarsh Yashvardhan - 2019B4A70704P

Total Pages - 05

Grammar

Group - 19

- $\langle \text{program} \rangle \rightarrow \langle \text{module Declarations} \rangle \langle \text{other Modules} \rangle \langle \text{driverModule} \rangle \vee \langle \text{other Modules} \rangle$
- $\langle \text{module Declarations} \rangle \rightarrow \langle \text{module Declaration} \rangle \langle \text{module Declarations} \rangle$
- $\langle \text{module Declarations} \rangle \rightarrow \epsilon$
- $\langle \text{module Declaration} \rangle \rightarrow \text{DECLARE MODULE ID SEMICOL}$
- $\langle \text{other Modules} \rangle \rightarrow \langle \text{module} \rangle \langle \text{other Modules} \rangle$
- $\langle \text{other Modules} \rangle \rightarrow \epsilon$
- $\langle \text{driver Module} \rangle \rightarrow \text{DRIVERDEF DRIVER PROGRAM DRIVER ENDDEF} \langle \text{module Def} \rangle$
- $\langle \text{module} \rangle \rightarrow \text{DEF MODULE ID ENDDEF TAKES INPUT SQBO} \langle \text{input-plist} \rangle \text{ SQBC SEMICOL} \langle \text{ret} \rangle \vee \langle \text{moduleDef} \rangle$
- $\langle \text{ret} \rangle \rightarrow \text{RETURNS SQBO} \langle \text{output-plist} \rangle \text{ SQBC SEMICOL}$
- $\langle \text{ret} \rangle \rightarrow \epsilon$
- $\langle \text{input-plist} \rangle \rightarrow \text{ID COLON} \langle \text{datatype} \rangle$
- $\langle \text{input-plist} \rangle \rightarrow \text{ID COLON} \langle \text{type} \rangle \langle \text{input-plist-bar} \rangle$
- $\langle \text{input-plist-bar} \rangle \rightarrow \text{COMMA ID COLON} \langle \text{datatype} \rangle \langle \text{input-plist-bar} \rangle$
- $\langle \text{input-plist-bar} \rangle \rightarrow \text{-----} \epsilon$
- $\langle \text{output-plist} \rangle \rightarrow \text{ID COLON} \langle \text{datatype} \rangle$
- $\langle \text{output-plist} \rangle \rightarrow \text{ID COLON} \langle \text{datatype} \rangle \langle \text{output-plist-bar} \rangle$
- $\langle \text{output-plist-bar} \rangle \rightarrow \text{COMMA ID COLON} \langle \text{datatype} \rangle \langle \text{output-plist-bar} \rangle$
- $\langle \text{output-plist-bar} \rangle \rightarrow \epsilon$
- $\langle \text{datatype} \rangle \rightarrow \text{INTEGER}$
- $\langle \text{datatype} \rangle \rightarrow \text{REAL}$
- $\langle \text{datatype} \rangle \rightarrow \text{BOOLEAN}$
- $\langle \text{datatype} \rangle \rightarrow \text{ARRAY SQBO} \langle \text{range} \rangle \text{ SQBC OF} \langle \text{type} \rangle$
- $\langle \text{type} \rangle \rightarrow \text{INTEGER}$
- $\langle \text{type} \rangle \rightarrow \text{REAL}$
- $\langle \text{type} \rangle \rightarrow \text{BOOLEAN}$
- $\langle \text{moduleDef} \rangle \rightarrow \text{START} \langle \text{statements} \rangle \text{ END}$
- $\langle \text{statements} \rangle \rightarrow \langle \text{statement} \rangle \langle \text{statements} \rangle$
- $\langle \text{statements} \rangle \rightarrow \epsilon$

- $\langle \text{statement} \rangle \rightarrow \langle \text{ioStmt} \rangle$
- $\langle \text{statement} \rangle \rightarrow \langle \text{simpleStmt} \rangle$
- $\langle \text{statement} \rangle \rightarrow \langle \text{declareStmt} \rangle$
- $\langle \text{statement} \rangle \rightarrow \langle \text{conditionalStmt} \rangle$
- $\langle \text{statement} \rangle \rightarrow \langle \text{iterativeStmt} \rangle$
- $\langle \text{ioStmt} \rangle \rightarrow \text{GET_VALUE } \text{BO } \text{ID } \text{BC } \text{SEMICOL}$
- $\langle \text{ioStmt} \rangle \rightarrow \text{PRINT } \text{BO } \langle \text{var} \rangle \text{ BC SEMICOL}$
- $\langle \text{var} \rangle \rightarrow \langle \text{var-numeral} \rangle \quad \text{REDACTED}$
- $\langle \text{var} \rangle \rightarrow \langle \text{bool Value} \rangle$
- $\langle \text{var-numeral} \rangle \rightarrow \text{ID } \langle \text{which ID} \rangle$
- $\langle \text{var-numeral} \rangle \rightarrow \text{NUM}$
- $\langle \text{var-numeral} \rangle \rightarrow \text{RNUM}$
- $\langle \text{bool Value} \rangle \rightarrow \text{True} \mid \text{false}$
- $\langle \text{which ID} \rangle \rightarrow \text{SQBO } \text{ID } \text{SQBC}$
- $\langle \text{which ID} \rangle \rightarrow \text{SQBO } \text{NUM } \text{SQBC}$
- $\langle \text{which ID} \rangle \rightarrow \epsilon$
- $\langle \text{simpleStmt} \rangle \rightarrow \langle \text{assignment Stmt} \rangle$
- $\langle \text{simpleStmt} \rangle \rightarrow \langle \text{module Reuse Stmt} \rangle$

- $\langle \text{assignment Stmt} \rangle \rightarrow \text{ID} (\langle \text{which Stmt} \rangle)$
- $\langle \text{which Stmt} \rangle \rightarrow \langle \text{Lvalue ID Stmt} \rangle$
- $\langle \text{which Stmt} \rangle \rightarrow \langle \text{Lvalue ARR Stmt} \rangle$
- $\langle \text{Lvalue ID Stmt} \rangle \rightarrow \text{ASSIGN OP} \langle \text{expression} \rangle \text{ SEMICOL}$
- $\langle \text{Lvalue ARR Stmt} \rangle \rightarrow \text{SQBO} \langle \text{index} \rangle \text{ SQBC ASSIGN OP} \langle \text{expression} \rangle \text{ SEMICOL}$.
- $\langle \text{index} \rangle \rightarrow \text{NUM}$
- $\langle \text{index} \rangle \rightarrow \text{ID}.$
- $\langle \text{module Reuse Stmt} \rangle \rightarrow \langle \text{optional} \rangle \text{ USE MODULE ID WITH PARAMETERS } \langle \text{idList} \rangle \text{ SEMICOL}$
- $\langle \text{optional} \rangle \rightarrow \text{SQBO} \langle \text{idList} \rangle \text{ SQBC ASSIGN OP}$
- $\langle \text{optional} \rangle \rightarrow \epsilon$
- $\langle \text{idList} \rangle \rightarrow \text{ID} \langle \text{idList_bar} \rangle$
- $\langle \text{idList_bar} \rangle \rightarrow \text{COMMA ID} \langle \text{idList_bar} \rangle$
- $\langle \text{idList_bar} \rangle \rightarrow \epsilon.$
- $\langle \text{expression} \rangle \rightarrow \langle \text{boolean Expr} \rangle \langle \text{remaining Expr} \rangle$
- $\langle \text{remaining Expr} \rangle \rightarrow \langle \text{logical Op} \rangle \langle \text{boolean Expr} \rangle$
- $\langle \text{boolean Expr} \rangle \rightarrow \langle \text{bool Value} \rangle + \langle \text{arithmeti} \rangle$
- $\langle \text{boolean Expr} \rangle \rightarrow \langle \text{arithmeti} \rangle \langle \text{Z} \rangle$
- $\langle \text{remaining Expr} \rangle \rightarrow \epsilon$
- $\langle \text{Z} \rangle \rightarrow \langle \text{relational Op} \rangle \langle \text{arithmeti} \rangle \&$
- $\langle \text{Z} \rangle \rightarrow \epsilon$
- $\langle \text{arithmeti} \rangle \rightarrow \langle \text{unary Expr} \rangle$
- $\langle \text{arithmeti} \rangle \rightarrow \langle \text{non Unary Expr} \rangle$
- $\langle \text{non Unary Expr} \rangle \rightarrow \langle \text{term} \rangle \langle \text{remaining non Unary Expr} \rangle$
- $\langle \text{remaining non Unary Expr} \rangle \rightarrow \langle \text{addSub} \rangle \langle \text{remaining non Unary Expr} \rangle$
- $\langle \text{remaining non Unary Expr} \rangle \rightarrow \langle \text{removing non Unary Expr} \rangle$
- $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \rightarrow \epsilon.$
- $\langle \text{remaining Term} \rangle \rightarrow \langle \text{multDiv} \rangle \langle \text{factor} \rangle \langle \text{remaining term} \rangle$
- $\langle \text{remaining Term} \rangle \rightarrow \epsilon$

- $\langle \text{factor} \rangle \rightarrow \text{BD} \langle \text{non Unary Expr} \rangle \text{BC} \mid \langle \text{var_Numerical} \rangle$
- $\langle \text{Unary Expr} \rangle \rightarrow \langle \text{AddSub} \rangle \langle \text{factor} \rangle$
- $\langle \text{relational Op} \rangle \rightarrow \text{LT} \mid \text{LE} \mid \text{GT} \mid \text{GE} \mid \text{EQ} \mid \text{NE}$
- $\langle \text{logical Op} \rangle \rightarrow \text{AND} \mid \cancel{\text{OR}}$
- $\langle \text{logical Op} \rangle \rightarrow \text{OR}$
- $\langle \text{Add Sub} \rangle \rightarrow \text{PLUS} \mid \cancel{\text{MINUS}}$
- $\langle \text{Add Sub} \rangle \rightarrow \text{MINUS}$
- $\langle \text{multDiv} \rangle \rightarrow \text{MUL}$
- $\langle \text{multDiv} \rangle \rightarrow \text{DIV}$
- $\langle \text{declare Stmt} \rangle \rightarrow \text{DECLARE} \langle \text{idList} \rangle \text{COLON} \langle \text{data Type} \rangle \text{SEMICOL}$
- $\langle \text{condition Stmt} \rangle \rightarrow \text{SWITCH} \text{ BD } \text{FD } \text{BC } \text{START}$
 $\langle \text{caseStmts} \rangle \langle \text{default} \rangle \text{END}$.
- $\langle \text{caseStmts} \rangle \rightarrow \text{CASE} \langle \text{value} \rangle \text{COLON} \langle \text{statements} \rangle$
 $\text{BREAK} \text{ SEMICOL} \langle \text{caseStmt} \rangle$
- $\langle \text{Case Stmt} \rangle \rightarrow \text{CASE} \langle \text{value} \rangle \text{COLON} \langle \text{statements} \rangle$
 $\text{BREAK} \text{ SEMICOL} \langle \text{caseStmt} \rangle$
- $\langle \text{case Stmt} \rangle \rightarrow \emptyset$
- $\langle \text{iterative Stmt} \rangle \rightarrow \text{FOR} \text{ BD } \text{FD } \text{IN} \langle \text{range} \rangle \text{BC } \text{START}$
 $\langle \text{statement} \rangle \text{END}$
- $\langle \text{iterative Stmt} \rangle \rightarrow \text{WHILE} \text{ BD } \langle \text{Boolean Expr} \rangle \text{ BC START}$
 $\langle \text{statements} \rangle \text{END}$
- $\langle \text{value} \rangle \rightarrow \text{NUM}$
- $\langle \text{Value} \rangle \rightarrow \cancel{\text{NUM}} \mid \langle \text{Bool Value} \rangle$
- $\langle \text{default} \rangle \rightarrow \text{DEFAULT} \text{ COLON} \langle \text{statements} \rangle \text{ BREAK}$
 SEMICOL
- $\langle \text{default} \rangle \rightarrow \emptyset$
- $\langle \text{range} \rangle \rightarrow \text{NUM} \text{ RANGEOP } \text{NUM}$.

FIRST SET

GROUP No 19

- FIRST (<program>) = { DECLARE, DEF, DRIVERDEF }
- FIRST (<moduleDeclarations>) = { DECLAR, E }
- FIRST (<moduleDeclaration>) = { DECLAR }
- FIRST (<module>) = { DEF }
- FIRST (<otherModules>) = { DEF, E }
- FIRST (<driverModule>) = { DRIVERDEF }
- FIRST (<set>) = { RETURNS, E }
- FIRST (<input plist>) = { ID }
- FIRST (<output plist>) = { ID }
- FIRST (<input plist box>) = { COMMA, e }
- FIRST (<output plist box>) = { COMMA, E }
- FIRST (<datatype>) = { INTEGER, REAL, BOOLEAN, ARRAY }
- FIRST (<type>) = { INTEGER, REAL, BOOLEAN }
- FIRST (<moduleDef>) = { START }
- FIRST (<statements>) = { E, GET-VALUE, PRINT, IP, SQBO, USE, }
DECLARE, SWITCH, FOR, WHILE, ~~END~~,
~~BREAK~~.
- FIRST (<statement>) = { GET-VALUE, PRINT, ID, SQBO, USE, }
DECLARE, SWITCH, FOR, WHILE, ~~END~~,
~~BREAK~~.
- FIRST (<iostmt>) = { GET-VALUE, PRINT }
- FIRST (<whichID>) = { SQBO, E }
- FIRST (<simplestmt>) = { ID, USE, SQBO }
- FIRST (<var>) = { ID, NUM, RNUM, TRUE, FALSE }
- FIRST (<var-id-num>) = { ID, NUM, RNUM }
- FIRST (<boolConst>) = { TRUE, FALSE }
- FIRST (<assignmentstmt>) = { ID }
- FIRST (<whichstmt>) = { ASSIGNOP, SQBO }
- FIRST (<lvalueIDstmt>) = { ASSIGNOP }

- FIRST(<value ARRStut>) = { \$QBO }
- FIRST(<Index>) = { NUM, ID }
- FIRST(<moduleReuse Stut>) = { \$QBO, USE }
- FIRST(<optional>) = { \$QBO }
- FIRST(<idList>) = { ID }
- FIRST(<idList_bar>) = { COMMA, E }
- FIRST(<relationOp>) = { GT, GE, LT, LE, EQ, NE }
- FIRST(<logicalOp>) = { AND, OR }
- FIRST(<factor>) = { BO, ID, NUM, RNUM }
- FIRST(<term>) = { BO, ID, NUM, RNUM }
- FIRST(<expression>) = { PLUS, MINUS, BO, ID, NUM, RNUM, TRUE, FALSE }
- FIRST(<booleanExp>) = { PLUS, MINUS, BO, ID, NUM, RNUM, TRUE, FALSE }
- FIRST(<arithmeticExp>) = { PLUS, MINUS, BO, ID, NUM, RNUM }
- FIRST(<addSub>) = { PLUS, MINUS }
- FIRST(<multDiv>) = { MUL, DIV }
- FIRST(<unary Exp>) = { PLUS, MINUS }
- FIRST(<remaining Term>) = { MUL, DIV, E }
- FIRST(<remaining Exp>) = { AND, OR, E }
- FIRST(<nonUnaryExp>) = { BO, ID, NUM, RNUM }
- FIRST(<remainNonUnaryExp>) = { PLUS, MINUS, E }
- FIRST(<remaining Term>) = { MUL, DIV, E }
- FIRST(<z>) = { GT, GE, LT, LE, EQ, NE, E }

~~FIRST~~

- FIRST(<range>) = { NUM }
- FIRST(<value>) = { NUM, TRUE, FALSE }
- FIRST(<default>) = { DEFAULT, E }
- FIRST(<iterativeStut>) = { FOR, WHILE }
- FIRST(<conditionStut>) = { SWITCH }
- FIRST(<declareStut>) = { DECLARE }
- FIRST(<caseStuts>) = { CASE }
- FIRST(<caseStut>) = { CASE, E }

Follow sets

Group-19

- Follow (<s'>) = { \$ }
- Follow (<program>) = { \$ }
- Follow (<module declarations>) = { DEF, DRIVERDEF }
Follow (<module Declaration>) = { DECLARE, DEF, DRIVERDEF }
- Follow (<other Modules>) = { DRIVERDEF, \$ }
- Follow (<driver module>) = { DEF, \$ }
- Follow (<module>) = { DEF, DRIVERDEF, \$ }
- Follow (<ret>) = { STARTY }
- Follow (<input_list>) = { SQBC }
- Follow (<A'>) = { SQBC }
- Follow (<output_list>) = { SQBC }
- Follow (<B'>) = { GQBC }
- Follow (<dataType>) = { SEMIOL, SQBC }
- Follow (<Type>) = { SQBC, SEMIOLY }
- Follow (<moduleDef>) = { DEF, \$, DRIVERDEF }
Follow (<statements>) = { END, BREAK }
- Follow (<statement>) = { GET-VALUE, PRINT, ID, SQBO, USE, DECLARE, SWITCH, FOR, WHILE, END, BREAK }
- Follow (<io statement>) = { KEY-VALUE, PRINT, ID, SQBO, USE, DECLARE, SWITCH, FOR, WHILE, END, BREAK }
- Follow (<simple Statement>) = { GETVALUE, PRINT, ID, SQBO, USE, DECLARE, SWITCH, FOR, WHILE, END, BREAK }
Follow (<Assignment Stmt>) = { GET-VALUE, PRINT, ID }
- Follow (<Statement>) = Follow (<which Stmt>) =
Follow (<value ID Stmt>) = Follow (<value ARR>
= Follow (<value ARR Stmt>)).
- Follow (<index>) = { SQBC }
- Follow (<Module Reuse Stmt>) = Follow (<simple Stmt>)

- follow (<optional>) = $\{\text{U}\}$
- follow (<id List>) = $\{\text{S}\}$
- follow (<I>) = follow (<id List>) = $\{\text{Q}\}$
- follow (<remaining Expr>) = $\{\text{SEMICOL}\}$
- follow (<Z>) = $\{\text{SEMICOL, AND, OR}\}$
- follow (<remaining Unary Expr>) = $\{\text{B C, ET, GE, LT, LE, NE, EQ, SEMICOL, AND, OR}\}$
- follow (<remaining Term>) = $\{\text{PLUS, MINUS, BC, ERT, GE, LT, LE, EQ, NE, SEMICOL, AND, OR}\}$
- follow (<range>) = $\{\text{B C, QBC}\}$
- follow (<value>) = $\{\text{COLON}\}$
- follow (<default>) = $\{\text{END}\}$
- follow (<case-stmt>) = $\{\text{DEFAULT, END}\}$
- follow (<declare-stmt>) = follow (<iterative-stmt>)
 = follow (<conditional-stmt>) = follow (<statement>)
 = $\{\text{GET-VALUE, PRINT, ID, SQBD, USE, DECLARE, SWITCH, FOR, WHILE, END, BREAK}\}$.

NOTE :- set of Non-terminals used in the grammar are written in FIRST SET.

Assumption :- The input string of the form
 $\langle\text{arithmetic Exp}\rangle \langle\text{logical Op}\rangle \langle\text{arithmetic Exp}\rangle$
is accepted by our Grammar.
It is assumed that relevant error will
be generated at ~~syn~~ Semantic Analysis
phase.