# BNF

```
start -> program id program_begin stmt* program_end
stmt -> ifStmt | whileStmt | assignStmt | defineStmt | atomicStmt

ifStmt -> ifStmtTmp | ifStmtTmp else begin stmt* end
ifStmtTmp -> ifStmtTmp elseif ( exp ) begin stmt* end | if ( exp ) begin stmt* end

whileStmt -> while ( exp ) begin stmt* end
forStmt -> for ( integer id = exp ; for_cond ; for_update) begin stmts end
for_cond -> simple_exp compop simple_exp
for_update -> exp op exp | exp ++ | exp --

assignStmt -> id op exp ;

defineStmt -> integer defineStmtTmp ;
defineStmtTmp -> defineStmtTmp , id | defineStmtTmp , id = exp | id | id = exp

atomicStmt -> continue; | break; | display( ); | display ( stringLiteral ) ;

exp -> exp compop simple_exp | simple_exp
simple_exp -> simple_exp addop term | term
term -> term mulop factor | factor
factor -> ( exp ) | number | id
compop -> < | > | <= | >= | ==
addop -> + | -     mulop -> * | /    op -> = | += | -= | *= | /=
```

BNF without
left recursion

```
start -> program id program_begin stmts program_end

stmt -> ifStmt | whileStmt | assignStmt | defineStmt | atomicStmt | forStmt
stmts -> stmt stmts | ε

ifStmt -> if ( exp ) begin stmts end elseif_part else_part
elseif_part -> elseif ( exp ) begin stmts end elseif_part | ε
else_part -> else begin stmts end | ε

whileStmt -> while ( exp ) begin stmts end
forStmt -> for ( integer id = exp ; for_cond ; for_update) begin stmts end
for_cond -> simple_exp compop simple_exp
for_update -> exp for_option
for_option -> op exp | ++ | --

assignStmt -> id op exp ;

defineStmt -> integer declarators ;
declarators -> id option declarators’
declarators’ -> , id option variable_declarators’ | ε
option -> = exp | ε

atomicStmt -> continue; | break; | display ( disoption ) ;
disoption -> stringLiteral | ε
```

BNF without
left recursion

exp -> simple_exp exp’
exp’ -> comop simple_exp exp’ | $\varepsilon$
simple_exp -> term simple_exp’
simple_exp’ -> addop term simple_exp’ | $\varepsilon$
term -> factor term’
term’ -> mulop factor term’ | $\varepsilon$
factor -> ( exp ) | number | id

compop -> < | > | <= | >= | ==
addop -> + | -
mulop -> * | /
op -> = | += | -= | /= | *=

# First set

| | |
|---|---|
| start | program |
| stmt | if, while, for, id, integer, continue, break, display |
| stmts | if, while, for, id, integer, continue, break, display, ε |
| ifStmt | if |
| elseif_part | elseif, ε |
| else_part | else, ε |
| whileStmt | while |
| forStmt | for |
| for_cond | (, number, id |
| for_update | (, number, id |
| for_option | ++, --, =, -=, +=, /=, *= |
| assignStmt | id |
| defineStmt | integer |
| declarators | id |
| declarators’ | , , ε |
| atomicStmt | continue, break, display |
| option | =, ε |

| | |
|---|---|
| exp | (, number, id |
| exp’ | <, >, <=, >=, ==, ε |
| simple_exp | (, number, id |
| simple_exp’ | +, -, ε |
| term | (, number, id |
| term’ | *, /, ε |
| factor | (, number, id |
| compop | <, >, <=, >=, == |
| addop | +, - |
| mulop | *, / |
| op | =, +=, -=, *=, /= |

# Follow set

| | |
|---|---|
| start | $ |
| stmt | if, while, for, id, integer, continue, break, display, end, program_end |
| stmts | end, program_end |
| ifStmt | Follow(stmt) |
| elseif_part | else, follow(stmt) |
| else_part | Follow(stmt) |
| whileStmt | Follow(stmt) |
| forStmt | Follow(stmt) |
| for_cond | ; |
| for_update | ) |
| for_option | ) |
| assignStmt | Follow(stmt) |
| defineStmt | Follow(stmt) |
| declarators | ; |
| declarators’ | ; |
| atomicStmt | Follow(stmt) |
| option | ,, ; |

| | |
|---|---|
| exp | ), ;, ,, ++, --, +=, -=, /=, *= |
| exp’ | Follow(exp) |
| simple_exp | ), ;, ,, ++, --, +=, -=, /=, *= , <, >, <=, >=, == |
| simple_exp’ | Follow(simple_exp) |
| term | ), ;, ,, ++, --, +=, -=, /=, *= , <, >, <=, >=, ==, +, - |
| term’ | Follow(term) |
| factor | ), ;, ,, ++, --, +=, -=, /=, *= , <, >, <=, >=, ==, +, -, *, / |
| compop | ==, <, >, <=, >= |
| addop | +, - |
| mulop | *, / |
| op | =, +=, -=, /=, *= |

| M[N,T] | ( | number | id | ) | + | - | * | / | > | < | >= | <= | == | ; | , | += | -= | /= | *= | = |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| exp | exp -> simple_exp exp’ | exp -> simple_exp exp’ | exp -> simple_exp exp’ | | | | | | | | | | | | | | | | | |
| exp’ | | | | exp’ -> ε | | | | | exp’ -> compop simple_exp exp’ | exp’ -> compop simple_exp exp’ | exp’ -> compop simple_exp exp’ | exp’ -> compop simple_exp exp’ | exp’ -> compop simple_exp exp’ | exp’ -> ε | exp’ -> ε | exp’ -> ε | exp’ -> ε | exp’ -> ε | exp’ -> ε | exp’ -> ε |
| simple_exp | simple_exp -> term simple_exp’ | simple_exp -> term simple_exp’ | simple_exp -> term simple_exp’ | | | | | | | | | | | | | | | | | |
| simple_exp’ | | | | simple_exp’ -> ε | simple_exp’ -> addop term simple_exp’ | simple_exp’ -> addop term simple_exp’ | | | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε | simple_exp’ -> ε |
| term | term -> factor term’ | term -> factor term’ | term -> factor term’ | | | | | | | | | | | | | | | | | |
| term’ | | | | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> mulop factor term’ | term’ -> mulop factor term’ | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε | term’ -> ε |
| factor | factor -> ( exp ) | factor -> number | factor -> id | | | | | | | | | | | | | | | | | |
| compop | | | | | | | | | compop -> > | compop -> < | compop -> >= | compop -> <= | compop -> == | | | | | | | |
| addop | | | | | addop -> + | addop -> - | | | | | | | | | | | | | | |
| mulop | | | | | | | mulop -> * | mulop -> / | | | | | | | | | | | | |
| op | | | | | | | | | | | | | | | | op -> += | op -> -= | op -> /= | op -> *= | op -> = |

| M[N,T] | program | id | program_end | end | if | elseif | else | while | for | integer | continue | break | display | ; | , | = |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| start | start -> program id program_begin stmts program_end | | | | | | | | | | | | | | | |
| stmt | | stmt -> assignStmt | | | stmt -> ifStmt | | | stmt -> whileStmt | stmt -> forStmt | stmt -> defineStmt | stmt -> atomicStmt | stmt -> atomicStmt | stmt -> atomicStmt | | | |
| stmts | | stmts -> stmt stmts | stmts -> ε | stmts -> ε | stmts -> stmt stmts | | | stmts -> stmt stmts | stmts -> stmt stmts | stmts -> stmt stmts | stmts -> stmt stmts | stmts -> stmt stmts | stmts -> stmt stmts | | | |
| ifStmt | | | | | ifStmt -> if ( exp ) begin stmts end elseif_part else_part | | | | | | | | | | | |
| elseif_part | | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> elseif ( exp ) begin stmts end elseif_part | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | | | |
| else_part | | else_part -> ε | else_part -> ε | else_part -> ε | else_part -> ε | | else_part -> else begin stmts end | else_part -> ε | else_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | elseif_part -> ε | | | |
| whileStmt | | | | | | | | whileStmt -> while ( exp ) begin stmts end | | | | | | | | |
| assignStmt | | assignStmt -> id = exp ; | | | | | | | | | | | | | | |

| M[N,T] | id | ( | number | ++ | -- | for | integer | continue | break | display | ; | , | = | ) | stringLiteral |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| defineStmt | | | | | | | defineStmt -> integer declarators ; | | | | | | | | |
| declarators | declarators -> id option declarators’ | | | | | | | | | | | | | | |
| declarators’ | | | | | | | | | | | declarators’ -> ε | declarators’ -> , id option declarators’ | | | |
| option | | | | | | | | | | | option -> ε | option -> ε | option -> = exp | | |
| atomicStmt | | | | | | | | atomicStmt -> continue ; | atomicStmt -> break ; | atomicStmt -> display ( disoption ) ; | | | | | |
| disoption | | | | | | | | | | | | | | disoption -> ε | disoption -> stringLiteral |
| forStmt | | | | | | forStmt -> for ( integer id = exp ; for_cond ; for_update) begin stmts end | | | | | | | | | |
| for_cond | for_cond -> simple_exp compop simple_exp | for_cond -> simple_exp compop simple_exp | for_cond -> simple_exp compop simple_exp | | | | | | | | | | | | |
| for_update | for_update -> exp for_option | for_update -> exp for_option | for_update -> exp for_option | | | | | | | | | | | | |
| for_option | | | | for_option -> ++ | for_option -> -- | | | | | | | | for_option -> operator exp | | |