1. program → declaration-list|comment| include\_command ###########

2. declaration-list → declaration-list declaration (left recursion) | declaration

2-declaration-list ->declaration declaration-list`###########

3-declaration-list`->declaration declaration-list`|epsilon#############

3. declaration → type-specifier ID declaration` ##########

declaration`-> ;| ( params ) compound-stmt| comment #### type-specifier ID

4. var-declaration → type-specifier ID ; ##########

5. type-specifier → Iow | SIow | Chlo | Chain | Iowf | SIowf | Worthless#######

6. fun-declaration → type-specifier ID fun-declaration`

fun-declaration`->( params ) compound-stmt|

comment #######type-specifier ID

*//fun-declaration → type-specifier* ***ID*** *( params ) compound-stmt| comment type-specifier* ***ID//***

7. params → param-list | Worthless | e ###########

8. param-list → param-list (left recursion), param | param

param list->param param-list`#########

param\_list`-> ,param param-list`|epsilon#########

9. param → type-specifier ID#########

10. compound-stmt → {comment local-declarations statement-list } | {local-declarations statement-list } (left factoring)

11.compound-stmt → { compound-stmt` ##########

12.compund-stmt ` -> comment local-declarations statement-list} |

local-declarations statement-list} #############

11. local-declarations →local-declarations (left recursion) var-declaration | e

local\_declaration-> local-declaration` ###

local declaration`-> var-declaration local-declaration` | e####

12. statement-list → statement-list (left recursion) statement | e

statement-list-> statment-list`###########

statement-list`-> statement statment-list` |e############

13. statement → expression-stmt | compound-stmt | selection-stmt |

iteration-stmt | jump-stmt #############

14. expression-stmt → expression ; | ; #############

15. selection-stmt → if(left factoring ) ( expression ) statement | if ( expression ) statement else statement

selection-stmt->if ( expression ) statement selection-stmt`

selection-stmt` → else statement | epsilon############

16. iteration-stmt → Loop-statement | Iterate-statement ###############

17. Loop-statement → Loopwhen( expression ) statement #############

18. Iterate -statement → Iteratewhen ( expression ; expression ; expression

) statement################

19. jump-stmt → Turnback expression ; | Stop ;#################

20. expression → id-assign =expression` ######

expression` -> simple-expression | id-assign (5\*4)+x==x-5

expression->id-assign expression`|simple-expression

expression`->=expression|e ####

21. id-assign → ID ##########

22. simple-expression

→ additive-expression (left factoring) relop additive-expression | additive-expression

simple-expression → additive-expression simple-expression`

simple-expression`->relop additive-expression |epsilon########

23. relop → <= | < | > | >= | == | != | && | ||########

we already handle it in scanner

24. additive-expression → additive-expression (left recursion)addop term | term

additive-expression->term additive-expression`

additive-expression`-> addop term additive-expression`|epsilon

25. addop → + | - ############

26. term → term(left recursion) mulop factor | factor

term->factor term`

term`->mulop factor term`| epsilon

27. mulop → \* | / ################

28. factor → ( expression ) | num |ID factor`

factor`->(args) |e ############

29. call → ID ( args ) ################

30. args → arg-list | e ############

31. arg-list → arg-list (left recursion), expression | expression

arg-list-> expression arg-list`

arg-list`-> , expression arg-list` | epsilon#############

32. num → Signed num |Unsigned num#########

33. Unsigned num → value########

34. Signed num → pos-num | neg-num#########

35. pos-num → + value##########

36. neg-num → - value###########

37. value → INT\_NUM | FLOAT\_NUM############

38. comment →/$ STR $/ | $$$ STR##########

39. include\_command →include (F\_name.txt); ##########

40. F\_name →STR##############