Team 5- 3rd year- Tiny Parser

NonTerminals:

Program, Function_Statements, DataType, Function_Statement ,Function_Statements _, Function_Declaration, Function_Body, Function_Declaration, Parameters, Parameter, Param, Set_Of_Statements , Return_Statement , Write_Statement, Declaration_Statement, Read_Statement, Repeat_Statement, Idlist, ID, Id_list, ID_, Write_St, Condition_Statement, Condition, Con_st, Boolean_Operator, Condition_Operator, Term, Term_, Arguments, Argument, Else_If_Statment, Else_Statment, Assignment_ Function_Call, Expression, Equation, Equation_, ArithOp

CFG:

1. **Program → Function_Statements**
2. **DataType →** int | float | string
3. **Function_Statements → Function_Statement**
   Function_Statements _
4. **Function_Statements _ → Function_Statements | ε**
5. **Function_Statement → Function_Declaration Function_Body**
6. **Function_Declaration → DataType** identifier (**Parameters**)
7. **Parameters →Parameter Param | ε**
8. **Param → ,Parameters| ε**
9. **Parameter → DataType** identifier
10. **Function_Body → { Set_Of_Statements**
    Return_Statement }
11. **Set_Of_Statements →**comment **Set_Of_Statements |**
    Declaration_Statement Set_Of_Statements |
    Write_Statement Set_Of_Statements |
    Read_Statement Set_Of_Statements |
    Return_Statement Set_Of_Statements |
    Repeat_Statement Set_Of_Statements | If_Statment
    Set_Of_Statements |
    identifier **Assignment_ Function_Call**
    Set_Of_Statements | ε
12. **Declaration_Statetment → DataType Idlist;**
13. **Idlist →ID Id_list**
14. **Id_list→ , ID Id_list| ε**
15. **ID→**identifier **ID_**
16. **ID_→:=Expression| ε**
17. **Write_Statement →** write **Write_St** ;
18. **Write_St → Expression |** endl
19. **Read_Statement →** read identifier ;
20. **Return_Statement →** return **Expression** ;

21. **Repeat_Statement** → repeat **Set_Of_Statements**   until **Condition_Statement**

22. **Condition_Statement** → **Condition   Con_st**

23. **Con_st** → **Boolean_Operator Condition Con_st |** ε

24. **Condition** → identifier **Condition_Operator Term**

25. **Condition_Operator** → < **|** > **|** = **|** <>

26. **Boolean_Operator** → && **|** | **|**

27. **Term** → number | identifier **Term_**

28. **Term_** → (**Arguments**);**|** ε

29. **Arguments** → identifier **Argument**

30. **Argument** → , identifier **Argument** | ε

31. **If_Statment** → if **Condition_Statement** then

   Set_Of_Statements Else_If_Statment

   Else_Statment end

32. **Else_If_Statment** → elseif **Condition_Statement**   then
   Set_Of_Statements   Else_If_Statment| ε

33. **Else_Statment** → else **Set_Of_Statements** | ε

34. **Assignment_ Function_Call** → := **Expression;**
   |(**Arguments**);

35. **Expression** → string | **Term Equation_|**
   (**Equation**)**Equation_**

36. **Equation** → (**Equation**)**Equation_|** **Term Equation_**

37. **Equation_** → **ArithOp Equation** | ε

38. **ArithOp** → + | - | / | *