**Task(1)**

Digits = [0-9]

Nat =Digits Digits\*

DQ = \”

SpecialChars ={ ! , @ , # , $ , & , ( , ) , - , ‘ , . , / , + }

Alphabet = ([a-z]|[A-Z])

1. Number = Nat ( “.” Nat)?
2. String =DQ ( SpecialChars |Alphabet | Digits )\* DQ
3. Reserved\_Keywords = ( int | float | string | read | write | repeat | until | if | elseif | else | then | return | endl )
4. Comment\_Statment = / \\* (DQ | SpecialChars |Alphabet | Digits )\* / \\*
5. Identifiers = Alphabet (Alphabet | Digits | “\_” )\*
6. Function\_Call = Identifiers “(“ (Identifiers ( “,” Identifiers)\* )? “)”
7. Term = ( Number | Identifiers | Function\_Call)
8. Arithmatic\_Operation = ( + | - | \* | / )
9. Equation = ( (Term ( Arithmatic\_Operation Term)+ ) ) | ( “(” Term ( Arithmatic\_Operation Term)+ ) “)” ) )+ 🡪 (Term|”(” (Term Arithmatic\_Operation)\* Term “)”) ((Arithmatic\_Operation Term)| (Arithmatic\_Operation”(” (Term Arithmatic\_Operation)\* Term “)”))\*
10. Expression = ( string | Term | Equation )
11. Assignment\_statement = identifier “:=” Expression
12. DataType = ( int | float | string )
13. Declaration\_statement = DataType ( ( identifiers | Assignment\_statement ) ( “,” ( identifiers | Assignment\_statement? ))\* “;”
14. Write\_Statement = “write” (Expression | “endl”) “;”
15. Read\_statement = “read” identifiers “;”
16. Return\_statement = “return” Expression “;”
17. Condition\_Operator = ( “<” | “>” | “=” | “<>” )
18. Condition = identifiers Condition\_Operator Term
19. Boolean\_operator = ( “&&” | “||” )
20. Condition\_statement = condition ( Boolean\_operator condition )\*
21. If\_Statement = “if” Condition\_Statement “then” STATEMENT\* ( Else\_If\_Statment | Else\_Statment | “end” )\*

STATEMENT = ( Condition\_statement | Write\_Statement | Read\_statement | Return\_statement | Declaration\_statement | Assignment\_statement )

1. Else\_If\_Statment = “elseif” ” Condition\_Statement “then” STATEMENT\* ( Else\_If\_Statment | Else\_Statment | “end” )\*
2. Else\_Statement = “else” STATEMENT\* “end”
3. Repeat\_Statement = “repeat” STATEMENT\* “until” Condition\_statement
4. FunctionName = identifiers
5. Parameter = DataType identifires
6. Function\_Declaration = DataType FunctionName “(“ (Parameter ( “,” Parameter )\* )? “)”
7. Function\_Body= “{” STATEMENT\* Return\_statement “}”
8. Function\_Statement = Function\_Declaration Function\_Body
9. Main\_Function = Datatype “main” “()” Function\_Body
10. Program = Function\_Statement\* Main\_Function