1. <prog> -> HEADER <prog_con>
2. <prog_con> -> func FID (<arg>) <types> {<statement_with_return>} <prog_con>
3. <prog_con> -> <variant> ID <option> <prog_con>
4. <prog_con> -> <id_type> <prog_con>
5. <prog_con> -> if <expression> {<statement>} <else_statement> <prog_con>
6. <prog_con> -> while <expression> {<statement>} <prog_con>
7. <prog_con> -> Ɛ EOF
8. <arg> -> <param_name> ID : <type_spec> <args_more>
9. <arg> -> Ɛ
10. <args_more> -> , <param_name> ID : <type_spec> <args_more>
11. <args_more> -> Ɛ
12. <types> -> -> <type_spec>
13. <types> -> Ɛ
14. <type_spec> -> Double
15. <type_spec> -> Int
16. <type_spec> -> String
17. <type_spec> -> Double?
18. <type_spec> -> Int?
19. <type_spec> -> String?
20. <variant> -> let
21. <variant> -> var
22. <option> -> : <type_spec> <assigns>
23. <option> -> <assign>
24. <assigns> -> <assign>
25. <assigns> -> Ɛ
26. <assign> -> = <expression>
27. <assign> -> = <id_type>
28. <id_type> -> <builtin>
29. <id_type> -> ID <preopt>
30. <builtin> -> readString ()
31. <builtin> -> readInt ()
32. <builtin> -> readDouble ()
33. <builtin> -> Int2Double (<term>)
34. <builtin> -> Double2Int (<term>)
35. <builtin> -> length (<length_t>)
36. <builtin> -> substring (of <length_t>, startingAt <term>, endingBefore <term>)
37. <builtin> -> ord (<length_t>)
38. <builtin> -> chr (<chr_t>)
39. <preopt> -> (<params>)
40. <preopt> -> Ɛ
41. <length_t> -> “STRINGLIT”
42. <length_t> -> ID //type STRING
43. <chr_t> -> INTLIT
44. <chr_t> -> ID //type INT
45. <else_statement> -> else {<statement>}
46. <statement_with_return> ->
47. <statement> -> <variant> ID <option> <statement>
48. <statement> -> <id_type> <statement>
49. <statement> -> if <expression> {<statement>} <else_statement> <statement>
50. <statement> -> while <expression> {<statement>} <statement>
51. <statement> -> return <return_value>
52. <statement> -> eps
53. <return_value>!!!!!!!!! ->