

1. start --> prog body END
2. prog --> PROGRAM var
3. body --> declpart procpart execpart
4. declpart execpart
5. declpart --> DECLARE decllist END
6. decllist --> decllist-
7. decllist- --> decllist- declstat ;
8. declstat ;
9. declstat --> declstat-
10. declstat- --> declstat- , var
11. type var
12. type var int
13. type var int :: int
14. type --> INTEGER
15. REAL
16. procpart --> proclist
17. proclist --> proclist proc
18. proc
19. proc --> prothead declpart statlist END
20. prothead statlist END
21. prothead --> procname null-list
22. procname fparmlist
23. procname --> PROCEDURE var
24. null-list --> { }
25. fparmlist --> fparmlist- }
26. fparmlist- --> fparmlist- , calltype type var
27. fparmlist- , calltype type var int
28. fparmlist- , calltype type var int :: int
29. { calltype type var
30. { calltype type var int
31. { calltype type var int :: int
32. calltype --> VALUE
33. REFERENCE
34. execpart --> exechead statlist END

```

35. exechead --> MAIN

36. statlist --> statlist-

37. statlist- --> statlist- stat
38.          stat

39. stat --> whilestat ;
40.     ifstat ;
41.     assignstat ;
42.     inputstat ;
43.     outputstat ;
44.     callstat ;

45. inputstat --> inputstat-

46. inputstat- --> inputstat- , var
47.     inputstat- , var [ aexpr ]
48.     inputstat- , var [ aexpr : aexpr ]
49.     INPUT var
50.     INPUT var [ aexpr ]
51.     INPUT var [ aexpr : aexpr ]

52. outputstat --> outputstat-

53. outputstat- --> outputstat- , var
54.     outputstat- , constant
55.     outputstat- , var [ aexpr ]
56.     outputstat- , var [ aexpr : aexpr ]
57.     OUTPUT var
58.     OUTPUT constant
59.     OUTPUT var [ aexpr ]
60.     OUTPUT var [ aexpr : aexpr ]

61. callstat --> callname null-list
62.     callname aparmlist

63. callname --> CALL var

64. aparmlist --> aparmlist- }

65. aparmlist- --> aparmlist- , calltype var
66.     aparmlist- , constant
67.     aparmlist- , calltype var [ aexpr ]
68.     aparmlist- , calltype var [ aexpr : aexpr ]
69.     { calltype var
70.     { constant
71.     { calltype var [ aexpr ]
72.     { calltype var [ aexpr : aexpr ]

73. ifstat --> ifhead statlist END

```

74. ifthen statlist END

75. ifthen --> ifhead statlist ELSE

76. ifhead --> IF (bexpr) THEN

77. whilestat --> whilehead statlist END

78. whilehead --> WHILE (bexpr) DO

79. assignstat --> astat

80. astat --> astat-

81. astat- --> var <- aexpr

82. var [aexpr] <- aexpr

83. var [aexpr : aexpr] <- aexpr

84. bexpr --> oexpr

85. oexpr --> oexpr || andexpr

86. andexpr

87. andexpr --> andexpr-

88. andexpr- --> andexpr- && notexpr

89. notexpr

90. notexpr --> ! reexpr

91. reexpr

92. reexpr --> aexpr < aexpr

93. aexpr <= aexpr

94. aexpr > aexpr

95. aexpr >= aexpr

96. aexpr == aexpr

97. aexpr != aexpr

98. aexpr

99. aexpr --> aexpr-

100. aexpr- --> aexpr- + term

101. aexpr- - term

102. - term

103. term

104. term --> term-

105. term- --> term- * primary

106. term- / primary

107. primary

- 108. primary --> (bexpr)
- 109. constant
- 110. var
- 111. var [aexpr]
- 112. var [aexpr : aexpr]

- 113. constant --> int
- 114. real