

```

1  /*program*/
2  program->function | e
3
4
5  /*function*/
6  function -> "function""identifier"";""beginparams"declaration"endparams""beginlocals""endlocals""beginbody"statement"endbody"
7
8  /*declararion*/
9  declaration->d";"declaration | e
10 d->i:"""array""["""number""]"""of""integer" | i:"""integer"
11 i -> "identifier":." | "identifier", "i
12
13 /*statement*/
14 statement->s";"statement | s
15 s->var":="expression
16     | "if"bool-expr"then" statement"else"statement"endif"
17     | "while"bool-expr"beginloop"statement"endloop"
18     | "do""beginloop"statement"endloop""while"bool-expr
19     | "read"v
20     | "write"v
21     | "break"
22     | "return"expresstion
23 v->var | var", "v
24
25 /*bool-expr*/
26 bool-expr->relation-and-expr| relation-and-expr"or"bool-expr
27
28 /*relation-and-expr*/
29 relation-and-expr -> relation-expr|relation-expr"or"relation-and-expr
30
31 /*relation-expr*/
32 relation-expr -> "not"r|r
33 r-> expresstion comp expression
34     | true
35     | false
36     | "("bool-expr")"
37
38 /*comp*/
39 comp -> "=="
40     | "<>"
41     | "<"
42     | ">"
43     | "<="
44     | ">="
45
46 /*expression*/

```

```
47 expression -> multiplicative-expr|multiplicative-expr"+"expression|multiplicative-expr "-"expression
48
49 /*multiplicative-expr*/
50 multiplicative-expr -> term|term"*"multiplicative-expr|term"/"multiplicative-expr|"%"multiplicative-expr
51
52 /*term*/
53 term -> "-"t
54         |t
55         |"identifier"("expression")
56         t -> var
57         |"number"
58         |("expression")
59
60 /*var*/
61 var -> "identifier"
62       |"identifier"["expression"]
63
```