

```
1  pgm -> d-list
2  d-list -> dclartn A
3  A -> dclartn A | @
4  dclartn -> type-specfr id B
5  B -> var-dclartn | fun-dclartn
6  var-dclartn -> ; | [ num ] ;
7  fun-dclartn -> ( params ) compnd-stmt
8  type-specfr -> int | void | float
9  params -> param-list | void
10 param-list -> param D
11 D -> , param D | @
12 param -> type-specfr id E
13 E -> [ ] | @
14 compnd-stmt -> { local-dclartn statmnt-list }
15 local-dclartn -> F
16 F -> type-specfr id var-dclartn F | @
17 statmnt-list -> G
18 G -> statmnt G | @
19 statmnt -> exprsn-stmt OR -> selctn-stmt OR -> itertn-stmt OR -> retrn-
    stmt OR -> compnd-stmt
20 exprsn-stmt -> exprsn ; | ;
21 selctn-stmt -> if ( exprsn ) statmnt H
22 H -> else statmnt | @
23 itertn-stmt -> while ( exprsn ) statmnt
24 retrn-stmt -> return exprsn-stmt
25 exprsn -> ( exprsn ) three , exprsn -> num three , exprsn -> id two
26 two -> one relop two2 OR -> ( args ) three OR -> three
27 three -> temp3 temp2 temp
28 temp3 -> @ | mulop factor temp3
29 temp2 -> @ | addop term temp2
30 temp -> @ | relop addexp temp
31 two2 -> three | exprsn
32 relop -> @ | < L | > L | = L | !=
33 L -> = | @
34 addexp -> term temp2
35 addop -> + | -
36 term -> factor temp3
37 mulop -> * | /
38 factor -> ( exprsn 0 ) OR -> id factor | id one OR -> num
39 one -> @ | [ exprsn ]
40 args -> @ | arg-list
41 arg-list -> exprsn 0
42 0 -> @ | , exprsn 0
```