function ::= B {statemt}

statemt ::= assnmt | ifstmt | loop | read | output | funcall

assnmt ::= ident ~ exprsn ;

ifstmt ::= I comprsn @ {statemt} [% {statemt}] &

loop ::= W comprsn L {statemt} T

read ::= R ident {, ident } ;

output ::= O ident {, ident } ;

funcall ::= C function E

comprsn ::= ( oprnd opratr oprnd )

exprsn ::= factor {+ factor}

factor ::= oprnd {\* oprnd}

oprnd ::= integer | ident | ( exprsn )

opratr ::= < | = | > | !

ident ::= letter {char}

char ::= letter | digit

integer ::= digit {digit}

letter ::= X | Y | Z

digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7