```
<program> -> DECLARATION <declarations> MAIN <statements> END

<declarations> -> <declaration> | <declaration> <declarations>

<declaration> -> <type> <variable> “!” | <comment>

<type> -> Z | Q | CHAR

<variable> -> <letter> | “_” {<letter> | <digit> | “_”}

<letter> -> [a-z]

<digit> -> [0-9]

<comment> -> # {.} #

<statements> -> <statement> | <statement> <statements>

<statement> -> <assignment> | <conditional> | <loop_construction>
              | EXIT “!” | OUT <or_expr> “!” | <comment>

<assignment> -> <variable> “:=” (<or_expr> | IN) “!”

<or_expr> -> <and_expr> {“\/” <and_expr>}

<and_expr> -> <comp> {“/\” <comp>}

<comparison> -> <expression>
                {(“<” | “>” | “<=” | “>=” | “=” | “X=”) <expr>}

<expr> -> <term> {(“+” | “-”) <term>}

<term> -> <factor> {(“*” | “:”) <factor>}

<factor> -> “(” <or_expr> “)” | <variable> | <const>

<const> -> <integer> | <float> | <char>

<integer> -> [(+ | -)] <digit> {<digit>}

<float> -> [(+ | -)] <digit> {<digit>} “.” <digit> {<digit>}

<char> -> “‘“ . “‘“

<conditional> -> “[” <expr> “]” “->” <statements>
                [“--” <statements>] DONE

<loop_construction> -> WHENEVER “[” <expr> “]” <statements> DONE
```