program := statement\_list

statement\_list := statement | statement "\n" statement

statement := (declaration | assignment | io| if\_stmt | while\_stmt | for\_stmt)

expression ::= expression + term | expression - term | term

term ::= term \* factor | term / factor | factor

factor ::= "(" expression ")" | ID | CONST

type := primitive | array\_decl

primitive := "int" | "string"

array\_decl := "list of " primitive

assignment := ID "=" expression

io := "read(" ID “)” | "write(" (ID|CONST) “)”

if\_stmt := "if " condition " then " statement\_list "end" | "if " condition " then " statement\_list "else" statement\_list "end"

while\_stmt := "while " condition " do" statement\_list "end"

for\_stmt := "for" for\_header "do" statement\_list "end"

for\_header := ("{" [expression ":"] expression [":" expression] "}") | ID

condition := expression relation expression

relation := "<" | "<=" | "=" | "!=" | ">=" | ">"