program_sym program

identifier example

lparen (

identifier input

comma ,

identifier output

rparen )

semicolon ;

var_sym var

litchar x

comma ,

litchar y

colon :

integer_sym integer

semicolon ;

function_sym function

identifier gcd

lparen (

litchar a

comma ,

litchar b

colon :

integer_sym integer

rparen )

colon :

integer_sym integer

semicolon ;

begin_sym begin

lbrace {

identifier gcd

rbrace }

if_sym if

litchar b

greaterequal >=

illegal 0ba

then_sym then

identifier gcd

assign :=

litchar a

else_sym else

identifier gcd

assign :=

lparen (

litchar b

comma ,

litchar a

mod_sym mod

litchar b

rparen )

end_sym end

semicolon ;

lbrace {

identifier gcd

rbrace }

identifier bedgin

lbrace {

minus -

identifier example

rbrace }

read read

lparen (

litchar x

comma ,

litchar y

rparen )

semicolon ;

identifier wwrite

lparen (

identifier gcd

lparen (

litchar x

comma ,

litchar y

rparen )

rparen )

semicolon ;

end_sym end

period .