G = ( N , E, P, S )

N = {

identifier

lowercase\_letter

uppercase\_letter

digit

integer

nonzero\_digit

arithmetic\_operator

comparison\_operator

assignment\_operator

logical\_operator

userinput\_operator

useroutput\_operator

inc\_identifier

dec\_identifier

subtype

array\_identifier

array\_declaration

type

declaration

declaration\_list

composed\_declaration\_list

composed\_statement

statement

statement\_list

term

factor

expression

assign\_statement

input\_statement

output\_statement

condition

if\_statement

while\_statement

for\_statement

dowhile\_statement

return\_statement

main

program

}

E = {

Int,

Char

If

Do

Else

While

For

Return

Cin

Cout

Main

Array

String

+

-

\*

/

=

==

>=

<=

!=

||

&&

>>

<<

++

==

[

]

{

}

(

)

;

space

,

}

S = program

P = {

program -> main\_statement

main\_statement -> “int” “main” “(“ ”)” “{“ statement\_list return\_statement “}”

statement\_list -> statement { statement\_list }

statement -> assign\_statement | input\_statement | output\_statement |

If\_statement | for\_statement | dowhile\_statement |

composed\_declaration\_list

composed\_declaration\_list -> declaration\_list { composed\_declaration\_list }

declaration\_list -> declaration “;” | declaration { “;” identifier } “;” |

declaration { “;” array\_identifier } “;”

declaration -> type identifier | array\_declaration

type -> subtype | array\_declaration

array\_declaration -> subtype | array\_identifier

subtype -> “int” | “char”

array\_identifier -> identifier “[“ integer “]”

assign\_statement -> identifier assignment\_operator expression “;”

input\_statement -> “cin” userinput\_operator identifier “;”

output\_statement -> “cout” useroutput\_operator string “;” |

“cout” useroutput\_operator identifier “;” |

“cout” useroutput\_operator integer “;”

condition -> expression comparison\_operator expression|

expression logical\_operator expression

if\_statement -> “if” “(“ condition “)” composed\_statement |

“if” “(“ condition “)” composed\_statement “else” composed\_statement

while\_statement -> "while" "(" condition ")" composed\_statement

for\_statement -> “for” “(“ assign\_statement condition “;” inc “)” |

“for” “(“ assign\_statement condition “;” dec “)”

dowhile\_statement -> “do” composed\_statement “while” “(“ condition “)” “;”

expression -> expression “+” term | expression “-” term | term

factor -> “(“ expression “)” | identifier

term -> term \* factor | term / factor | term % factor | factor

return\_statement -> “return” “0” “;”

identifier -> lowercase\_letter {lowercase\_letter} {uppercase\_letter } {digit}

lowercase\_letter -> a | b | … | z

uppercase\_letter -> A | B | ... | Z

digit -> 0 | 1 | ... | 9

integer -> nonzero\_digit {digit} | 0

nonzero\_digit -> 1 |... | 9

arithmetic\_operator -> + | - | \* | /

comparison\_operator -> == | > | < | <= | >= | !=

assignment\_operator -> =

logical\_operator -> || | &&

userinput\_operator -> >>

useroutput\_operator -> <<

inc\_identifier -> ++

dec\_identifier -> --