BNF

<type>::=int | float | boolean | char

<identifier>::=A-Z<seg\_identifier> | a-z<seg\_identifier>

<seg\_identifier>::= A-Z<seg\_identifier> | a-z<seg\_identifier> | 0-9<seg\_identifier> | ε

<var\_declaration>::=<type> <identifier>;

<proc\_wo\_var>::= <for\_structure> <proc\_wo\_var\_aux> | <if\_structure> <proc\_wo\_var\_aux> | <attribution\_structure>;<proc\_wo\_var\_aux> | <write><proc\_wo\_var\_aux> | <read><proc\_wo\_var\_aux>

<proc\_wo\_var\_aux>::= <proc\_wo\_var> | ε

<proc>::=<var\_declaration> <proc> | <for\_structure> <proc> | <if\_structure> <proc> | <attribution\_structure>;<proc> | <write><proc > | <read><proc > | ε

<for\_structure>::=for<identifier> in range (<for\_parameters>){<proc\_wo\_var>}

<for\_parameters>::= <begin\_int> , <final\_int>

<if\_structure>::= if(<condition>){<proc\_wo\_var>} <matched>

<matched>::= else{<proc\_wo\_var>} | ε

<condition> ::= <identifier> <operation> | <number> <relational> <logic\_continue> | not(<logic\_possibilities>) <logic\_continue> | <boolean> <operation>

<logic\_possibilities> ::= <identifier> <operation> | <boolean> <operation > | not(<logic\_possibilities>)

<relational> ::= > <relational\_aux> <id\_num> | < <relational\_aux> <id\_num> | == <id\_num> | != <id\_num>

<relational\_aux>::= = | ε

<id\_num> ::= <identifier> | <number>

<operation> ::= <logic\_continue> | <relational> <logic\_continue>

<logic\_continue> ::= and <condition> | or <condition> | ε

<arithmetic\_operation>::= <arith\_aux><lower>

<lower>::= +<arith\_aux><lower> | -<arith\_aux><lower> | ε

<arith\_aux>::= <factor><higher>

<higher>::= \*<factor><higher> | /<factor><higher> | ε

<factor>::= ( <factor\_aux> ) | - ( <factor\_aux> ) | <identifer> | <number>

<factor\_aux>::= <identifier> | <number>

<attribution\_structure>::=<identifier>=<id\_or\_type\_or\_arith>

<id\_or\_type\_or\_arith> ::= <boolean>|<char>| <arithmetic\_operation>

<begin\_int>::= 0-9 <int\_aux>

<final\_int>::= 0-9 <int\_aux>

<number> ::= <pos\_int> | <neg\_int>

<pos\_int> ::= 0-9 <int\_aux> <float>

<int\_aux> ::= 0-9 <int\_aux> | ε

<float> ::= . <int\_aux> | ε

<neg\_int> ::= - <pos\_int>

<boolean>::=true|false

<char>::= ‘ <char\_aux> ‘

<char\_aux>::= A-Z | a-z | 0-9

<write>::= write(<identifier>);

<read>::= read(<identifier>);