CORE Imperative Language

```
<prog>    ::= program <decl-seq> begin <stmt-seq> end
<decl-seq>::= <decl>
            | <decl> <decl-seq>
<stmt-seq>::= <stmt>
            | <stmt> <stmt-seq>
<decl>    ::= int <id-list>;
<id-list> ::= <id>
            | <id>, <id-list>
<stmt>    ::= <assign>
            | <if>
            | <loop>
            | <in>
            | <out>
<assign>  ::= <id> = <exp>;
<if>      ::= if <cond> then <stmt-seq> end;
            | if <cond> then <stmt-seq> else <stmt-seq> end;
<loop>    ::= while <cond> loop <stmt-seq> end;
<in>      ::= read <id-list>;
<out>     ::= write <id-list>;
<cond>    ::= <comp>
            | !<cond>
            | [ <cond> and <cond> ]
            | [ <cond> or <cond> ]
<comp>    ::= ( <fac> <comp-op> <fac> )
<exp>     ::= <term>
            | <term> + <exp>
            | <term> - <exp>
<term>    ::= <fac>
            | <fac> * <term>
<fac>     ::= <int>
            | <id>
            | ( <exp> )
<comp-op> ::= !=
            | ==
            | <
            | >
            | <=
            | >=
<id>      ::= <let-seq>
            | <let-seq><int>
<let-seq> ::= <let>
            | <let><let-seq>
<let>     ::= A | B | C | ... | X | Y | Z

<int>     ::= <digit>
            | <digit><int>
<digit>   ::= 0 | 1 | 2 | 3 | ... | 9
```