```
<program> ::=   <stmt_list>

<stmt_list> ::=   <stmt>
                | <stmt_list> <stmt>

<stmt> ::= <assign_stmt>
         | <io_stmt>
         | <control_stmt>
         | <loop-stmt>

<assign_stmt> ::= IDENT "=" <expr> ";"

<io_stmt> ::=  "plot" <expr> ";"
            | "ask" IDENT ";"

<loop-stmt> ::= "loop" IDENT "in" <expr> ".." <expr>
                "{" <stmt_list> "}"

<control_stmt> ::= <check-stmt>
                 | <choose_stmt>

<check_stmt> ::= "check" <expr> "{" <stmt_list> "}"
                 "else" "{" <stmt_list> "}"

<choose_stmt> ::= "choose" <expr> "{" <case_list>
                  <default_case> "}"

<case_list> ::= <case-item>
              | <case_list> <case_item>
```

```
<case_item> ::= <literal> "->" <stmt_list>

<default_case> ::= "default" "->" <stmt_list>

<expr> ::= <logic_or>

<logic_or> ::= <logic_and>
            | <logic_or> "||" <logic_and>

<logic_and> ::= <equality>
            | <logic_and> "&&" <equality>

<equality> ::= <relational>
            | <equality> "==" <relational>
            | <equality> "!=" <relational>

<relational> ::= <addictive>
            | <addictive> "<" <addictive>
            | <addictive> ">" <addictive>
            | <addictive> "<=" <addictive>
            | <addictive> ">=" <addictive>

<addictive> ::= <term>
            | <addictive> "+" <term>
            | <addictive> "-" <term>
            | <addictive> "~" <term>

<term> ::= <factor>
            | <term> "*" <factor>
            | <term> "/" <factor>
            | <term> "%" <factor>
```

```
<factor> ::= IDENT | <literal> | "(" <expr> ")"
           | "!" <factor> | "-" <factor>


<literal> ::= NUMBER
            | STRING
```

Syntax Design Notes

- **Terminator:** The colon (:) acts as a statement terminator

- **Case Separation:** The arrow (→) separates case literals from their execution blocks in 'choose' statements

- **Precedence:** Unary Minus > Multiplication > Addition > logic