Syntax

- Variable *shadowing* is **not** allowed. Having a variable with the same name in an inner scope is invalid, unless the variable is first declared in the inner scope.
- You must terminate a statement with a semicolon, *unless* that statement **ends** with a scope.
- Single line comments are denoted by '//' and Multi-line comments are denoted with '/* */'
- You must say **please** or **PLEASE** enough to satisfy the compiler.

Producers

$$
\begin{aligned}
[\text{Program}] &\rightarrow [\text{Statement}]^* \\
[\text{Statement}] &\rightarrow \begin{cases} \text{please} \\ \text{PLEASE} \\ \text{exit}([\text{Expr}]); \\ [\text{PrintStmt}] \\ [\text{SetStmt}] \\ [\text{ResetStmt}] \\ \text{if}([\text{Expr}])[\text{Scope}][\text{AfterIf}] \\ \text{while}([\text{Expr}])[\text{Scope}] \\ \text{for}([\text{SetStmtID}]; [\text{Expr}]; [\text{ResetStmt}])[\text{Scope}] \\ [\text{Scope}] \end{cases} \\
[\text{SetStmt}] &\rightarrow \begin{cases} [\text{SetStmtID}] \\ [\text{SetListID}] \end{cases} \\
[\text{PrintStmt}] &\rightarrow \begin{cases} \text{print}([\text{Expr}],^* [\text{Expr}]^*); \\ \text{printn}([\text{Expr}],^* [\text{Expr}]^*); \end{cases} \\
[\text{SetStmtID}] &\rightarrow \text{set } [\text{ID}] = [\text{Expr}]; \\
[\text{SetListID}] &\rightarrow \text{set } [\text{ListID}] = [\text{List}]; \\
[\text{ResetStmt}] &\rightarrow \begin{cases} \text{reset } [\text{ID}] = [\text{Expr}]; \\ \text{reset } [\text{ID}] + = [\text{Expr}]; \\ \text{reset } [\text{ID}] - = [\text{Expr}]; \\ \text{reset } [\text{ID}] * = [\text{Expr}]; \\ \text{reset } [\text{ID}] / = [\text{Expr}]; \\ \text{reset } [\text{ID}] \% = [\text{Expr}]; \end{cases} \\
[\text{Scope}] &\rightarrow \{[\text{Stmt}]^*\} \\
[\text{AfterIf}] &\rightarrow \begin{cases} \text{elsif}([\text{Expr}])[\text{Scope}][\text{AfterIf}] \\ \text{else}([\text{Expr}])[\text{Scope}] \\ \epsilon \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
[\text{Expr}] &\rightarrow \begin{cases} [\text{Term}] \\ [\text{BinaryExpr}] \end{cases} \\
[\text{Term}] &\rightarrow \begin{cases} \text{int\_lit} \\ \text{ID} \\ (\,[\text{Expr}]\,) \\ \text{len}\,(\,[\text{List}]\,) \\ \text{'char'} \end{cases} \\
[\text{ID}] &\rightarrow \begin{cases} \text{id\_lit} \\ [\text{ListLocation}] \end{cases} \\
[\text{ListLocation}] &\rightarrow \text{id\_lit}[\ [\text{Expr}]\ ] \\
[\text{ListID}] &\rightarrow \text{id\_lit} \\
[\text{List}] &\rightarrow \begin{cases} [\text{ListInit}] \\ [\text{ListNotInit}] \end{cases} \\
[\text{ListInit}] &\rightarrow \begin{cases} \{[\text{Expr}], [\text{Expr}], \cdots\} \\ \text{"char*"} \end{cases} \\
[\text{ListNotInit}] &\rightarrow \begin{cases} [\ [\text{Expr}]\ ] \\ [\ [\text{Expr}], [\text{Expr}]\ ] \\ [\ [\text{Expr}]..[\text{Expr}]\ ] \\ \text{str}([\text{Expr}]) \end{cases} \\
[\text{BinaryExpr}] &\rightarrow \begin{cases} [\text{Expr}] \times [\text{Expr}] & precedence = 2 \\ [\text{Expr}] \div [\text{Expr}] & precedence = 2 \\ [\text{Expr}]\,\%\,[\text{Expr}] & precedence = 2 \\ [\text{Expr}] + [\text{Expr}] & precedence = 1 \\ [\text{Expr}] - [\text{Expr}] & precedence = 1 \\ [Expr] \leq [Expr] & precedence = 0 \\ [Expr] \geq [Expr] & precedence = 0 \\ [Expr] > [Expr] & precedence = 0 \\ [Expr] < [Expr] & precedence = 0 \\ [Expr] == [Expr] & precedence = 0 \\ [Expr]\ != [Expr] & precedence = 0 \end{cases}
\end{aligned}
$$

Terminals

$$
\begin{aligned}
\text{int\_lit} &\rightarrow \text{[0-9]}^* \\
\text{char} &\rightarrow \text{(int)[0-9a-zA-Z]}^1 \\
\text{id\_lit} &\rightarrow \text{[a-zA-Z][0-9a-zA-Z\_]}^*
\end{aligned}
$$