

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.

Keywords

- ‘space’ → int_lit with value 32 (ASCII code for space)
- ‘newline’ → int_lit with value 10 (ASCII code for \n)
- ‘true’ → int_lit with value 1
- ‘false’ → int_lit with value 0
- 🙏 → please

Producers

$$\begin{aligned}
 [\text{Program}] &\rightarrow [\text{Statement}]^* \\
 [\text{Statement}] &\rightarrow \left\{ \begin{array}{l} \text{please} \\ \text{PLEASE} \\ \text{exit}([\text{Expr}]); \\ [\text{PrintStmt}] \\ [\text{SetStmt}] \\ [\text{ResetStmt}] \\ \text{if}([\text{Expr}])[\text{Scope}][\text{AfterIf}] \\ \text{while}([\text{Expr}])[\text{Scope}] \\ \text{for}([\text{SetStmtID}]; [\text{Expr}]; [\text{ResetStmt}])[\text{Scope}] \\ [\text{Scope}] \end{array} \right. \\
 [\text{SetStmt}] &\rightarrow \left\{ \begin{array}{l} [\text{SetStmtID}] \\ [\text{SetListID}] \end{array} \right. \\
 [\text{PrintStmt}] &\rightarrow \left\{ \begin{array}{l} \text{print}([\text{Expr}], * [\text{Expr}]^*); \\ \text{println}([\text{Expr}], * [\text{Expr}]^*); \end{array} \right. \\
 [\text{SetStmtID}] &\rightarrow \text{set } [\text{ID}] = [\text{Expr}]; \\
 [\text{SetListID}] &\rightarrow \text{set } [\text{ListID}] = [\text{List}];
 \end{aligned}$$

$$\begin{aligned}
[\text{ResetStmt}] &\rightarrow \begin{cases} \text{reset } [\text{ID}] = [\text{Expr}]; \\ \text{reset } [\text{ID}] + = [\text{Expr}]; \\ \text{reset } [\text{ID}] - = [\text{Expr}]; \\ \text{reset } [\text{ID}] * = [\text{Expr}]; \\ \text{reset } [\text{ID}] / = [\text{Expr}]; \\ \text{reset } [\text{ID}] \% = [\text{Expr}]; \end{cases} \\
[\text{Scope}] &\rightarrow \{ [\text{Stmt}]^* \} \\
[\text{AfterIf}] &\rightarrow \begin{cases} \text{elseif}([\text{Expr}])[\text{Scope}][\text{AfterIf}] \\ \text{else}([\text{Expr}])[\text{Scope}] \\ \epsilon \end{cases} \\
[\text{Expr}] &\rightarrow \begin{cases} [\text{Term}] \\ [\text{BinaryExpr}] \end{cases} \\
[\text{Term}] &\rightarrow \begin{cases} \text{int_lit} \\ \text{ID} \\ ([\text{Expr}]) \\ \text{len}([\text{List}]) \\ \text{'char'} \end{cases} \\
[\text{ID}] &\rightarrow \begin{cases} \text{id_lit} \\ [\text{ListLocation}] \end{cases} \\
[\text{ListLocation}] &\rightarrow \text{id_lit}[[\text{Expr}]] \\
[\text{ListID}] &\rightarrow \text{id_lit} \\
[\text{List}] &\rightarrow \begin{cases} [\text{ListInit}] \\ [\text{ListNotInit}] \end{cases} \\
[\text{ListInit}] &\rightarrow \begin{cases} \{ [\text{Expr}], [\text{Expr}], \dots \} \\ \text{"char*" } \end{cases} \\
[\text{ListNotInit}] &\rightarrow \begin{cases} [[\text{Expr}]] \\ [[\text{Expr}], [\text{Expr}]] \\ [[\text{Expr}]..[\text{Expr}]] \\ \text{str}([\text{Expr}]) \end{cases}
\end{aligned}$$

$$[\text{BinaryExpr}] \rightarrow \left\{ \begin{array}{ll} [\text{Expr}] \times [\text{Expr}] & \textit{precedence} = 2 \\ [\text{Expr}] \div [\text{Expr}] & \textit{precedence} = 2 \\ [\text{Expr}] \% [\text{Expr}] & \textit{precedence} = 2 \\ [\text{Expr}] + [\text{Expr}] & \textit{precedence} = 1 \\ [\text{Expr}] - [\text{Expr}] & \textit{precedence} = 1 \\ [Expr] \leq [Expr] & \textit{precedence} = 0 \\ [Expr] \geq [Expr] & \textit{precedence} = 0 \\ [Expr] < [Expr] & \textit{precedence} = 0 \\ [Expr] > [Expr] & \textit{precedence} = 0 \\ [Expr][Expr] & \textit{precedence} = 0 \\ [Expr] \neq [Expr] & \textit{precedence} = 0 \end{array} \right.$$

Terminals

$\text{int_lit} \rightarrow [0-9]^*$

$\text{char} \rightarrow (\text{int})[0-9\text{a-zA-Z}]^1$

$\text{id_lit} \rightarrow [\text{a-zA-Z}][0-9\text{a-zA-Z_}]^*$