Lexical Structure

Σ is the set of ascii characters.
Notation: a? is an abbreviation for (a | Ɛ)

<LineTerminator> ::= LF | CR | CR LF

LF is the ASCII LF character, also known as "newline", CR is the ASCII CR character, also known as "return". A CR immediately followed by an LF is considered one LineTerminator, not two.

<InputCharacter> ::= Not(CR | LF)

<Input> ::= <InputElements>? SUB?

<Input defines a well-formed input file>
SUB is th4e ASCII SUB character, also known as "control-Z"

<InputElements> ::= <InputElement> *

<InputElement> ::= <WhiteSpace> | <Comment> | <Token>

<Token> ::= <Identifier> | <Keyword> | <Literal> | <Separator> | <Operator>

<Separator> ::= . | ; | , | (|) | [|] | : | { | }

<Operator> ::= = | | | & | == | != | < | > | <= | >= | + | - | * | / | !

<Literal> ::= <IntLiteral> | <StringLiteral> | <BooleanLiteral>

<WhiteSpace> ::= SP | HT | FF | <LineTerminator>

SP is the ASCII SP character, also known as "space"
HT is the ASCII HT character, also known as "horizontal tab"
FF is the ASCII FF character, also known as "form feed"

<Comment> ::=   ## (#? Not(#))* ##

<Identifier> ::= <IdentifierStart> <IdentifierPart>*

but not <Keyword> and not <BooleanLiteral>

<IdentifierStart> ::= A .. Z | a .. z | $ | _

<IdentifierPart> ::= <IdentifierStart> | 0 .. 9

<Keyword> ::= prog | gorp | string | int | boolean | map | if | else | fi | do | od | print | println

<IntegerLiteral> ::= 0 | ((1..9) (0..9))*

<BooleanLiteral> ::= true | false

<StringLiteral> ::= "StringCharacter* "

<StringCharacter> ::= <InputCharacter> but not " or \ | <EscapeSequence>

<EscapeSequence> ::= \ b | \n | \f | \r | \" | \\

\t is a horizontal tab HT, \n is a linefeed LF, \f is a form feed FF, \r is a carriage return CR, \" is a double quote, \\ is a backslash