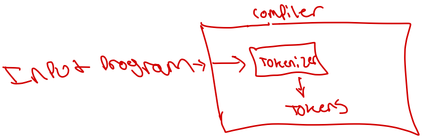
Tokenization / Lexing



```
if (34 < 57){
  return 7
}
```

Tokens:
ifToken
LeftParenToken
LessthanToken
NumberToken(57)
RightParenToken
LeftCurlyToken
ReturnToken
NumberToken(7)
SemiColonToken
RightCurlyToken

Var is a Variable

num is a Number

Type ::= 'int' | 'bool'

vardec ::= 'vardec' type var expression ')'

expression ::= num | 'true' | 'false' | var |
'(' op expression expression ')'

loop ::= '(' 'while' expression statement ')'

assign ::= '(' '=' var expression ')'

Statement ::= vardec | loop | assign

op ::= '+' | '-' | '&&' | '||' | '<'

program ::= statement

Possible tokens:

IdentifierToken(String)

IdentifierToken(int)

IntToken

BoolToken

LeftParenToken

VardecToken

RightParenToken

TrueToken

FalseToken

SingleEqualsToken

PlusToken

MinusToken

LogicalAndToken

LogicalOrToken

LessThanToken

(vardec int x 7)

LeftParenToken
VardecToken
IntToken
IdentifierToken
NumberToken(7)
Right ParenToken