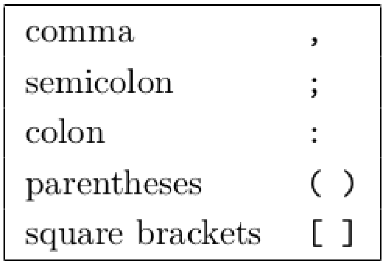
**Platform**: Linux, Unix like system

**Usage**: ./scanner [filename]

**Abilities**:

**Tokens**:

1. Delimiters

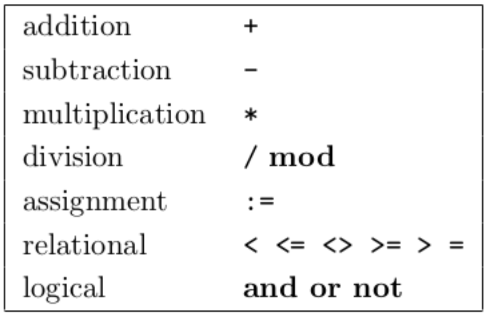


| | |
|---|---|
| comma | , |
| semicolon | ; |
| colon | : |
| parentheses | ( ) |
| square brackets | [ ] |

eg:

```
input    =>  ;
output  => <;>
```

2. Arithmetic, Relational, and Logical Operators

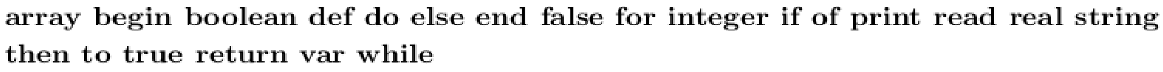


| | |
|---|---|
| addition | + |
| subtraction | - |
| multiplication | * |
| division | / **mod** |
| assignment | := |
| relational | < <= <> >= > = |
| logical | **and or not** |

eg:

```
input    =>  +
output  => <+>
```

3. Keywords (case sensitive)



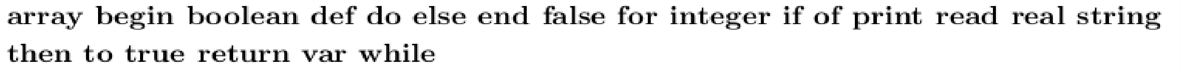
**array begin boolean def do else end false for integer if of print read real string then to true return var while**

eg:

```
input    =>  if
output  => <KWif>
```

### 4. Identifiers

A string of letters and digits beginning with a letter. Case of letters is relevant, i.e. ident, Ident, and IDENT are different identifiers.



**array begin boolean def do else end false for integer if of print read real string then to true return var while**

eg:

input => var abc

output => <Kwvar> <id: abc>

### 5. Constants

a. Integer Constants

As in the C language, an integer that begins with 0 is assumed to be octal; otherwise, it is assumed to be decimal.

b. Floating-Point Constants

A sequence of one or more digits with a dot (.) symbol separating the integral part from the fractional part.

c. Scientific Notations

All numbers are written like aEb or aeb. eg: 1.23E4, 1.23E+4, 1.23E-4, 123E4, etc.

d. String Constants

A string constant is a sequence of zero or more ASCII characters appearing between double-quote (”) delimiters. A double-quote appearing with a string must be written twice. For example, ”aa””bb” denotes the string constant aa”bb.

eg:

input => 12.23E1

output => <scientific: 12.23E1>

### 6. Discarded

Whitespace, C/C++ style comments, pseudocomments.

* Pseudocomments:

Define two options, S and T. S turns source program listing on or off, and T turns token listing on or off.

Eg:

```
//&S+
//&S-
//&S+&S-
```