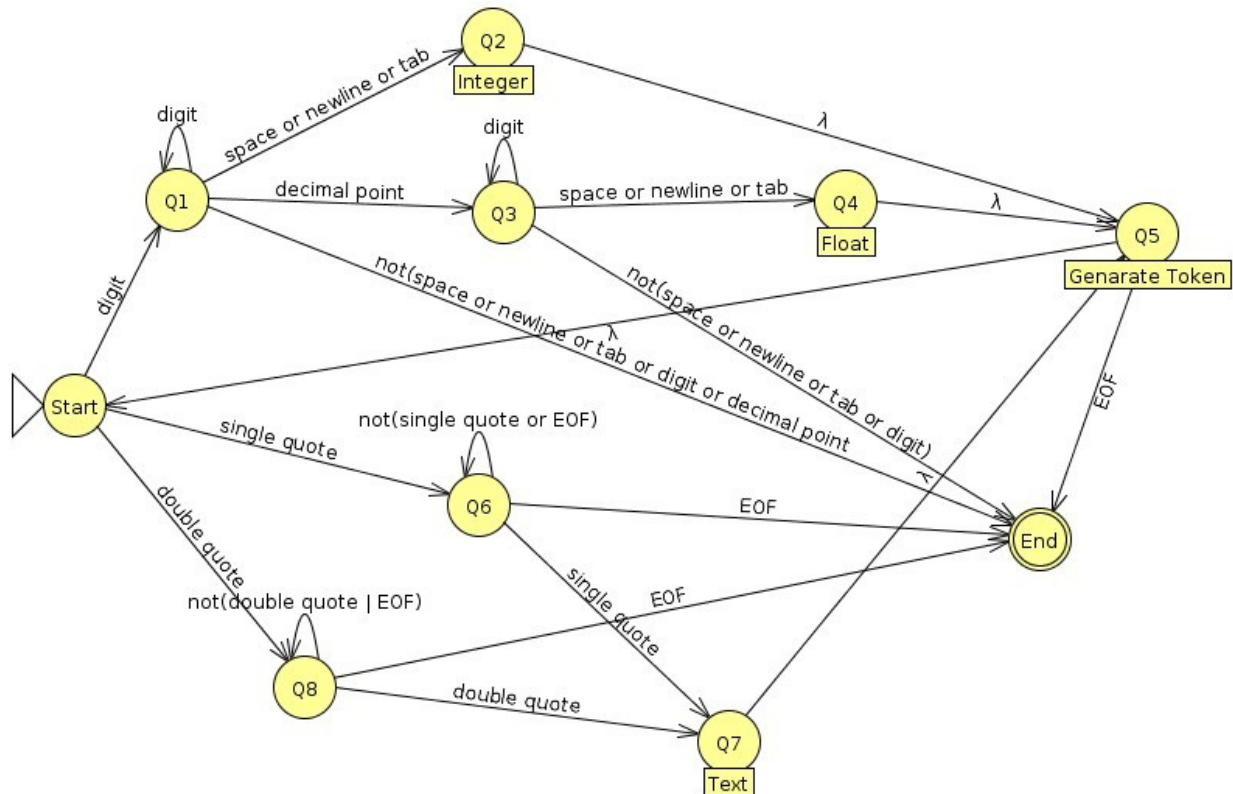# Test Document - Lexical Analysis Assignment 01

Assumptions:-
- Tokenizer tokenize the input file until it read a invalid character
- Float is only defines as a non-empty sequence of digits followed by a decimal point followed by an optional sequence of digits

A diagram of the automaton:-

This is the diagram of the automaton I have drawn. The example input file is including in the given folder as "file.txt". In this document includes screen shots of some test cases and test results I have used to test the program, "Tokenizer.py".

01. File include only integers

File :-   Result :-

```
≡ file.txt ×

1 11 111
2 22 222
3 33 333
```

```
>>>
Found a INTEGER token :- 1
Line no =  1

Found a INTEGER token :- 1 1
Line no =  1

Found a INTEGER token :- 1 1 1
Line no =  1

Found a INTEGER token :- 2
Line no =  2

Found a INTEGER token :- 2 2
Line no =  2

Found a INTEGER token :- 2 2 2
Line no =  2

Found a INTEGER token :- 3
Line no =  3

Found a INTEGER token :- 3 3
Line no =  3

Found a INTEGER token :- 3 3 3
Line no =  3

End of the file reached
>>> |
```

02. String with double quotation and some integers

```
≡ file.txt ×

1 11 111

"ishani samaraweera"

582 56 120 "lexical"
```

```
>>>
Found a INTEGER token :- 1
Line no =  1

Found a INTEGER token :- 1 1
Line no =  1

Found a INTEGER token :- 1 1 1
Line no =  1

Found a TEXT token :-
ishani samaraweera
Line number :-  3

Found a INTEGER token :- 5 8 2
Line no =  5

Found a INTEGER token :- 5 6
Line no =  5

Found a INTEGER token :- 1 2 0
Line no =  5

Found a TEXT token :-
lexical
Line number :-  5

End of the file reached
>>> |
```
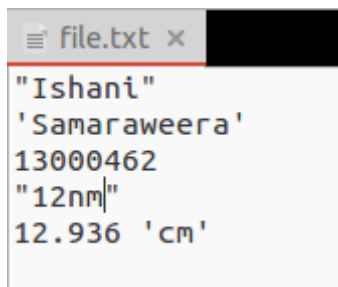
03. File contains integers, floats and strings

```
≡ file.txt ×

"Ishani"
'Samaraweera'
13000462
"12nm"
12.936 'cm'
```

```
>>>
Found a TEXT token :-
Ishani
Line number :-  1

Found a TEXT token :-
Samaraweera
Line number :-  2

Found a INTEGER token :- 1 3 0 0 0 4 6 2
Line number =  3

Found a TEXT token :-
12nm
Line number :-  4

Found a FLOAT token :- 1 2 . 9 3 6
Line number =  5

Found a TEXT token :-
cm
Line number :-  5

End of the file reached
>>> |
```
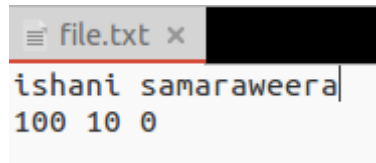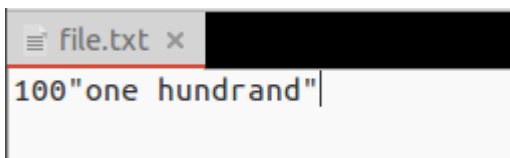
04. File has a string that not contains in double quotation or single quotation

```
file.txt ×

ishani samaraweera
100 10 0


>>>
Error occured..! Check the content of the input file in line number  1
>>> |
```

05. File contains sting and a number that not including space, tab or a new line between them

```
file.txt ×

100"one hundrand"


>>>
Error occured..! Check the content of the input file in line number  1
>>> |
```

06. File contains a invalid number that with two decimal points

```
file.txt ×

13.00.45


>>>
Error occured..! Check the content of the input file in line number  1
>>> |
```

07. Closing quotation of the string not including

```
file.txt ×

"Ishani Samaraweera


>>>
Error occured..! Check the content of the input file in line number  1
>>> |
```

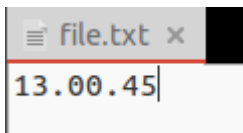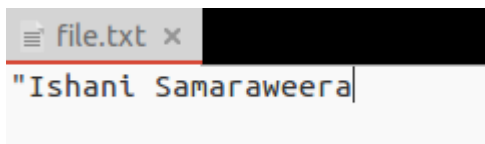08. Miss the space between two tokens

```
≡ *file.txt ×
100
55"abc|"
```

```
>>>
Found a INTEGER token :- 1 0 0
Line number =  1

Error occured..! Check the content of the input file in line number  2
>>>
```

09. A number contains that not defines as a float or integer in the scenario(In the question float is defines as a non-empty sequence of digits followed
by a decimal point followed by an optional sequence of digits.).

```
≡ file.txt ×
1000.301|
.456
```

```
>>>
Found a FLOAT token :- 1 0 0 0 . 3 0 1
Line number =  1

Error occured..! Check the content of the input file in line number  2
>>> |
```

10. File contains float values, integers, strings in single quotations and strings in double quotations

```
≡ file.txt ×
"LEXICAL ANALYSIS"

'Lexical analysis is the process of analyzing a stream
of individual characters (normally arranged as lines),
into a sequence of lexical tokens (tokenization. for
instance of "words" and punctuation symbols that make up
source code) to feed into the parser.'

4667     477        485


120.456             |45.23
12.36               456.63
5131.55


12


3.333333333333333333333333333
```

```
>>>
Found a TEXT token :-
LEXICAL ANALYSIS
Line number :-  1

Found a TEXT token :-
Lexical analysis is the process of analyzing a stream of individual characters (normally
arranged as lines), into a sequence of lexical tokens (tokenization. for instance of "wo
rds" and punctuation symbols that make up source code) to feed into the parser.
Line number :-  3

Found a INTEGER token :- 4 6 6 7
Line no =  5

Found a INTEGER token :- 4 7 7
Line no =  5

Found a INTEGER token :- 4 8 5
Line number =  5

Found a FLOAT token :- 1 2 0 . 4 5 6
Line number =  7

Found a FLOAT token :- 4 5 . 2 3
Line number =  7

Found a FLOAT token :- 1 2 . 3 6
Line number =  8

Found a FLOAT token :- 4 5 6 . 6 3
Line number =  8

Found a FLOAT token :- 5 1 3 1 . 5 5
Line number =  9

Found a INTEGER token :- 1 2
Line number =  11

Found a FLOAT token :- 3 . 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
Line number =  13

End of the file reached
>>> |
```

****
End of the document