Compiler Design Midsem Lab Evaluation

1. Write up : 20 mins [Only Tokenizer]
2. Execution : 30 mins[Tokenizer+ST]
3. Students are allowed to edit their own existing code.

Construct the lexical analyzer with a local symbol table for the following input source code. The symbol table should contain a Lexeme name, token type, and argument for functions and Lexeme name and token type for Identifiers.

```
$(document).ready(function( ) {
 $("#submit").on("click", function( )) {
 let input = $("#numbers"). val( );
 let numbers = input. split(/\s+/). map (num);
 let divisibleByThree = numbers. filter(num => !isNaN(num) && num % 3 == 0);
    $("#output").text("Numbers divisible by 3: " + (divisibleByThree.length ?
   divisibleByThree.join(",") : "None"));
  });
 });
```

1. let, function and document are keywords.
2. ready( ), on( ), val( ), split( ), map( ), filter( ), text( ), join( ),isNan() are predefined jQuery functions.
3. Submit, numbers and output are identifiers. These are surrounded by double quotes but begin with # symbol. input,s,num and divisibleByThree are also identifiers.
4. Anything enclosed in double quotes ("..") or single quotes ('..') is a literal string except for case 3.
5. $, =, + , - , * , / , % , && , ! , ?, : ,== are the operators.
6. Rest of the tokens are special symbols.
7. Token format : <TokenName, RowNumber, ColumnNumber>

Token Name	Token Type	Argument
ready	FUNC	
#submit	Identifier	
split	FUNC	(/\s+/
filter	FUNC	num => ! isNaN(num) && num % 3 == 0
On	FUNC	
Text	FUNC	
#numbers	Identifier	
map	FUNC	Number