Assignment 1
CS323
Due dates: Softcopy by 3/3 (Friday), 11:59 pm

The first assignment is to write a lexical analyzer (lexer).
You can build your *entire* lexer using a FSM, **Or build using at least FSMs for identifier, integer and real (the rest can be written ad-hoc) but *YOU HAVE TO CONSTRUCT A FSM for this assignment otherwise, there will be a deduction of 2 points!***

Note: In your documentation (design section), YOU MUST write the REs for Identifiers, Real and Integer, and also show the NFSM using Thompson.

The lexer

A major component of your assignment will be to write a procedure (Function) – lexer (), that returns a token when it is needed. Your lexer() should return a record, one field for the token and another field the actual "value" of the token (lexeme), i.e. the instance of a token.

Your main program should test the lexer i.e., your program should read a file containing the source code of Rat23S to generate tokens and write out the results to an output file. Make sure that you print both, the tokens and lexemes.

Basically, your main program should work as follows:

```
while not finished (i.e. not end of the source file) do
    call the lexer for a token
    print the token and lexeme
endwhile
```

Do at least 3 test cases and make sure that you turn in proper documentation using the documentation template.

A simple test case

Input: (Partial) **Source code**

```
while  (fahr <= upper)   a = 23.00; endwhile  [* this is sample *]
```

Output:

```
 token                    lexeme
keyword                    while
separator                  (
identifier                 fahr
operator                   <=
identifier                 upper
separator                   )
identifier                 a
operator                    =
real                       23.00
Separator                   ;
keyword                  endwhile
```