FORMAN CHRISTIAN COLLEGE (A Chartered University)
Department of Computer Science
Spring 2022

Assignment 1

Course: COMP-451: Compiler Construction **Section: B** **Due Date: April 7, 2022**

Assignment Statement:

Create a Lexical Analyzer without using any existing code or tool. You have to write the C code using Ubuntu environment. Only the following constructs should be supported by your lexical analyzer.

Keyword/Operation	**Description**
+, -, *, /	Adding, subtracting and multiplying floats
>, >=, <, <=, !=, ||, &&, =, ==	To compare two float values.
if	conditional branch
while	conditional loop
break	to exit from a conditional loop
print	prints a float value.
{, }	start or end of a if or while loop
(,)	start and end of a comparison statement

Rules:

1. Only float data type is supported.
2. Float value maybe stored in an identifier.
3. Valid identifier are letters followed by (optional) digits. (e.g. temp1, Name)
4. Newline is considered End of Line. No restriction for semi colon (;)

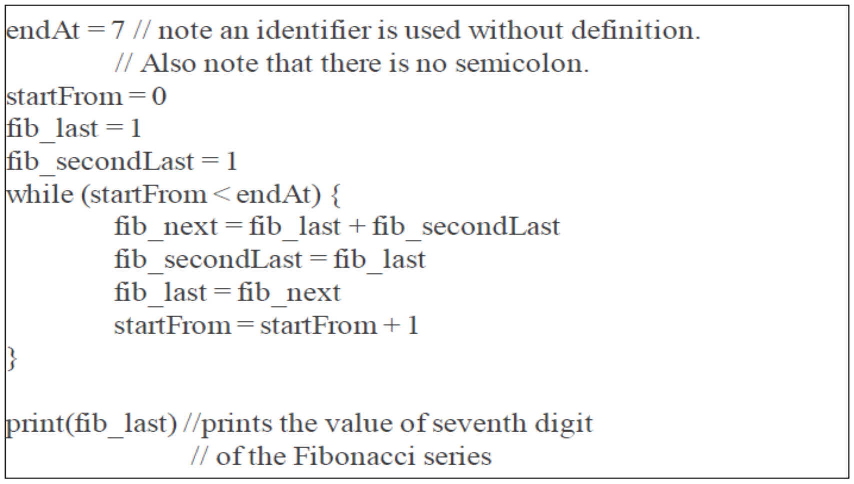
Programming Guidelines:

Your program must read the input file character by character and fill in an *input buffer*. Two pointers (as discussed in class) should be used to keep track of each lexeme. Your major function should be a *DFASimulator*() which will accept or reject lexeme as a valid token. DFA can easily be implemented using *Transition Tables*. (Refer to the textbook). Every valid token must be enlisted in the symbol table.

The generated output file should contain all the tokens from the input code.

Input:

The input should be a code file. Sample input is as follows



```
endAt = 7 // note an identifier is used without definition.
          // Also note that there is no semicolon.
startFrom = 0
fib_last = 1
fib_secondLast = 1
while (startFrom < endAt) {
          fib_next = fib_last + fib_secondLast
          fib_secondLast = fib_last
          fib_last = fib_next
          startFrom = startFrom + 1
}

print(fib_last) //prints the value of seventh digit
                 // of the Fibonacci series
```

Output:

The program should produce two files

Output – Containing all the tokens

Symbol Table – shows the structure of the table