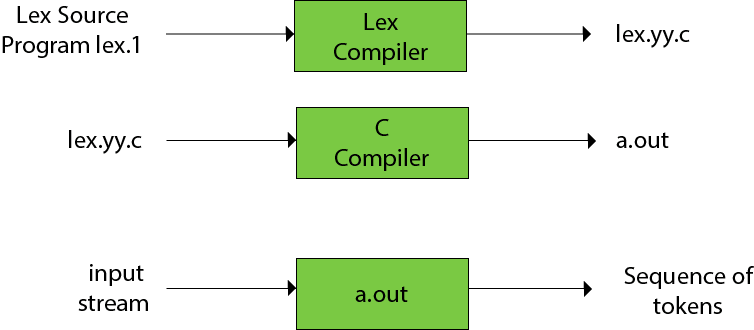
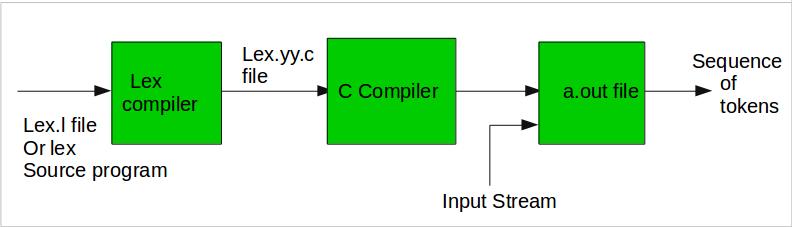
**CD Practical 5:**

**To Study about Lexical Analyzer Generator(LEX) and Flex(Fast Lexical Analyzer)**

* **LEX**
  + Lex is a program that generates lexical analyzer. It is used with YACC parser generator.
  + The lexical analyzer is a program that transforms an input stream into a sequence of tokens.
  + It reads the input stream and produces the source code as output through implementing the lexical analyzer in the C program.
  + **Function of lex:**
    - Firstly lexical analyzer creates a program lex.1 in the Lex language. Then Lex compiler runs the lex.1 program and produces a C program lex.yy.c.
    - Finally C compiler runs the lex.yy.c program and produces an object program a.out.
    - a.out is lexical analyzer that transforms an input stream into a sequence of tokens.
  + 
* **FLEX**
  + **FLEX (fast lexical analyzer generator)**is a tool/computer program for generating lexical analyzers (scanners or lexers)
  + Flex is more flexible than Lex and Yacc and produces faster code.
  + The function yylex() is automatically generated by the flex when it is provided with a .l file and this yylex() function is expected by parser to call to retrieve tokens from current/this token stream.
  + 
* **Program structure**
  + **Definition Section:**
    - The definition section contains the declaration of variables, regular definitions, manifest constants. In the definition section, text is enclosed in **“%{ %}”** brackets. Anything written in this brackets is copied directly to the file **lex.yy.c**

**%{**

**//definitions**

**%}**

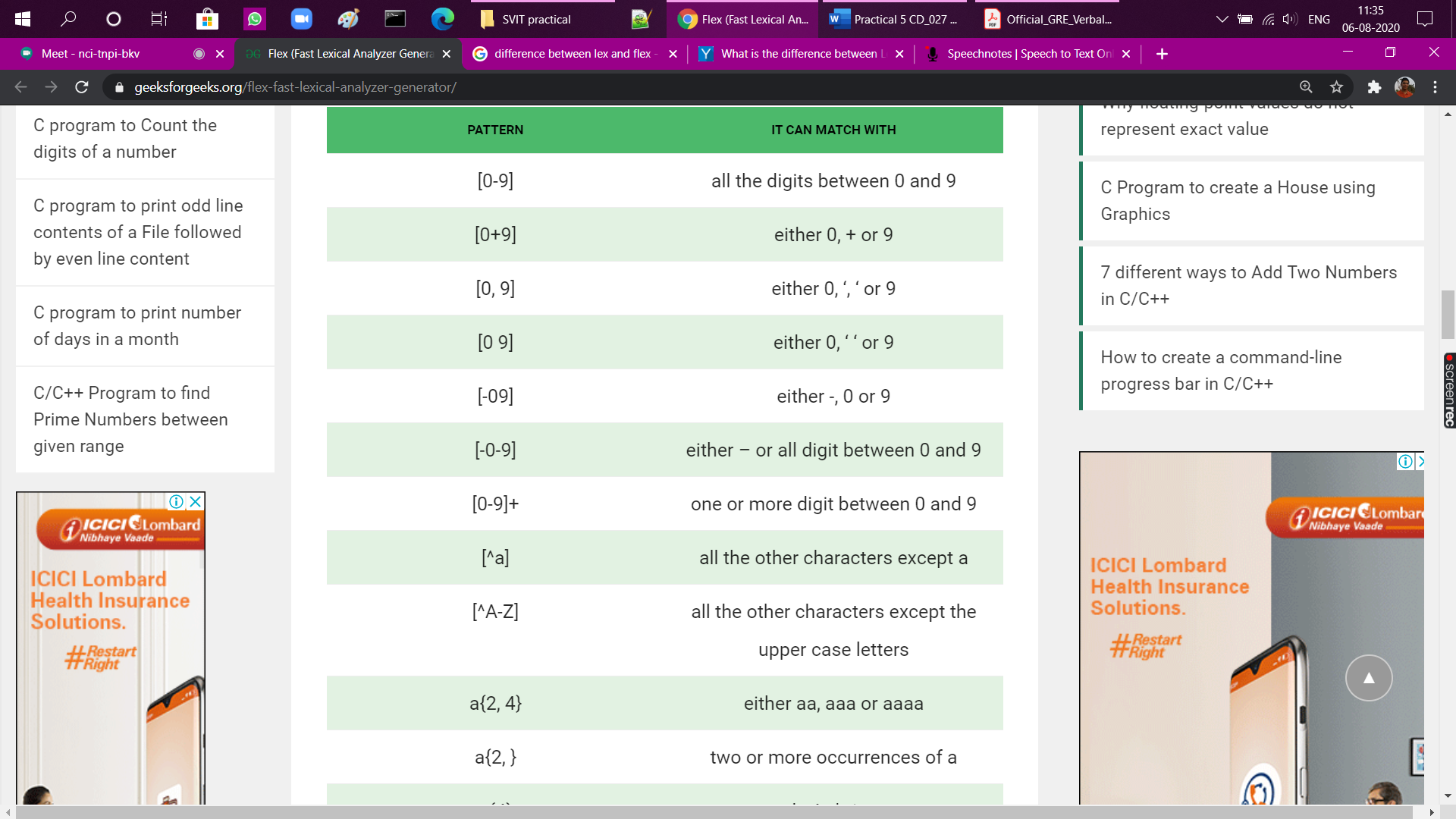
* + **Rules Section:**
    - The rules section contains a series of rules in the form: pattern action and pattern must be unintended and action begin on the same line in {} brackets. The rule section is enclosed in “%% %%”.

**%%**

**Pattern action**

**%%**

* + - Table below shows some of the pattern matches.



* + **User Code Section**
    - This section contains C statements and additional functions. We can also compile these functions separately and load with the lexical analyzer.
* **How to run the program?**
  + To run the program, it should be first saved with the extension .l or .lex. Run the below commands on terminal in order to run the program file.
    - Step 1: lex filename.l or lex filename.lex depending on the extension file is saved with
    - Step 2: gcc lex.yy.c
    - Step 3: ./a.out
    - Step 4: Provide the input to program in case it is required