LEX code using Regular Grammar (without file-handling):

1. Design a LEX Code to count the number of lines, space, tab-meta character and rest of characters in a given Input pattern.
2. Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.
3. Design a LEX Code to identify and print integer and float value in given Input pattern.
4. Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPERATORS, KEYWORDS, IDENTIFERS) the following C-fragment:

```
int p=1,d=0,r=4;

float m=0.0, n=200.0;

while (p <= 3)

   { if(d==0)

       { m= m+n*r+4.5; d++;  }

     else

       { r++; m=m+r+1000.0;  }

    p++;  }
```

LEX code using Regular Grammar (with file-handling):

5. Design a LEX Code to count and print the number of total characters, words, white spaces in given 'Input.txt' file.
6. Design a LEX Code to replace white spaces of 'Input.txt' file by a single blank character into 'Output.txt' file.
7. Design a LEX Code to remove the comments from any C-Program given at run-time and store into 'out.c' file.
8. Design a LEX Code to extract all html tags in the given HTML file at run time and store into Text file given at run time.