Question 1

Design a LEX Code to count the number of lines, space, tab-meta character, and rest of characters in each Input pattern.

Code:

```
%{
    // c code
    #include <stdio.h>
    #include <stdlib.h>

    int noLines = 0;
    int noSpace = 0;
    int noTabs = 0;
    int noCharacter = 0;
%}

%%
\n noLines++;
\t noTabs++;
([ ])+ noSpace++;
[a-zA-Z0-9] noCharacter++;
%%

int main(){
    yylex();
    printf("noLines: %d\n", noLines);
    printf("noTabs: %d\n", noTabs);
    printf("noSpaces: %d\n", noSpace);
    printf("noWords: %d\n", noCharacter);
    return 0;
}
```

Output:

Question 2

Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.

Code:

```
%{
    // c code
    #include <stdio.h>
    #include <stdlib.h>
%}

%%
^[a-zA-Z_][a-zA-Z0-9_]* printf("Valid Identifier: %s\n", yytext);
.* printf("Invalid Identifier: %s\n", yytext);
%%

int main(){
    yylex();
    return 0;
}
```

Output:

Question 3

Design a LEX Code to identify and print integer and float value in given Input pattern.

Code:

```
%{
    // c code
    #include <stdio.h>
    #include <stdlib.h>
%}

%%
[0-9]+ printf("Integer Value: %s\n", yytext);
[0-9]*"."[0-9]* printf("Float value: %s\n", yytext);
.|\n {/* Ignore all other characters. */}
%%

int main(){
    yylex();
    return 0;
}
```

Output:

Question 4

Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPARATORS, KEYWORDS, IDENTIFIERS) the following C-fragment:

Code:

```
%{
    // c code
    #include <stdio.h>
    #include <stdlib.h>

    /*
    ^{KEYWORD}[\ \n] { printf("Keyword: %s\n", yytext); }
     {KEYWORD}[ \n] { printf("Keyword: %s\n", yytext); }
    */
%}

KEYWORD int|float|if|else|while|main
SEPARATOR [,;(){}]
OPERATOR  [*+-/%=<>!~]
ID [a-zA-Z_][a-zA-Z0-9_]*

%%
{KEYWORD} { printf("Keyword: %s\n", yytext); }
{OPERATOR} { printf("Operator: %s\n", yytext); }
{SEPARATOR} { printf("Seperator: %s\n", yytext); }
{ID} { printf("Identifier: %s\n", yytext); }
.|\n {/* Ignore all other characters. */}
%%

int main(){
    yylex();
    return 0;
}
```

Output:

Question 5

Design a LEX Code to count and print the number of total characters, words, white spaces in given Input.txt file.

Code:

```
%{
    // c code
    #include <stdio.h>

    int noCharacters = 0;
    int noWords = 0;
    int noSpace = 0;
%}

%%
[ ] {noSpace++; printf("space: \"%s\"", yytext);}
[^ \n\t]+ {noWords++, noCharacters=noCharacters+yyleng; printf("words: \"%s\""
    , yytext);}
\n {noCharacters++; printf("char: \"%s\"", yytext);}
%%

int main(){
    extern FILE *yyin;
    yyin = fopen("input.txt", "r");
    FILE *fp = fopen("output.txt", "w");
    yylex();
    fprintf(fp, "noWords: %d\n", noWords);
    fprintf(fp, "noSpaces: %d\n", noSpace);
    fprintf(fp, "noCharacters: %d\n", noCharacters);
    fclose(fp);
    return 0;
}
```

Output:

Question 6

Design a LEX Code to replace white spaces of Input.txt file by a single blank character into Output.txt file.

Code:

Output:

Question 7

Design a LEX Code to remove the comments from any C-Program given at run-time and store into out.c file.

Code:

Output:

Question 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.

Code:

Output: