29. Write a LEX program to print all HTML tags in the input file.

%{

#include <stdio.h>

%}

%%

<[^>]+> { printf("HTML tag: %s\n", yytext); } // Match HTML tags (e.g., <html>, <body>, etc.)

%%

int main() {

FILE \*file = fopen("input.html", "r"); // Open the HTML file

if (file) {

yyin = file; // Set the input file for Lex

yylex(); // Start lexical analysis

fclose(file); // Close the file after processing

} else {

printf("Unable to open the file.\n");

return 1;

}

return 0;

}  
30. Write a LEX program which adds line numbers to the given C program file and display the same in the standard output.

%{

#include <stdio.h>

int line\_number = 1; // Variable to keep track of line numbers

%}

%%

\n { printf("%d: %s", line\_number++, yytext); } // Print line number with each line

.|\t { printf("%s", yytext); } // Print other characters (except newline) as-is

%%

int main() {

FILE \*file = fopen("input.c", "r"); // Open the C source file

if (file) {

yyin = file; // Set the input file for Lex

yylex(); // Start lexical analysis

fclose(file); // Close the file after processing

} else {

printf("Unable to open the file.\n");

return 1;

}

return 0;

}  
31. Write a LEX program to count the number of comment lines in a given C program and eliminate them and write into another file.

%{

#include <stdio.h>

int comment\_count = 0; // Counter for comment lines

FILE \*output\_file; // Output file to write the content without comments

%}

%%

/\\*[^\*]\*\\*+([^/\*][^\*]\*\\*+)\*/ { comment\_count++; } // Match block comments (/\* ... \*/)

\/\/[^\n]\* { comment\_count++; } // Match line comments (// ...)

[^/\n]+ { fprintf(output\_file, "%s", yytext); } // Write non-comment content to output file

%%

int main() {

FILE \*input\_file = fopen("input.c", "r"); // Open the C source file

if (!input\_file) {

printf("Unable to open the input file.\n");

return 1;

}

output\_file = fopen("output.c", "w"); // Open the output file to write non-comment content

if (!output\_file) {

printf("Unable to open the output file.\n");

return 1;

}

yyin = input\_file; // Set the input file for Lex

yylex(); // Start lexical analysis

fclose(input\_file); // Close the input file after processing

fclose(output\_file); // Close the output file after writing

printf("Number of comment lines: %d\n", comment\_count); // Print the comment count

return 0;

}