35. Write a LEX program to recognise numbers and words in a statement.

%{

#include <stdio.h>

%}

%%

[0-9]+ { printf("Number: %s\n", yytext); } // Recognize numbers

[A-Za-z]+ { printf("Word: %s\n", yytext); } // Recognize words

%%

int main() {

yylex(); // Start lexical analysis

return 0;

}  
36. Write a LEX program to identify and count positive and negative numbers.

%{

#include <stdio.h>

int positive\_count = 0;

int negative\_count = 0;

%}

%%

[+]?[0-9]+(\.[0-9]+)? { // Matches positive numbers, including decimal numbers

if (yytext[0] == '-') {

negative\_count++;

} else {

positive\_count++;

}

printf("Number: %s\n", yytext);

}

%%

int main() {

yylex(); // Start lexical analysis

printf("Total positive numbers: %d\n", positive\_count);

printf("Total negative numbers: %d\n", negative\_count);

return 0;

}  
37. Write a LEX program to validate the URL.

%{

#include <stdio.h>

int is\_valid\_url(char\* str) {

// Simple check for URL pattern (basic validation)

if (strstr(str, "://") != NULL) {

return 1; // Valid URL

}

return 0; // Invalid URL

}

%}

%%

https?://[A-Za-z0-9.-]+\.[A-Za-z]{2,6} {

if (is\_valid\_url(yytext)) {

printf("Valid URL: %s\n", yytext);

} else {

printf("Invalid URL: %s\n", yytext);

}

}

%%

int main() {

yylex(); // Start lexical analysis

return 0;

}