**AIM:** Write a lex program to count positive and negative numbers from the input file. (Note: It is compulsory to read the input from the file and display the results in another file)

**CODE:**

**Input.txt**

6 -10 9.6 -17.1 1-e -2-e

**Practical\_3.l**

%{

#include <stdio.h>

#include <stdlib.h>

int pos\_int\_count = 0;

int neg\_int\_count = 0;

int pos\_float\_count = 0;

int neg\_float\_count = 0;

int pos\_exp\_int\_count = 0;

int neg\_exp\_int\_count = 0;

int pos\_exp\_float\_count = 0;

int neg\_exp\_float\_count = 0;

%}

%%

[+]?[0-9]+[eE][-]?[0-9]\* { pos\_exp\_int\_count++; }

[+]?[0-9]+"-e" { pos\_exp\_int\_count++; }

-[0-9]+[eE][-]?[0-9]\* { neg\_exp\_int\_count++; }

-[0-9]+"-e" { neg\_exp\_int\_count++; }

[+]?[0-9]\*\.[0-9]+[eE][-]?[0-9]\* { pos\_exp\_float\_count++; }

[+]?[0-9]\*\.[0-9]+"-e" { pos\_exp\_float\_count++; }

-[0-9]\*\.[0-9]+[eE][-]?[0-9]\* { neg\_exp\_float\_count++; }

-[0-9]\*\.[0-9]+"-e" { neg\_exp\_float\_count++; }

[+]?[0-9]+ { pos\_int\_count++; }

-[0-9]+ { neg\_int\_count++; }

[+]?[0-9]\*\.[0-9]+ { pos\_float\_count++; }

-[0-9]\*\.[0-9]+ { neg\_float\_count++; }

[ \t\n] ; // Ignore whitespace

. ; // Ignore any other character

%%

int main(int argc, char \*\*argv) {

if (argc != 3) {

fprintf(stderr, "Usage: %s <input file> <output file>\n", argv[0]);

return 1;

}

FILE \*infile = fopen(argv[1], "r");

FILE \*outfile = fopen(argv[2], "w");

if (!infile || !outfile) {

perror("File error");

return 1;

}

yyin = infile;

yyout = outfile;

yylex();

fprintf(outfile, "Positive integers: %d\n", pos\_int\_count);

fprintf(outfile, "Negative integers: %d\n", neg\_int\_count);

fprintf(outfile, "Positive floats: %d\n", pos\_float\_count);

fprintf(outfile, "Negative floats: %d\n", neg\_float\_count);

fprintf(outfile, "Positive exponential integers: %d\n", pos\_exp\_int\_count);

fprintf(outfile, "Negative exponential integers: %d\n", neg\_exp\_int\_count);

fprintf(outfile, "Positive exponential floats: %d\n", pos\_exp\_float\_count);

fprintf(outfile, "Negative exponential floats: %d\n", neg\_exp\_float\_count);

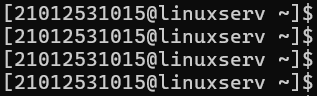
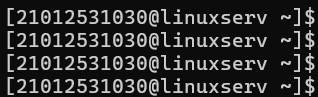
fclose(infile);

fclose(outfile);

return 0;

}

**OUTPUT:**

****

