**CSA1443- Compiler Design for Intraprocedural Analysis**

**192321047 – KAUSHIK NARAYANAN.V**

**17. Write a C program for implementing a Lexical Analyzer to Scan and Count the number of characters, words, and lines in a file.**

**Aim:**To implement a lexical analyzer in C that scans and counts characters, words, and lines from a given file.

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int main() {

FILE \*file;

char filename[50], ch;

int characters = 0, words = 0, lines = 0;

printf("Enter filename: ");

scanf("%s", filename);

file = fopen(filename, "r");

if (file == NULL) {

printf("File not found!\n");

return 1;

}

while ((ch = fgetc(file)) != EOF) {

characters++;

if (ch == ' ' || ch == '\t' || ch == '\n') words++;

if (ch == '\n') lines++;

}

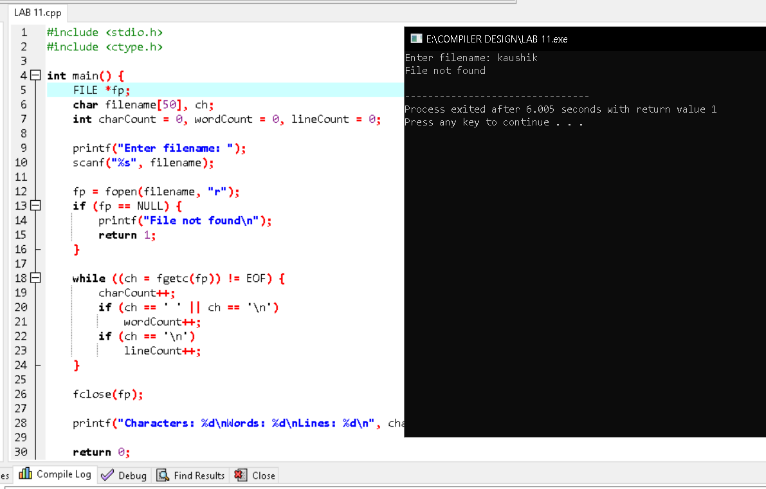
fclose(file);

printf("Characters: %d\nWords: %d\nLines: %d\n", characters, words, lines);

return 0;

}

**Output:**

****

**18. Write a C program to implement the back end of the compiler.**

**Aim:**To implement the back end of a compiler in C that generates machine code for arithmetic expressions.

**Code:**

#include <stdio.h>

int main() {

char expr[50];

printf("Enter arithmetic expression: ");

scanf("%s", expr);

printf("\nGenerated Assembly Code:\n");

for (int i = 0; expr[i] != '\0'; i++) {

if (expr[i] == '+') printf("ADD\n");

else if (expr[i] == '-') printf("SUB\n");

else if (expr[i] == '\*') printf("MUL\n");

else if (expr[i] == '/') printf("DIV\n");

else printf("PUSH %c\n", expr[i]);

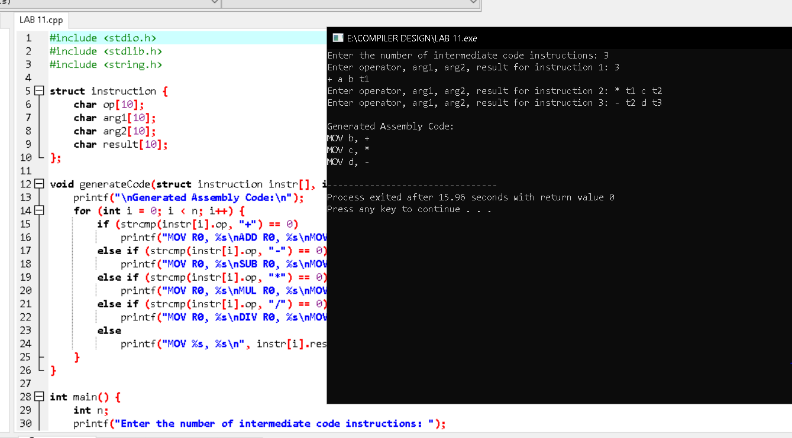
}

printf("POP RESULT\n");

return 0;

}

**Output:**

****