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

**Code:**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define MAX 10

char productions[MAX][MAX];

char leading[MAX][MAX];

int numProductions;

void findLeading(char nonTerminal, int index) {

for (int i = 0; i < numProductions; i++) {

if (productions[i][0] == nonTerminal) {

for (int j = 2; productions[i][j] != '\0'; j++) {

if (!isupper(productions[i][j])) { // If terminal, add to LEADING

int len = strlen(leading[index]);

if (!strchr(leading[index], productions[i][j])) {

leading[index][len] = productions[i][j];

leading[index][len + 1] = '\0';

}

break; // Stop after the first terminal

} else { // If non-terminal, recursively find LEADING

findLeading(productions[i][j], index);

}

}

}

}

}

int main() {

printf("Enter the number of productions: ");

scanf("%d", &numProductions);

getchar(); // Clear newline buffer

printf("Enter the productions (E -> aE format):\n");

for (int i = 0; i < numProductions; i++) {

fgets(productions[i], MAX, stdin);

productions[i][strcspn(productions[i], "\n")] = 0; // Remove newline

}

// Initialize leading array

for (int i = 0; i < MAX; i++) {

leading[i][0] = '\0';

}

printf("\nComputing LEADING():\n");

for (int i = 0; i < numProductions; i++) {

char nonTerminal = productions[i][0];

if (leading[nonTerminal - 'A'][0] == '\0') {

leading[nonTerminal - 'A'][0] = nonTerminal;

leading[nonTerminal - 'A'][1] = ':';

leading[nonTerminal - 'A'][2] = '\0';

findLeading(nonTerminal, nonTerminal - 'A');

}

}

for (int i = 0; i < MAX; i++) {

if (leading[i][0] != '\0') {

printf("LEADING(%c) = { %s }\n", leading[i][0], leading[i] + 2);

}

}

return 0;

}

**OUTPUT:**

