**Write a program to count the number of characters in a given line of string using flex.**

%{

#include <stdio.h>

%}

%%

.|\n { printf("%c", yytext[0]); } // Print each character

%%

int main() {

printf("Enter a line of string:\n");

yylex(); // This function starts the scanning process defined in the Flex rules

printf("\n");

return 0;

}

**Write a program to implement identification of keywords, numbers, operators and identifiers using flex.**

%{

#include <stdio.h>

%}

/\* Definitions \*/

%option noyywrap

/\* Regular Definitions \*/

DIGIT [0-9]

LETTER [a-zA-Z]

%%

/\* Rules \*/

"int"|"float"|"char"|"double"|"void" { printf("Keyword: %s\n", yytext); }

{DIGIT}+(\.{DIGIT}+)? { printf("Number: %s\n", yytext); }

"="|"+"|"-"|"\*"|"/" { printf("Operator: %s\n", yytext); }

{LETTER}({LETTER}|{DIGIT})\* { printf("Identifier: %s\n", yytext); }

/\* Ignore whitespace and newline characters \*/

[ \t\n] ;

. { printf("Invalid: %s\n", yytext); }

%%

int main() {

printf("Enter an expression:\n");

yylex();

return 0;

}

**Write a program to print all the keywords, literals, valid identifiers, invalid identifiers, integer number, real number in a given C program**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

int isKeyword(char \*str) {

char keywords[32][10] = {"auto", "break", "case", "char", "const", "continue", "default",

"do", "double", "else", "enum", "extern", "float", "for", "goto",

"if", "int", "long", "register", "return", "short", "signed", "sizeof",

"static", "struct", "switch", "typedef", "union", "unsigned", "void",

"volatile", "while"};

int i, flag = 0;

for(i = 0; i < 32; ++i) {

if(strcmp(keywords[i], str) == 0) {

flag = 1;

break;

}

}

return flag;

}

int isValidIdentifier(char \*str) {

if (!isalpha(str[0]) && str[0] != '\_')

return 0;

for (int i = 1; i < strlen(str); i++) {

if (!isalnum(str[i]) && str[i] != '\_')

return 0;

}

return 1;

}

int main() {

FILE \*file;

char filename[100], c, buffer[1000], word[100];

int i = 0, j = 0;

printf("Enter the filename to scan: ");

scanf("%s", filename);

file = fopen(filename, "r");

if (file == NULL) {

printf("File %s not found!\n", filename);

exit(0);

}

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

if (isalnum(c) || c == '.' || c == '\'' || c == '\"') {

buffer[i++] = c;

} else {

buffer[i] = '\0';

i = 0;

if (isKeyword(buffer)) {

printf("Keyword: %s\n", buffer);

} else if (isdigit(buffer[0]) || buffer[0] == '.' || buffer[0] == '\'' || buffer[0] == '\"') {

int isReal = 0;

for (int k = 0; k < strlen(buffer); k++) {

if (buffer[k] == '.') {

isReal = 1;

break;

}

}

if (isReal) {

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

} else {

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

}

} else if (buffer[0] == '\'' || buffer[0] == '\"') {

printf("Literal: %s\n", buffer);

} else {

if (isValidIdentifier(buffer)) {

printf("Valid Identifier: %s\n", buffer);

} else {

printf("Invalid Identifier: %s\n", buffer);

}

}

}

}

fclose(file);

return 0;

}

**Write a Lex program to recognize valid arithmetic expressions and identify the identifiers and operators**

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

%}

/\* Regular Definitions \*/

digit [0-9]

letter [a-zA-Z]

%%

{digit}+ printf("Number: %s\n", yytext);

{letter}({letter}|{digit})\* printf("Identifier: %s\n", yytext);

"="|"+|"-"|"\*"|"/"|"%" printf("Operator: %s\n", yytext);

"(" printf("Left Parenthesis\n");

")" printf("Right Parenthesis\n");

"\n" /\* Ignore newline \*/;

[ \t]+ /\* Ignore whitespace \*/;

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

%%

int main() {

printf("Enter an arithmetic expression:\n");

yylex();

return 0;

}