%{

#include <stdio.h>

#include <string.h>

#define MAX\_LEXEMES 100

typedef struct {

char lexeme[50];

char category[20];

} Lexeme;

Lexeme lexemeTable[MAX\_LEXEMES];

int lexemeCount = 0;

void addToTable(const char \*category, const char \*lexeme) {

if (lexemeCount < MAX\_LEXEMES) {

strcpy(lexemeTable[lexemeCount].lexeme, lexeme);

strcpy(lexemeTable[lexemeCount].category, category);

lexemeCount++;

}

}

void printTable() {

printf("\n-----------------------------------\n");

printf("| %-20s | %-15s |\n", "Lexeme", "Category");

printf("-----------------------------------\n");

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

printf("| %-20s | %-15s |\n", lexemeTable[i].lexeme, lexemeTable[i].category);

}

printf("-----------------------------------\n");

}

%}

%%

"int"|"float"|"if"|"else"|"while"|"for"|"return"|"void"|"char" { addToTable("Keyword", yytext); }

[a-zA-Z\_][a-zA-Z0-9\_]\* {

addToTable("Identifier", yytext); }

"=="|"<="|">="|"<"|">"|"!="|"="|"+"|"-"|"\*"|"/"|"%" {

addToTable("Operator", yytext); }

[,;.:?\*()] { addToTable("Punctuation", yytext); }

[ \t\n]+ ;

. { addToTable("Unrecognized", yytext); }

%%

int main() {

printf("Enter the code to analyze:\n");

fflush(stdout);

yylex();

printTable();

return 0;

}

**Commands to compile**

lex filename.l

gcc lex.yy.c -lfl

./a.out

