Lexical Analyzer Implementation in C

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

// List of keywords
const char *keywords[] = {"int", "float", "char", "if", "else", "while", "for",
"return"};
const int num_keywords = 8;

// Function to check if a string is a keyword
int isKeyword(const char *word) {
    for (int i = 0; i < num_keywords; i++) {
        if (strcmp(word, keywords[i]) == 0)
            return 1;
    }
    return 0;
}

// Function to check if a character is an operator
int isOperator(char ch) {
    char operators[] = "+-*/=<>";
    for (int i = 0; i < strlen(operators); i++) {
        if (ch == operators[i])
```

```
            return 1;
    }
    return 0;
}

// Function to perform lexical analysis
void lexicalAnalyzer(const char *filename) {
    FILE *file = fopen(filename, "r");
    if (!file) {
        perror("Error opening file");
        exit(1);
    }

    char ch, buffer[100];
    int i = 0;

    printf("Lexical Analysis:\n");
    while ((ch = fgetc(file)) != EOF) {
        if (isalnum(ch)) {
            buffer[i++] = ch;
        } else {
            if (i > 0) {
                buffer[i] = '\0';
                if (isKeyword(buffer)) {
                    printf("Keyword: %s\n", buffer);
                } else {
                    printf("Identifier: %s\n", buffer);
```

```c
                }
                i = 0;
            }
            if (isOperator(ch)) {
                printf("Operator: %c\n", ch);
            }
        }
    }
    fclose(file);
}

// Main function
int main() {
    const char *filename = "input.txt";

    printf("Reading from file: %s\n", filename);

    // Perform lexical analysis on the input file
    lexicalAnalyzer(filename);

    return 0;
}
```