Lab experiment 3rd question

Design a lexical Analyzer to validate operators to recognize the operators +,-,\*,/ using regular Arithmetic operators .

**Aim:**

To design a **Lexical Analyzer** that validates and recognizes arithmetic operators (+, -, \*, /) using regular expressions in C. The program will analyze an input source file and identify valid operators while ignoring spaces, tabs, new lines, and comments.

**Procedure:**

1. **Read the Input File**
   * Open and read a source code file.
2. **Ignore Whitespaces and Comments**
   * Skip extra spaces, tabs, and new lines.
   * Remove single-line (//) and multi-line (/\* ... \*/) comments.
3. **Identify and Validate Operators**
   * Check each character and determine if it is one of the arithmetic operators (+, -, \*, /).
   * If valid, store and display the recognized operator.
4. **Display Tokens**
   * Print out identified operators along with their token type.

**Implementation (C Program)**

This program reads a source code file (input.c), ignores comments, and identifies arithmetic operators.

c

CopyEdit

#include <stdio.h>

#include <stdlib.h>

int isArithmeticOperator(char ch) {

return (ch == '+' || ch == '-' || ch == '\*' || ch == '/');

}void lexicalAnalyzer(FILE \*fp) {

char ch;

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

if (ch == ' ' || ch == '\t' || ch == '\n') continue;

if (ch == '/' && (ch = fgetc(fp)) == '/') {

while ((ch = fgetc(fp)) != '\n' && ch != EOF);

continue;

}

if (ch == '/' && (ch = fgetc(fp)) == '\*') {

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

if (ch == '\*' && (ch = fgetc(fp)) == '/') b

reak;

}

continue;

}

if (isArithmeticOperator(ch)) {

printf("Arithmetic Operator: %c\n", ch);

}

}

}

int main() {

FILE \*fp = fopen("input.c", "r");

if (!fp) {

printf("Error opening file!\n");

return 1;

}

printf("Lexical Analysis Output:\n");

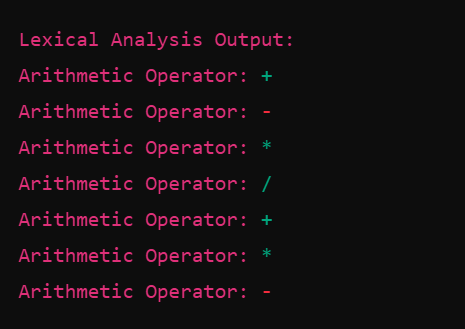
lexicalAnalyzer(fp);

fclose(fp);

return 0;

}

**Output:**

****