Experiment 1

**Aim**: Design and implement a lexical analyzer for given language using C. The lexical analyzer should ignore redundant spaces, tabs and newlines.

Algorithm

0. Assumption:- The delimiter for tokenization is one among [ <space>, <comma>, <semicolon> ]. ie., operators cannot be delimiters.

Also, macro expansions are not performed.

1. Start
2. Provide the filename as a command line argument.
3. Open the file `file` in read mode.
4. If the file does not exist or the user does not have the permission to open it, exit the program. Else goto next step.
5. Read a character from the file into `ch`.
6. While `ch` is not `EOF` do
7. If `ch` is '/' and the next character is also '/' then skip the line: single line comment.
8. If `ch` is '/' and next character is '\*' keep skipping characters till '\*' and '/' occurring together is encountered: multiline comments.
9. If `ch` is '#' skip the line as a preprocessor directive.
10. If `ch` is alphanumeric append `ch` into `token`.
11. Else do
    1. If `token` is a keyword print `kwd`.
    2. Else if `token` is an identifier print `id`.
    3. Else if `token` is a number print the number.
    4. Else if `ch` is an operator, print operator token.
    5. Else if `ch` is ',' or ';' print `ch`.
12. Read the next character into `ch`.
13. Goto step 6, end while.
14. Close the file.
15. Stop

Input file

program.c

#include <stdio.h>

#include <stdlib.h>

#define PI 3.1415926

// Single line comment

int a,b,c;

/\* Multiline

\* comment \*/

float PI; // PI is a constant

c = a + b;

int var = b / c % a;

return var / PI;

/\* This is a demo program to

\* test the 'lexical analyser' created using C

\*/

Output

$ ./a.out program.c

kwd id, id, id; kwd id; id oper-equ id oper-plus id; kwd id oper-equ id oper-div id oper-mod id; kwd id oper-div id;

Screenshot of Program output



Readme

1. Compile the program using the command

**gcc 37Jishnu\_P1.c**

2. Execute the program using the command

**./a.out program.c**

3. The stream of tokens obtained is displayed in the terminal.

**Result**:

Successfully created a lexical analyzer for the given language using C.