S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 1

Aim: Implementation of LEX (Compiler Writing Tool) using Linux LEX Utility: -

(a) To recognize whether the given input string is a keyword, identifier, number (integer or real) and operators.

(b) To count the number of vowels and consonants in a given input.

Name of Student: Shrutika Pradeep Bagdi

Roll No.: CS22130

Semester/Year: 6th Semester/3rd Year

Academic Session: 2024-25

Date of Performance:

Date of Submission:

Aim: Implementation of LEX (Compiler Writing Tool) using Linux LEX Utility: -

(a) To recognize whether the given input string is a keyword, identifier, number (integer or real) and operators.

(b) To count the number of vowels and consonants in a given input.

OBJECTIVE / EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- To illustrate the use of Compiler writing Tool (LEX) in implementing Lexical Analyzer
- To understand the format and demonstrate the LEX program.

HARDWARE AND SOFTWARE REQUIRMENTS:

Hardware Requirement:

- Processor: Dual Core
- RAM: 1GB
- Hard Disk Drive: > 80 GB

Software Requirement:

- LEX utility on Linux

THEORY:

1) Role of Lexical Analyzer in Compiler

2) Implementation of Lexical Analyzer using LEX tool

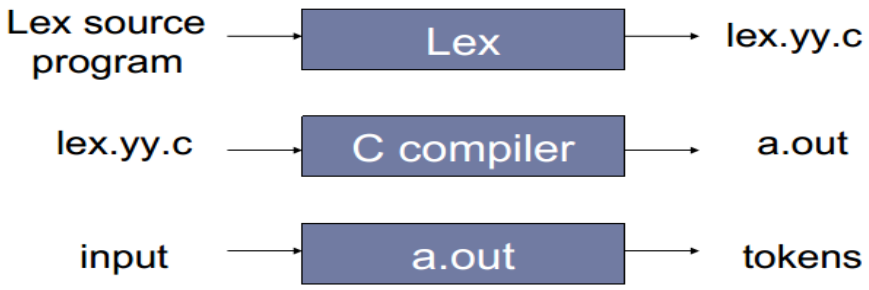
3) Introduction of LEX tool

4) LEX functions and its purpose

5) LEX Variables and its use

Procedure to Execute LEX Program:

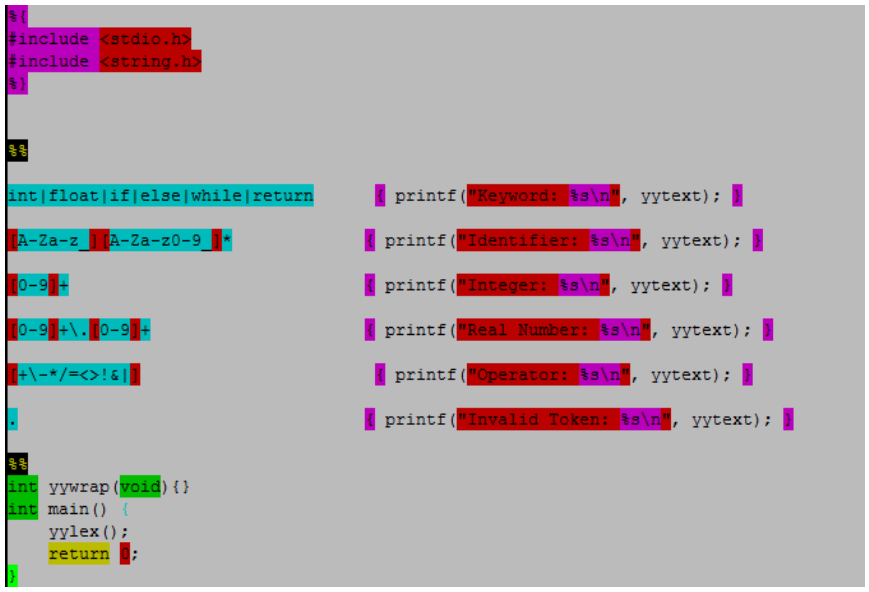
Diagrammatic Representation of LEX:



AIM: (a) To recognize whether the given input string is a keyword, identifier, number (integer or real) and operators.

CODE:

```
%{
#include <stdio.h>
#include <string.h>
%}

%%

int|float|if|else|while|return      { printf("Keyword: %s\n", yytext); }

[A-Za-z_][A-Za-z0-9_]*            { printf("Identifier: %s\n", yytext); }

[0-9]+                          { printf("Integer: %s\n", yytext); }

[0-9]+\.[0-9]+                    { printf("Real Number: %s\n", yytext); }

[+\-*/=<>!&|]                     { printf("Operator: %s\n", yytext); }

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

%%
int yywrap(void){}
int main() {
   yylex();
   return 0;
}
```



OUTPUT:

```
csc15@linux-p2-1272il:~/CS22130$ ./a.out
int
Keyword: int

S
Identifier: S

5
Integer: 5

5.6
Real Number: 5.6

+
Operator: +
```

AIM: (b) To count the number of vowels and consonants in a given input.

CODE:

```
%{
int vow_count = 0;
int const_count = 0;
%}
%%
[aeiouAEIOU] { vow_count++; }
[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z] { const_count++; }
.|\n {}
%%
int yywrap() {
    return 1;  // End of input
}
int main() {
    printf("Enter the string of vowels and consonants: ");
    yylex();  // Start Lexical analysis
    printf("Number of vowels: %d\n", vow_count);
    printf("Number of consonants: %d\n", const_count);
    return 0;
}
```

```
csc15@linux-p2-1272il:~/CS22130$ vi Practical1B.lex
%{
#include<stdio.h>
int vow_count = 0;
int const_count = 0;
%}

%%

[aeiouAEIOU] { vow_count++; }
[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z] { const_count++; }
.|\n {}

%%

int yywrap() {
    return 1;
}

int main() {
    printf("Enter the string of vowels and consonants: ");
    yylex();
    printf("Number of vowels: %d\n", vow_count);
    printf("Number of consonants: %d\n", const_count);
    return 0;
}
```

OUTPUT:

```
Enter the string of vowels and consonants: Shrutika
Number of vowels: 3
Number of consonants: 5
csc15@linux-p2-1272il:~/CS22130$
```

CONCLUSION:

DISCUSSION AND VIVA VOCE:

1. What is the difference between token and lexeme?
2. What is lexical analyzer?
3. Which tool is used for lexical analyzer?
4. What is the output of Lexical analyzer?
5. What is LEX source Program?

REFERENCE:

- Lab Manual of Compiler Design (Institute of Aeronautical Engineering, Dundigal, Hyderabad)
- https://en.wikipedia.org/wiki/Lex_(software)