NIKIT GOKHE
Class – SY Comp D1
Roll No. 224024
GR No. 21810522

# ASSIGNMENT NO 3

**AIM:**

Write a program using LEX specifications to implement lexical analysis phase of compiler to count no. of words, lines and characters of given input file.

Lex is officially known as a ”Lexical Analyser”.It’s main job is to break up an input stream into more usable elements.Or in, other words, to identify the ”interesting bits” in a text file.For exam- ple, if you are writing a compiler for the C programming language, the symbols ( ) ; all have significance on their own. The letter a usually appears as part of a keyword or variable name, and is not interesting on it’s own. Instead, we are interested in the whole word. Spaces and newlines are completely uninteresting, and we want to ignore them completely, unless they appear within quotes ”like this”All of these things are handled by the Lexical Analyser.

Lex helps write programs whose control flow is directed by instances of regular expressions in the input stream. It is well suited for editor-script type transformations and for segmenting input in preparation for a parsing routine.

Lex source is a table of regular expressions and corresponding program fragments. The table is translated to a program which reads an input stream, copying it to an output stream and parti- tioning the input into strings which match the given expressions. As each such string is recognized the corresponding program fragment is executed. The recognition of the expressions is performed by a deterministic finite automaton generated by Lex. The program fragments written by the user are executed in the order in which the corresponding regular expressions occur in the input stream.

The lexical analysis programs written with Lex accept ambiguous specifications and choose the longest match possible at each input point. If necessary, substantial lookahead is performed on the input, but the input stream will be backed up to the end of the current partition, so that the user has general freedom to manipulate it.

Lex can generate analyzers in either C or Ratfor, a language which can be translated auto- matically to portable Fortran. It is available on the PDP-11 UNIX, Honeywell GCOS, and IBM OS systems. This manual, however, will only discuss generating analyzers in C on the UNIX sys- tem, which is the only supported form of Lex under UNIX Version 7. Lex is designed to simplify interfacing with Yacc, for those with access to this compiler-compiler system.

**Running Lex:**

*>lex filename.l*
*>gcc lex.yy.c -lfl*
*>./a.out*

To compile a lex program, do the following: Use the lex program to change the specification file into a C language program. The resulting program is in the lex.yy.c file. Use the cc command with the -ll flag to compile and link the program with a library of lex subroutines. The resulting executable program is in the a.out file.

## Algorithm:

Step1: Start
Step2:Read input file name
Step3 :Initialize word count,letter count,line count and character count to zero Step4 :If input file matches

[ \n] + then increment the line count,

[ a–zA–Z0 –9] then increment character count

[\t \n] + then increment word count

Step5 :Print the count of word, line, character count
Step6 :Stop

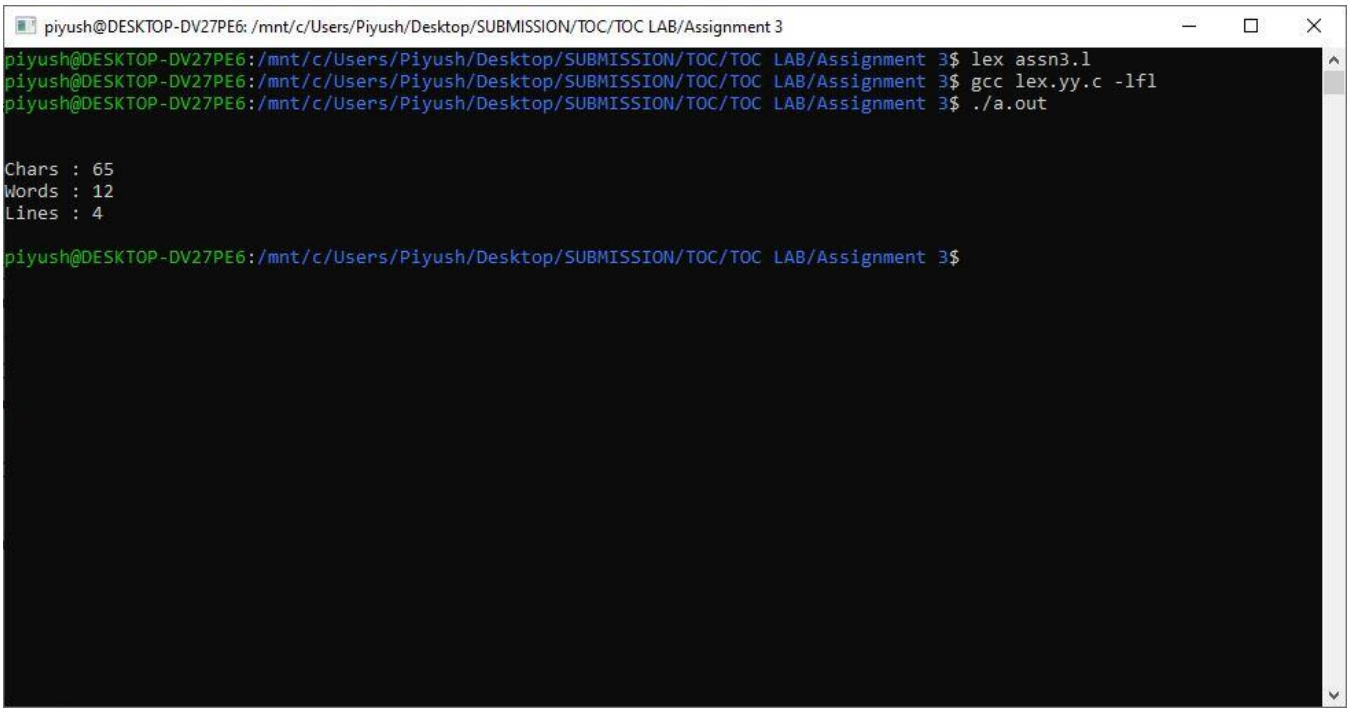
## CODE:

```
%{
int nchars = -1, nwords = 0, nlines = 0;
%}
DIGIT [0-9]+
LINE \n
TAB [ \t\n]+
%%
{LINE}
{TAB}
.
%%
int yywrap()
{
	return 1;
}
int main()
{
	yyin = fopen("myfile.txt","r");		//FILE *yyin
	yylex();
	printf("\n\nChars : %d \nWords : %d \nLines : %d \n\n", nchars, nwords, nlines);
	return 0;
}
```

**INPUT:**

```
Is this gonna work?
Who knows?
Space Tab	NewLine
Hope this works.
```

**OUTPUT**


piyush@DESKTOP-DV27PE6: /mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 3
piyush@DESKTOP-DV27PE6:/mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 3$ lex assn3.l
piyush@DESKTOP-DV27PE6:/mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 3$ gcc lex.yy.c -lfl
piyush@DESKTOP-DV27PE6:/mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 3$ ./a.out
Chars : 65
Words : 12
Lines : 4
piyush@DESKTOP-DV27PE6:/mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 3$