

### Assignment B3

Title : Lexical Analysis - Character Counters.

Date of Completion: 11-02-2020

#### Problem Statement:

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

#### Objectives :

To implement lexical analysis phase of compiler

To understand LEX analysis specifications

To count no. of lines, words, characters

#### Outcomes :

Students will be able to :

Implement Lexical analysis phase of compiler

Understand LEX specifications

Maintain word counter, character count & line count

#### Software & Hardware Requirements:

Fedora 64 bit OS

Eclipse IDE / gedit

i5 Processor

4 GB RAM

500 GB HDD

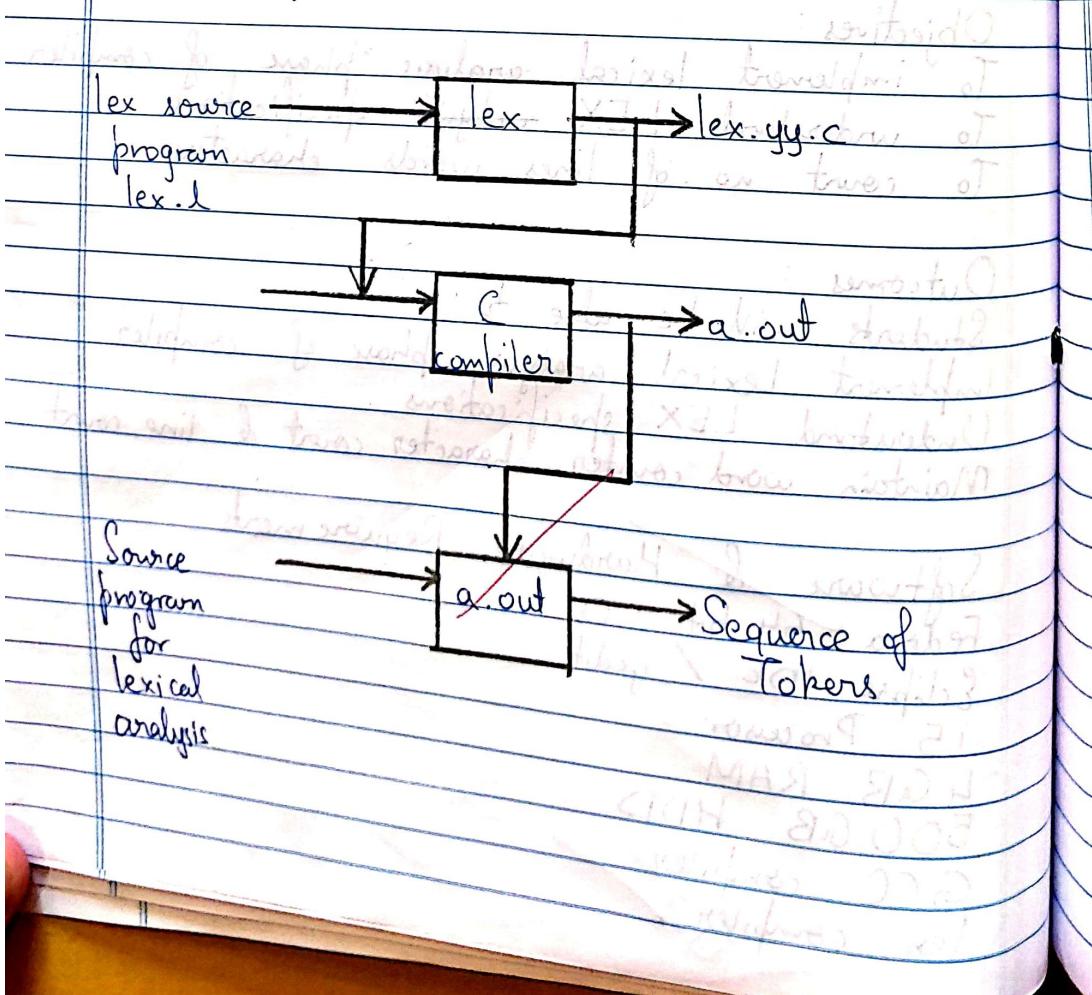
GCC compiler

lex compiler

## Theory :-

Lex is a lexical analyzer generator.  
It is a software tool for compiler construction.  
Designing a compiler from scratch is a very  
cumbersome task & as such many aspects  
of the same can be automated.  
This tool is available under UNIX  
Operating System.

## Working of lex :-



During lexical analysis phase, the compiler reads the input & converts strings in the stream to tokens.

With regular expressions we can specify patterns to lex so that it can generate code that will allow it to scan & match strings in the input.

Each pattern specified in the input to lex has an associated action.

Typically an action returns a token that represents the matched string for subsequent use by the parser.

Lex is a software tool that takes as input a specification of regular expression together with action to be taken on recognising each of these patterns.

The output of lex is a C-program for lexical analysis (lex.y.c)

~~lex.y.c~~ is run through C-Compiler to produce an obj program a.out which is a lexical analyser that transforms an input stream into a sequence of tokens.

## Lex Operators

Operator Notation	Example	Meaning
1 * (asterisk)	a*	Set of all strings of zero/more a's
2   (or)	alb	Either a or b
3 +	a+	Set of all strings of one/more a's
4 ?	a?	Zero/one instance of a
5 [ ]	[abc]	Either a or b or c.

## Patterns of the corresponding functions

```
\n { nline++; } nchar++; } nword++; nchar += eng; } nchar++;
[^t\n]+ { nword++; nchar += eng; } nchar++;
. { nchar++; }
```

## Commands on Ubuntu to run the program

- 1] lex <lex program name>.l
- 2] gcc lex.y.c
- 3] ./a.out <input file>

- Commands to run for a bash file
- 1] chmod +x execute.sh
  - 2] ./execute.sh

### Test Cases :

Description	Input	Output (Expected)	Output (Actual)	Result
Number of characters	Hi I am Apoorv. I am 16. PICT school.	35	35	Success
Number of lines	Hi I am Apoorv I am 21. PICT college.	3	3	Success
Number of words.	Hi I am Apoorv I am 21. PICT college.	9	9	Success

### Conclusion :

We have successfully implemented lexical analysis phase of compiler to count the number of words, characters & lines.

~~Success~~  
13/02