**G. H. RAISONI COLLEGE OF ENGG., NAGPUR**

**(An Autonomous Institute)**

**Department of Computer Science & Engg.**



**Date: 06-09-2021**

**Practical Subject: COMPILER DESIGN**

**Session: 2021-22**

**Student Details:**

| **Roll Number** | 01 |
| --- | --- |
| **Name** | Anand Suralkar |
| **Semester** | 9th |
| **Section** | A |
| **Batch** | CSE |

**Practical Details: Practical Number-2;**

| Practical Aim | Write a lex program to count the no. of new lines, spaces, words and characters. |
| --- | --- |
| Theory & Syntax | LEX  Lex is a program that generates lexical analyzer. It is used with YACC parser generator.  The lexical analyzer is a program that transforms an input stream into a sequence of tokens.  It reads the input stream and produces the source code as output through implementing the lexical analyzer in the C program.  The function of Lex is as follows:  Firstly lexical analyzer creates a program lex.1 in the Lex language. Then Lex compiler runs the lex.1 program and produces a C program lex.yy.c.  Finally C compiler runs the lex.yy.c program and produces an object program a.out.  a.out is lexical analyzer that transforms an input stream into a sequence of tokens.  Specification of lex:  The lex code consists of three parts:-  • Definition section  • Rule section  • User subroutines |
| Program | %{  #include<stdio.h>  int lines=0, spaces=0, words=1, c\_char=0;  %}  %%  [\n] lines++;  [' '] spaces++, words++;  [^' '\n\t] c\_char=c\_char+yyleng;  %%  int main()  {  yylex();  printf("\n No. of total lines: %d", lines);  printf("\n No of total words: %d", words);  printf("\n No of total spaces: %d", spaces);  printf("\n No of total characters: %d\n", c\_char);  return 0;  } |
| Output |  |
| Conclusion | Performed and executed lex program to count the no. of new lines, spaces, words and characters. |