**PSG COLLEGE OF TECHNOLOGY**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**VI SEM – M.Sc SOFTWARE SYSTEMS**

**15XW66 PRINCIPLES OF COMPILER DESIGN LAB**

**LEX TOOL -WORKSHEET - 4**

1. Write lex programs to perform the following.
2. Count the number of characters and lines in the input. Display the counts as output. ***(answer attached example.1)***.
3. Identify the following tokens: Identifier, constants, keywords ***(answer attached for identifier and constants, example.2)***.
4. Accept floating point number with exponent notation.
5. Identity the hexadecimal number. Consider upper or lower case for the digits above 9.
6. Accept all strings of lowercase letters that contain the five vowels in order.
7. Recognize all strings of lowercase letters in which the letters are in ascending lexicographic order
8. Find all strings of digits with at most one repeated digit.
9. Accept all strings of a’s and b’s with an even number of a’s and an odd number of b’s.
10. Recognize all strings of a’s and b’s that do not contain the substring abb.
11. Find all strings of a’s and b’s that do not contain the subsequence abb.
12. Recognize all strings of digits with no repeated digits.

Hits: try this problem first with a few digits, such as {0, 1, 2}

1. Write lex programs to perform the following with files **(refer example program 3)**.
2. Repeat any two programs of Exercise I, by reading inputs from file.
3. Find all comment strings (ie. Strings between /\* …. \*/ and also //… )

III Write a Lex program that copies a file, replacing each non-empty sequence of white space by a single blank.

IV Write a Lex program that copies a C program, replacing each instance of the keyword float by double.

1. **Count the number of characters and lines in the input. Display the counts as output.**

%{

#include<stdio.h>

#include<stdlib.h>

int charcount=0,linecount=0;

%}

%%

. charcount++;

\n {linecount++; charcount++;}

%%

int yywrap() { return 1;}

int main()

{

**yylex();**

printf("There were %d characters in %d lines\n",charcount,linecount);

return 0;

}

1. **Identify the following tokens: Identifier and constants**

%{

#include<stdio.h>

int count=0,c1=0;

%}

digit [0-9]

letter [A-Za-z]

%%

{letter}({letter}|{digit})\* { count++; printf("\n%s is identifier",yytext);}

{digit}+ { c1++; printf("\n%s is number",yytext); }

%%

int main(void) {

yylex();

printf("\nNumber of identifiers = %d\n Number of integer constants: %d", count,c1);

return 0;

}

int yywrap() { return 1; }

1. **Count the number of characters and lines in the input. Display the counts as output. Reading input from file.**

%{

#include<stdio.h>

#include<stdlib.h>

int charcount=0,linecount=0;

%}

%%

. charcount++;

\n {linecount++; charcount++;}

%%

int yywrap() { return 1;}

int main(int n,char \*\* r)

{

FILE \*file;

file = fopen(r[1], "r");

yyin=file;

yylex();

printf("There were %d characters in %d lines\n",charcount,linecount);

return 0;

}