Lex Programs

1) Count of Words Starting with a

Program:

```
// Lex file: aa.l

%{
        int count=0;
%}
alpha    [a-zA-Z]
digit     [0-9]
space    [ \t\n]
start     ^a


%%

{start}                                        {count++;}
{space}(a|A)({alpha}|{digit})*    {count++;}
.                                                ;

%%

main()
{
   yylex();
   printf("count= %d\n",count);
}
```

Output:

```
nn@linuxmint ~ $ lex aa.l
nn@linuxmint ~ $ gcc lex.yy.c -ll
nn@linuxmint ~ $ ./a.out<tst.txt

count= 6
nn@linuxmint ~ $
```

// tst.txt

```
afsal ARIFA aaa www.2k8618.blogspot.com
aiswarya saranya    sooraj
arun reshmi
a www.2k8cse.cu.cc
```

2) Select Lines Ending with 'com'

Program:

```
// Lex file: com.l

%{
int count=0;
%}

%%
.*com\n {count++;ECHO;}
. ;

%%

main()
{
   yylex();
   printf("\nCount= %d\n",count);
   return 0;

}
```

Output:

```
nn@linuxmint ~ $ lex com.l
nn@linuxmint ~ $ gcc lex.yy.c -ll
nn@linuxmint ~ $ ./a.out<com.txt
```

```
www.google.com
www.yahoo.com
www.2k8618.blogspot.com




Count= 3
nn@linuxmint ~ $
```

// **com.txt**

```
www.2k8cs.tk
www.google.com
www.yahoo.com
www.2k8618.blogspot.com
www.2k8cse.cu.cc
```

3) Convert Lowercase to Uppercase & Reverse

Program:

// **Lex file: cap.l**

```
lower [a-z]
CAPS  [A-Z]
space    [ \t\n]

%%
{lower}        {printf("%c",yytext[0]- 32);}
{CAPS}         {printf("%c",yytext[0]+ 32);}
{space}        ECHO;
.              ECHO;
```

```
%%

main()
{
   yylex();

}
```

<u>Output:</u>

```
nn@linuxmint ~ $ lex cap.l
nn@linuxmint ~ $ gcc lex.yy.c -ll
nn@linuxmint ~ $ ./a.out<tst.txt
WWW.2K8618.BLOGSPOT.COM
sanjana    jamsheena chaithanya neethu
GOVINDAPRASAD VIPIN ADARSH SHIVIN
baby brinda kavya helen
SALMAN TINU RICHARD  SIBIN
SHIVIN laji NABEEL
www.2k8cse.cu.cc
nn@linuxmint ~ $
```

// **tst.txt**

```
www.2k8618.blogspot.com
SANJANA    JAMSHEENA CHAITHANYA NEETHU
govindaprasad vipin adarsh shivin
BABY BRINDA KAVYA HELEN
salman tinu richard  sibin
shivin LAJI nabeel
WWW.2K8CSE.CU.CC
```

4)Number Each Line

Program:

```
%{
int lineno =1;
%}

line .*\n

%%
{line} {printf("%5d %s", lineno++,yytext);}

%%
main()
{
yylex();
return 0;
}
```

Output:

input: compilerdesign.txt

```
Design of a Lexical Analyzer using Finite Automation
Design of lexical analyzer using LEX
Design of recursive descent and LL (1) parsers
Implementation of Operator precedence Parsing
Design of parser for arithmetic expressions using YACC
Design of a simple type checker
Generation of IC for arithmetic expressions
Simple code optimization strategies
Design of a code generator
Writing a simple Compiler
```

```
nn@linuxmint ~ $ lex l1.l
nn@linuxmint ~ $ gcc lex.yy.c -lfl
nn@linuxmint ~ $ ./a.out <compilerdesign.txt
    1 Design of a Lexical Analyzer using Finite Automation
    2 Design of lexical analyzer using LEX
    3 Design of recursive descent and LL (1) parsers
    4 Implementation of Operator precedence Parsing
    5 Design of parser for arithmetic expressions using YACC
    6 Design of a simple type checker
    7 Generation of IC for arithmetic expressions
    8 Simple code optimization strategies
    9 Design of a code generator
   10 Writing a simple Compiler
   11                    www.2k8618.blogspot.com
```

5) Count The Number of lines ending with "com"

Program:

//**Lex file: com.l**

```
%{
int count=0;
%}
DIGIT [0-9]
ALPHA [a-zA-Z]
%%
({ALPHA}|{DIGIT})*com {count++;}
%%

main()
{
    yylex();
    printf("Count= %d\n",count);
    return 0;
}
```

Output :

```
n@linuxmint ~ $ lex com.l
nn@linuxmint ~ $ gcc lex.yy.c -ll
nn@linuxmint ~ $ ./a.out<com.txt
www.2k8618.blogspot.
www.2k8cs.tk
www.google.
www.gmail.

Count= 3
nn@linuxmint ~ $
```

6) **LEX Program to identify Keywords and convert it into uppercase**

```
%{#include<stdio.h>
int i;
%}keyword main|int|scanf|printf|if|else
%%

{keyword} {
 for(i=0;i<yyleng;i++)
 printf("%c",toupper(yytext[i]));
   }
%%

main()
{
yyin=fopen("num.c","r");
yylex();
}
```

```
int yywrap()
{
return 1;
}
```

<u>Output</u>Let num.c contains following program fragment.

```
main()
{
int num;
scanf("%d",&num);
if(num%2)printf("Odd");
else printf("Even")
}
```

The output will be,

```
MAIN()
{
INT num;
SCANF("%d",&num);
IF(num%2)PRINTF("Odd");
ELSE PRINTF("Even")
}
```