**Practical-4**

1. **Write a lex program to count number of words, characters and lines from user input.**

%{

#include<stdio.h>

int count=0;

int letter=0;

int lcount=0;

%}

%%

[a-zA-Z0-9] {letter++;}

" " {count++;}

"\n" {lcount++; count++; printf("\n Total words are: %d\n Total Letters are: %d\n Total Lines are: %d\n",count,letter,lcount); lcount=0; count=0; letter=0;}

%%

int yywrap()

{

return 1;

}

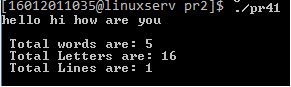
int main()

{

yylex();

return 0;

}



1. **Write a lex program to count number of words, characters and lines from File.**

%{

#include<stdio.h>

int count=0;

int letter=0;

int lcount=0;

%}

%%

[a-zA-Z0-9] {letter++;}

" " {count++;}

"\n" {lcount++; count++; printf("\n Total words are: %d\n Total Letters are: %d\n Total Lines are: %d\n",count,letter,lcount); lcount=0; count=0; letter=0;}

%%

int yywrap()

{

return 1;

}

int main()

{

yyin=fopen("test.txt","r");

yylex();

return 0;

}



1. **Write a lex program to convert lower case letter to upper case from user input and terminate the program if user enters 0.**

%{

#include<stdio.h>

%}

%%

[a-z] printf("%c",yytext[0] - ('a' - 'A'));

[A-Z] printf("%c",yytext[0] - ('A' - 'A'));

0 { return 0;}

%%

int main()

{

yylex();

return 0;

}



1. **Write a lex program to convert lower case letter of given file to upper case.**

%{

#include<stdio.h>

%}

%%

/\*\*\* Rules section \*\*\*/

[a-z] printf("%c",yytext[0] - ('a' - 'A'));

[A-Z] printf("%c",yytext[0] - ('A' - 'A'));

0 { return 0;}

%%

int main()

{

FILE \*fp;

fp = fopen("test.txt", "r");

if (fp == NULL) { printf("File not found"); }

yyin = fp;

yylex();

return 0;

}



1. Write a lex program to check whether IP address entered by user is valid or not.

%{

#include<stdio.h>

%}

%%

^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$ printf("Valid IP Address");

[a-zA-Z0-9.]\* printf("Invaild IP Address");

0 { return 0;}

%%

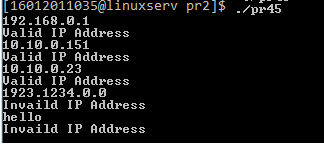
int main()

{

yylex();

return 0;

}



1. **Write a lex program to validate IP addresses from user specified file.**

%{

#include<stdio.h>

%}

%%

^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$ printf("Vaild IP Address");

[A-Za-z0-9.]\* printf("Invalid IP Address");

%%

int main()

{

FILE \*fp;

fp = fopen("ipaddress.txt", "r");

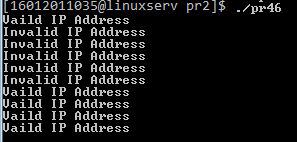
if (fp == NULL) { printf("File not found"); }

yyin = fp;

yylex();

return 0;

}



1. **Write a lex program to check whether Date entered by user in format (dd/mm/yyyy) is valid or not.**

%{

#include<stdio.h>

%}

%%

((0[1-9])|([1-2][0-9])|(3[0-1]))\/((0[1-9])|(1[0-2]))\/(19[0-9]{2}|2[0-9]{3}) printf("Valid");

.\* printf("Invalid");

%%

int main()

{

yylex();

return 0;

}

