**LAB ASSIGNMENT-3**

**Q. Count the number of words starting with A**

%{

#include<stdio.h>

int count=0;

%}

%%

((^A)|([ \t\n](A))) {count++;}

.

%%

int main(int argc,char\*\*argv)

{

FILE\*fp=fopen(argv[1],"r");

yyin=fp;

yylex();

printf("Number of word starting with A: %d\n",count);

return 0;

fclose(fp);

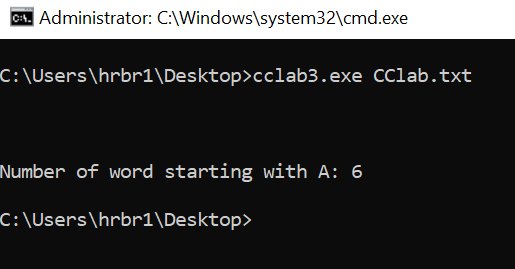
}

int yywrap()

{

return 1;

}



**LAB ASSIGNMENT-4**

**Q. Convert lower case alphabets to upper case and vice versa**

%option noyywrap

%{

#include<stdio.h>

%}

lower [a-z]

upper [A-Z]

%%

{lower} {printf("%c",yytext[0]-32);}

{upper} {printf("%c",yytext[0]+32);}

%%

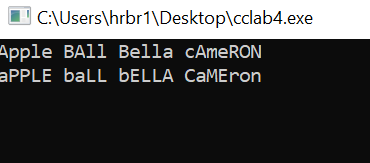
int main()

{

yylex();

return 0;

}



**LAB ASSIGNMENT-5**

**Q. Convert decimal to hexadecimal and vieversa**

%option noyywrap

%{

#include<stdio.h>

%}

digit [0-9]

number {digit}+

%%

{number} {int n =atoi(yytext);

printf("%x",n);}

. {;}

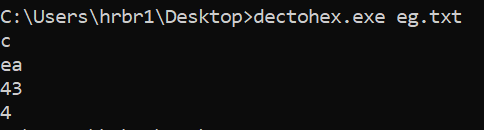
%%

int main()

{

return 0;

}



%option noyywrap

%{

#include <stdio.h>

#include <string.h>

#include <math.h>

int decimal = 0, i = 0, val, temp;

char \*hex;

%}

hex [A-Fa-f0-9]

%%

{hex}+ {

hex = yytext;

for (i = 0; i < strlen(hex); i++)

{

switch(hex[i])

{

case 'A':

case 'a': val = 10; break;

case 'B':

case 'b': val = 11; break;

case 'C':

case 'c': val = 12; break;

case 'D':

case 'd': val = 13; break;

case 'E':

case 'e': val = 14; break;

case 'F':

case 'f': val = 15; break;

case '1': val = 1; break;

case '2': val = 2; break;

case '3': val = 3; break;

case '4': val = 4; break;

case '5': val = 5; break;

case '6': val = 6; break;

case '7': val = 7; break;

case '8': val = 8; break;

case '9': val = 9; break;

case '0': val = 0; break;

}

decimal = decimal + (val \* (double)pow(16, strlen(hex) - i - 1));

}

printf("%d", decimal);

decimal = 0;

}

%%

int main(int argc, char \*\*argv)

{

FILE\*fp=fopen(argv[1],"r");

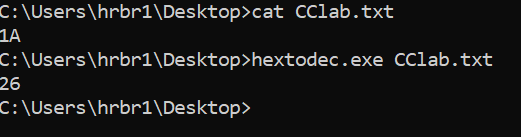
yyin=fp;

yylex();

return 0;

fclose(fp);

}



%{

#include<string.h>

#include<stdlib.h>

void stringToDecToHex(char\* c){

char\* cherry = malloc(yyleng );

strncpy(cherry, c, yyleng);

int a = atoi(cherry);

printf("\ndec to hex: 0x%x\n", a);

}

void stringToHextoDec(char\* c){

char\* cherry = malloc(yyleng );

strncpy(cherry, c, yyleng);

int a = strtol(cherry, NULL, 16);

printf("\nhex to dec: %d\n", a);

}

%}

hexnum 0[xX][0-9A-F]+?[\n]

decnum [1-9][0-9]\*?[\n]

eol \n

%%

{eol} {return 1;}

{decnum} { stringToDecToHex(yytext);}

{hexnum} { stringToHextoDec(yytext);}

%%

int main(){

yylex();

}

int yywrap(){

return 1;

}

**LAB ASSIGNMENT-6**

**Q. Count the number of lines ending .com, .org,.in, .edu**

%option noyywrap

%{

#include<stdio.h>

int com\_count=0,org\_count=0,in\_count=0,edu\_count=0;

%}

%%

.\*org {org\_count++;ECHO;}

.\*com {com\_count++;ECHO;}

.\*edu {edu\_count++;ECHO;}

.\*in {in\_count++;ECHO;}

.;

%%

int main(int argc,char\*\*argv)

{

FILE\*fp=fopen(argv[1],"r");

yyin=fp;

yylex();

printf("\n com count %d",com\_count);

printf("\n org count %d",org\_count);

printf("\n in count %d",in\_count);

printf("\n edu count %d",edu\_count);

return 0;

fclose(fp);

}

