BASICS::/

Enter lex programme in notepad

And save it in the c user; see what name comes in cmd, and save it in that(with ---.l.txt)))

Step 1>>>> set path=C:\Program Files (x86)\GnuWin32\bin

Step 2>>>> flex (file name)

Ex: calculator programme means(flex calculator.l.txt)

Step 3>>>set path=C:\Program Files (x86)\MinGW\bin

Step 4>>>> gcc lex.yy.c

Steps5>>> a or direct inputs….

\*\*\*Mathematical operations(( add, subb, div, mul))……….

%{

#include <stdio.h>

#include <stdlib.h>

int f1=0,f2=0; char oper; float op1,op2,ans;

void eval() { if(oper=='+') ans=op1+op2; else if(oper=='-') ans=op1-op2;

else if(oper=='\*') ans=op1\*op2; else if(oper=='/' && op2!=0) ans=op1/op2;

else { printf("ERROR\n"); return; } printf("Result: %lf\n", ans); }

%}

%%

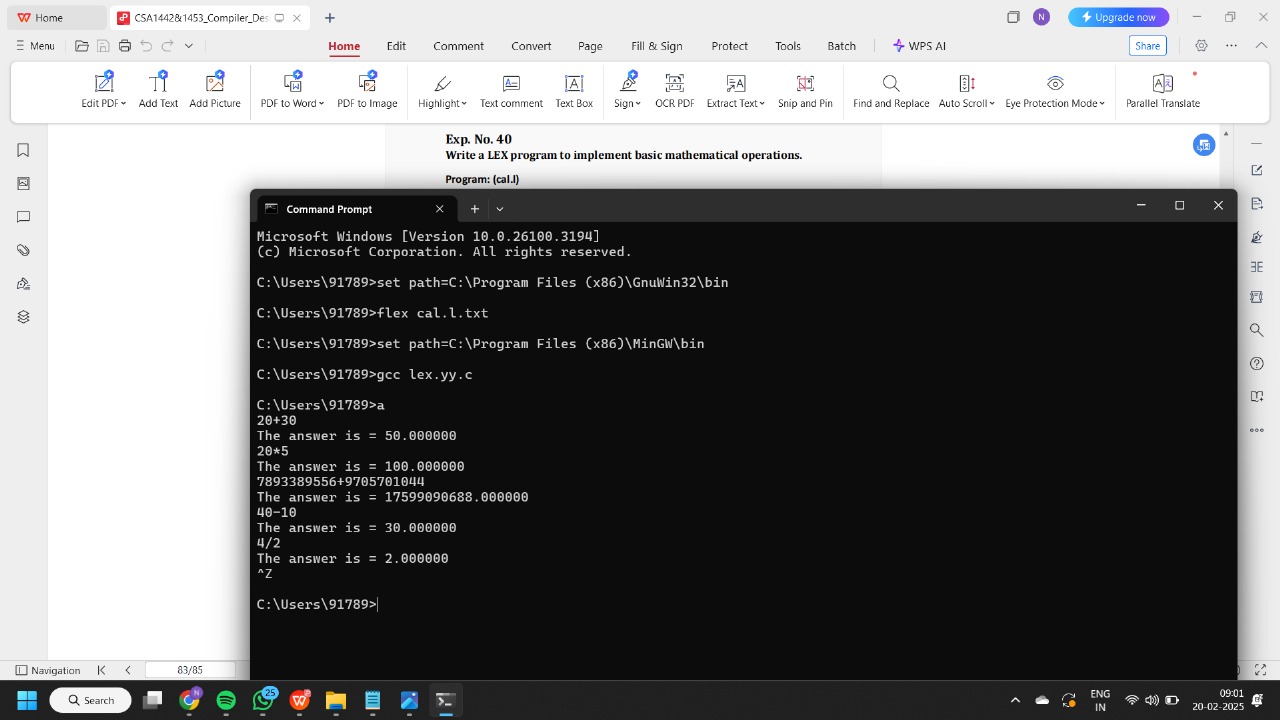
[0-9]+(\.[0-9]+)? { if(!f1) { op1=atof(yytext); f1=1; } else { op2=atof(yytext); f2=1; eval(); f1=f2=0; } }

[\*/+-] { oper=\*yytext; }

%%

int main() { yylex(); return 0; }

int yywrap() { return 1; }



\*\*\* valid digit or not………..

%%

[0-9]+ {printf("\nValid digit \n");}

.\* printf("\nInvalid digit\n");

%%

int yywrap(){}

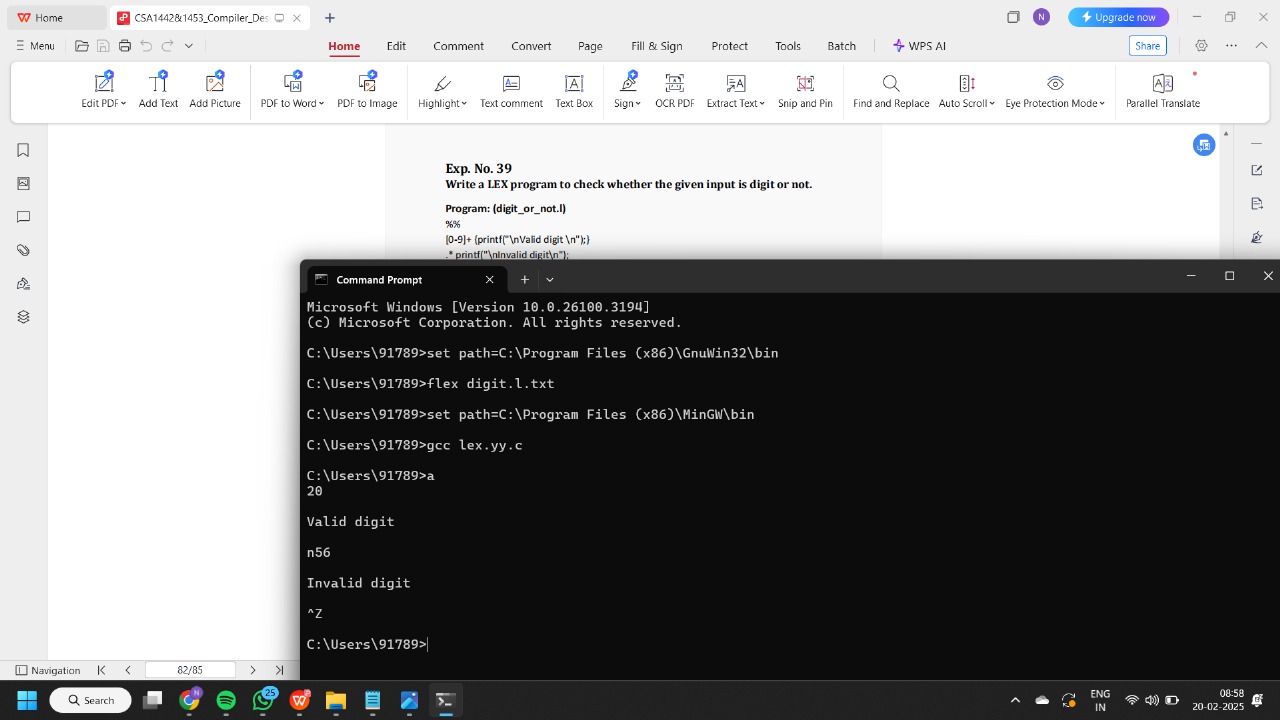
int main()

{

yylex();

return 0;

}



\*\*\*VALID DOB OR NOT>>>

%{

#include <stdio.h>

%}

%%

[0-2][0-9]-[0-1][0-9]-[0-9]{4} { printf("Valid DOB: %s\n", yytext); }

30-0[1-9]|30-1[0-2]-[0-9]{4} { printf("Valid DOB: %s\n", yytext); }

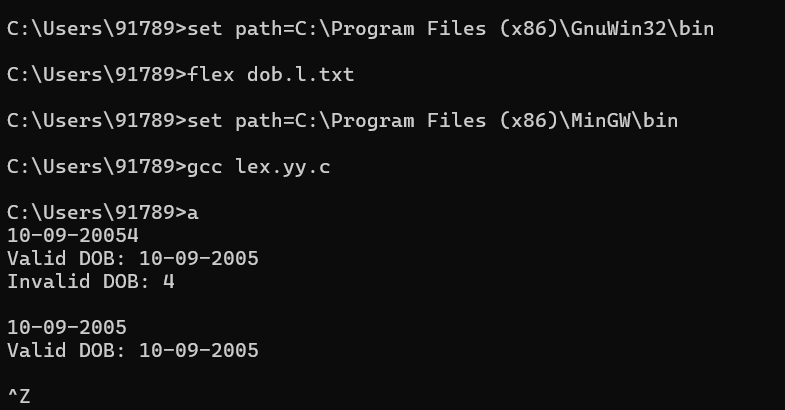
31-0[13578]|31-1[02]-[0-9]{4} { printf("Valid DOB: %s\n", yytext); }

. { printf("Invalid DOB: %s\n", yytext); }

%%

int main() { yylex(); return 0; }

int yywrap() { return 1; }



>>>VALID URL OR NOT

%%

((http)|(ftp))s?:\/\/[a-zA-Z0-9](.[a-z])+(.[a-zA-Z0-9+=?]\*)\* {printf("\nURL Valid\n");}

.+ {printf("\nURL Invalid\n");}

%%

void main()

{

printf("\nEnter URL : ");

yylex();

printf("\n");

}

int yywrap()

{

}

I/P ::== Redirected git hub link

>>>COUNT NO. OF +VE AND -VE NO.S

%{

int positive\_no = 0, negative\_no = 0;

%}

%%

^[-][0-9]+ {negative\_no++;

printf("negative number = %s\n",

yytext);} // negative number

[0-9]+ {positive\_no++;

printf("positive number = %s\n",

yytext);} // positive number

%%

int yywrap(){}

int main()

{

yylex();

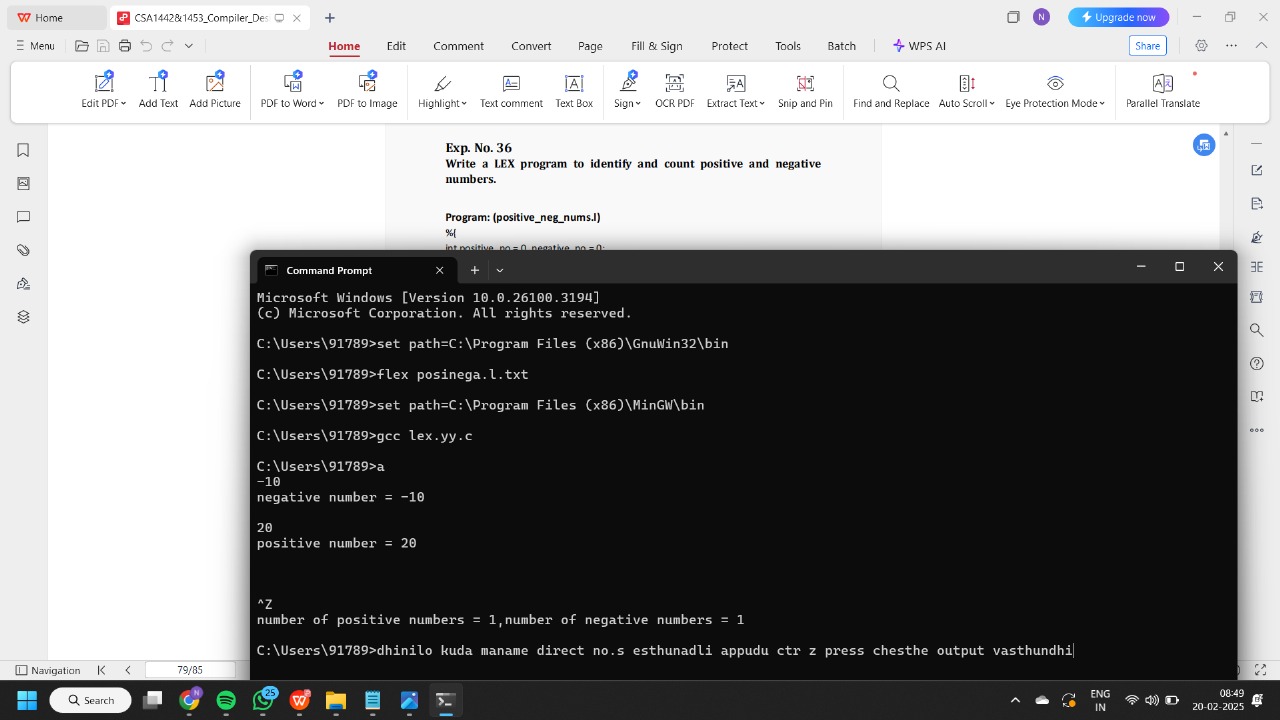
printf ("number of positive numbers = %d,"

"number of negative numbers = %d\n",

positive\_no, negative\_no);

return 0;

}



>>>Recognize words and no.s in statement

%%

[\t ]+ ;

[0-9]+|[0-9]\*\.[0-9]+ { printf("\n%s is NUMBER", yytext);}

#.\* { printf("\n%s is COMMENT", yytext);}

[a-zA-Z]+ { printf("\n%s is WORD", yytext);}

\n { ECHO;}

%%

int main()

{

while( yylex());

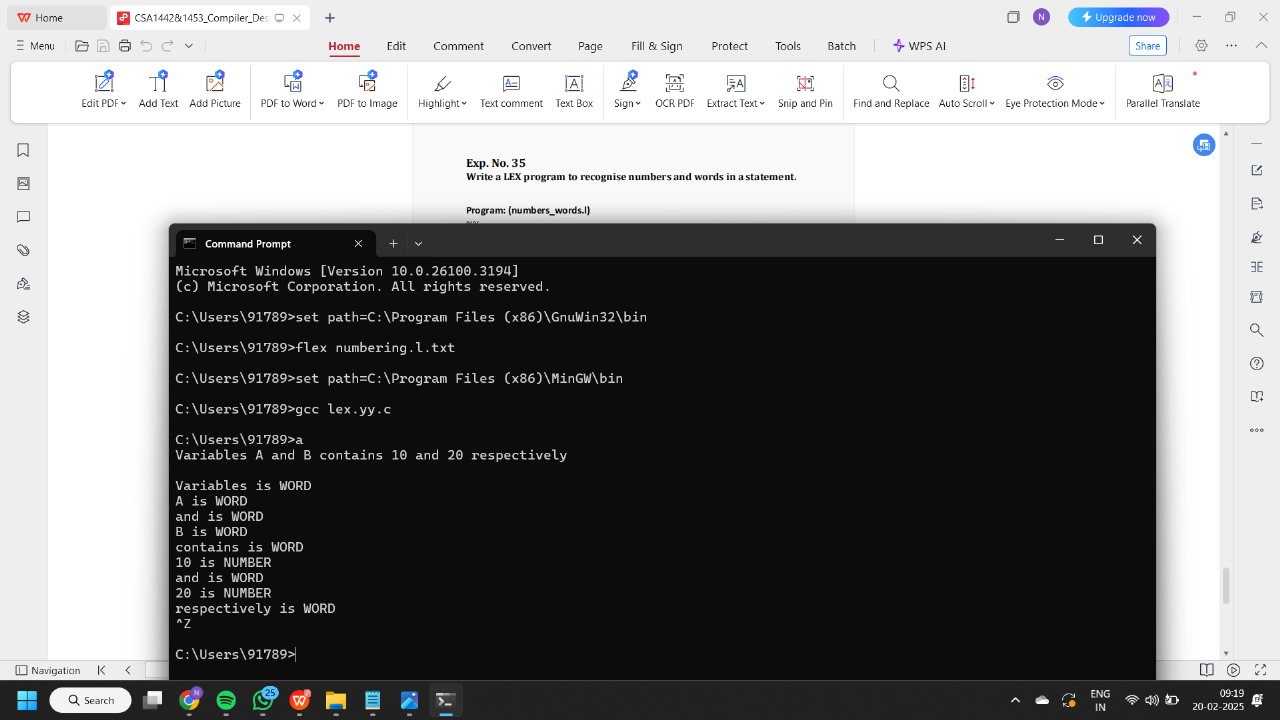
}

int yywrap( )

{

return 1;

}



>>>count no. of keywords and identifiers..

digit [0-9]

letter [A-Za-z]

%{

int count\_id,count\_key;

%}

%%

(stdio.h|conio.h) { printf("%s is a standard library\n",yytext); }

(include|void|main|printf|int) { printf("%s is a keyword\n",yytext); count\_key++; }

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

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

\"(\\.|[^"\\])\*\" { printf("%s is a string literal\n", yytext); }

.|\n { }

%%

int yywrap(void) {

return 1;

}

int main(int argc, char \*argv[]) {

yyin = fopen(argv[1], "r");

yylex();

printf("number of identifiers = %d\n", count\_id);

printf("number of keywords = %d\n", count\_key);

fclose(yyin);

}

Ip code is >>

#include<stdio.h>

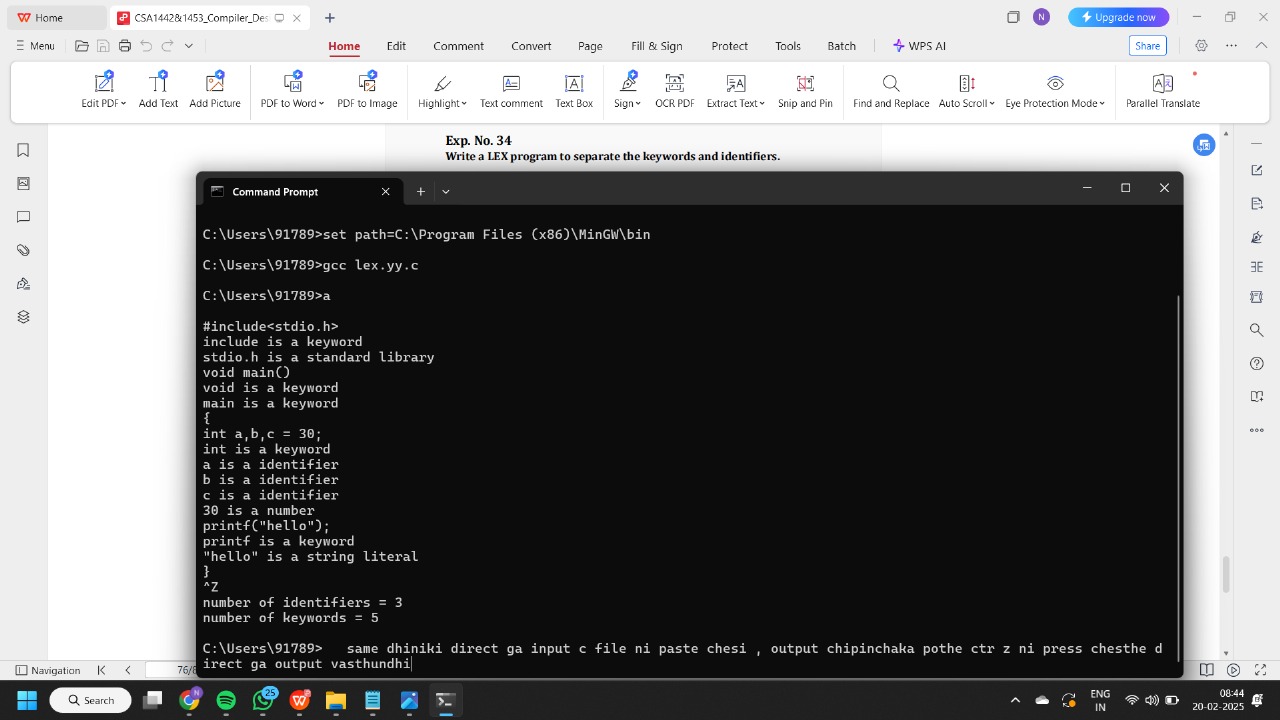
void main()

{

int a,b,c = 30;

printf("hello");

}



>>>>COUNT NO. OF VOWELS AND CONSONANTS

%{

int vow\_count=0;

int const\_count =0;

%}

%%

[aeiouAEIOU] {vow\_count++;}

[a-zA-Z] {const\_count++;}

%%

int yywrap(){}

int main()

{

printf("Enter the string of vowels and consonants:");

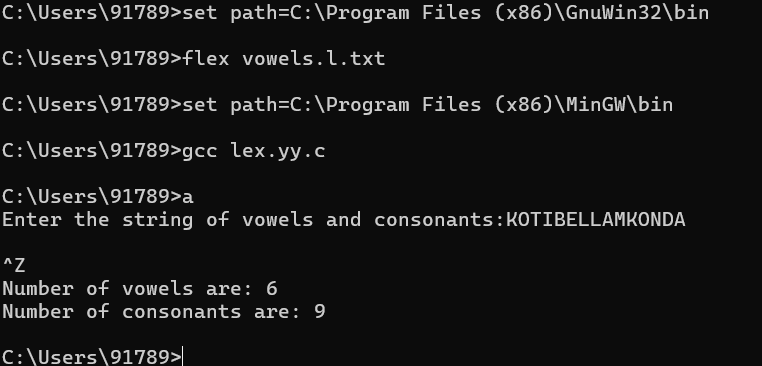
yylex();

printf("Number of vowels are: %d\n", vow\_count);

printf("Number of consonants are: %d\n", const\_count);

return 0;

}



>>>Valid Mobile or not

%{

%}

%%

[6-9][0-9]{9} {printf("\n mobile number valid\n");}

.+ {printf("\n mobile number invalid\n");}

%%

int yywrap(void){}

int main()

{

printf("\n enter the mobile number:");

yylex();

printf("\n");

return 0;

}

>>CONVERT abc to ABC

%{

int i;

%}

%%

[a-z A-Z]\* { for(i=0;i<=yyleng;i++)

{ if((yytext[i]=='a')&&(yytext[i+1]=='b')&&(yytext[i+2]=='c'))

{ yytext[i]='A';

yytext[i+1]='B';

yytext[i+2]='C';

}

}

printf("%s",yytext);

}

[\t]\* return 1;

.\* {ECHO;}

\n {printf("%s",yytext);}

%%

int main()

{

yylex();

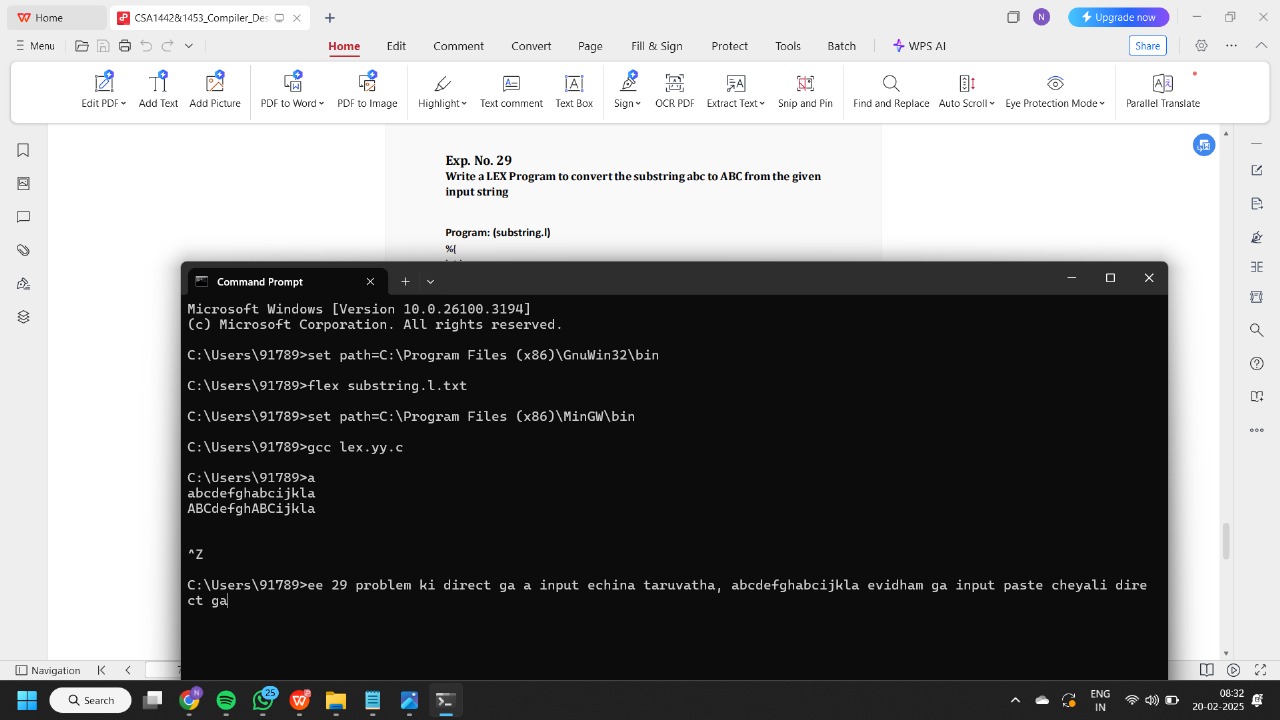
}

int yywrap()

{

return 1;

}



>>> VALID GMAIL OR NOT>>

%{

%}

%%

[a-z.0-9\_]+@[a-z]+".com"|".in" { printf("it is valid");}

.+ { printf("it is not valid");}

%%

int yywrap(){}

int main()

{

printf("enter the mail:");

yylex();

}

>>> IDENTIFY CAPITAL WORDS IN GIVEN INPUT….

%%

[A-Z]+[\t\n ] { printf("%s is a capital word\n",yytext); }

. ;

%%

int main( )

{

printf("Enter String :\n");

yylex();

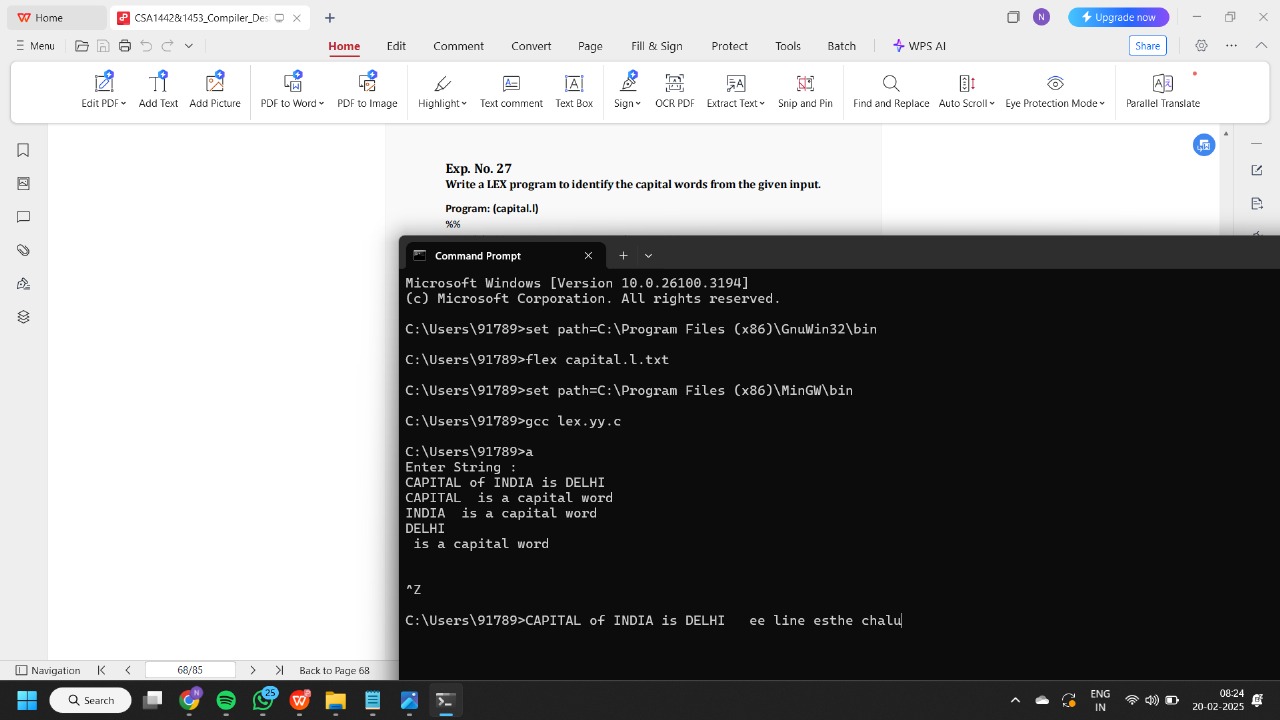
}

int yywrap( )

{

return 1;

}



>>>>NUMBER OF COMMENTS LINES

%{

int com=0;

%}

%s COMMENT

%%

"/\*" {BEGIN COMMENT;}

<COMMENT>"\*/" {BEGIN 0; com++;}

<COMMENT>\n {com++;}

<COMMENT>. {;}

\/\/.\* {; com++;}

.|\n {fprintf(yyout,"%s",yytext);}

%%

void main(int argc, char \*argv[])

{

if(argc!=3)

{

printf("usage : a.exe input.c output.c\n");

exit(0);

}

yyin=fopen(argv[1],"r");

yyout=fopen(argv[2],"w");

yylex();

printf("\n number of comments are = %d\n",com);

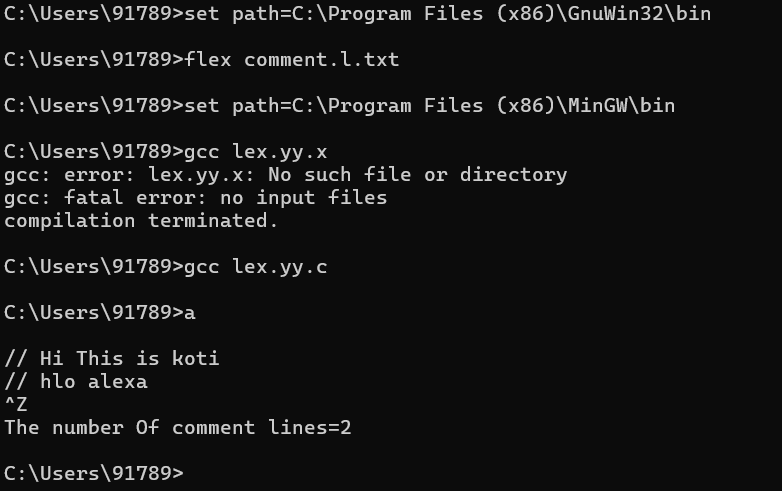
}

int yywrap()

{

return 1;

}



>>ADDING LINE NUMBERS

%{

int yylineno;

%}

%%

^(.\*)\n printf("%4d\t%s", ++yylineno, yytext);

%%

int yywrap(void) {

return 1;

}

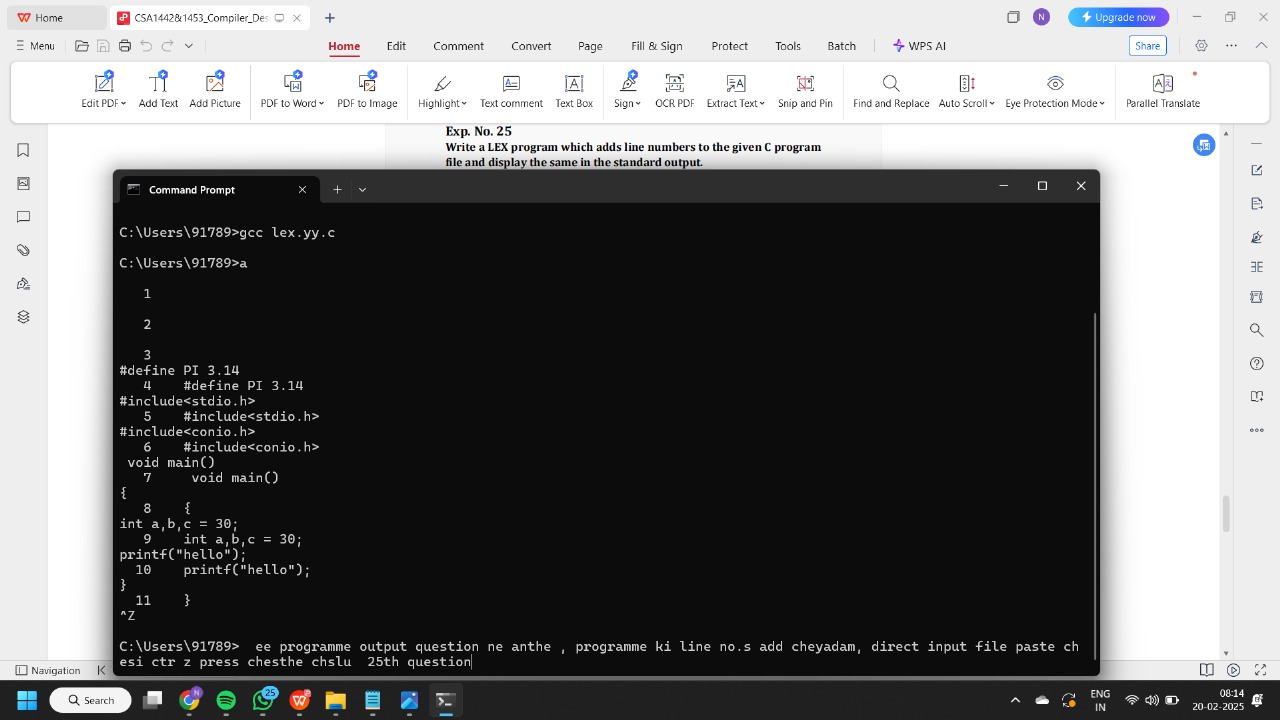
int main(int argc, char \*argv[]) {

yyin = fopen(argv[1], "r");

yylex();

fclose(yyin);

}



>>>HTML TAGS IN IP CODE

%{

int tags;

%}

%%

"<"[^>]\*> { tags++; printf("%s \n", yytext); }

.|\n { }

%%

int yywrap(void) {

return 1; }

int main(void)

{

FILE \*f;

char file[10];

printf("Enter File Name : ");

scanf("%s",file);

f = fopen(file,"r");

yyin = f;

yylex();

printf("\n Number of html tags: %d",tags);

fclose(yyin);

}

>>>>i/p code

<html>

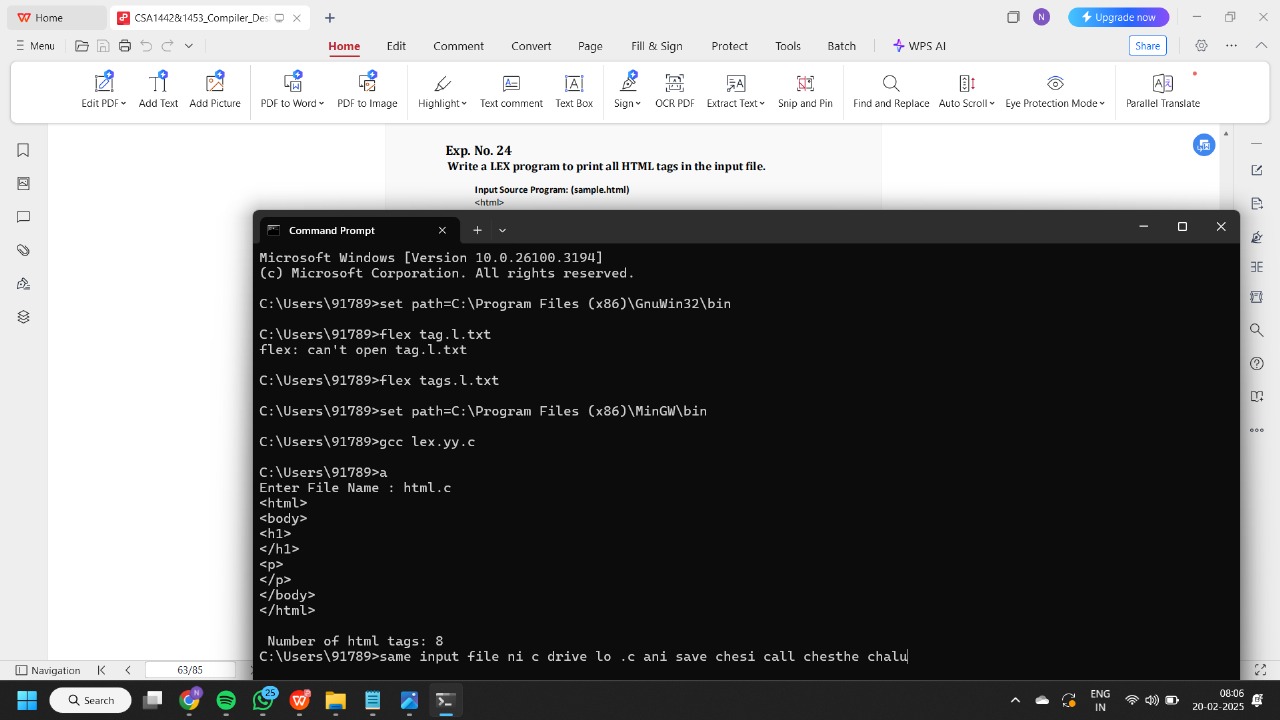
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>

</html>



>>>>COUNT MACRO LINES….

%{

int nmacro, nheader;

%}

%%

^#define { nmacro++; }

^#include { nheader++; }

.|\n { }

%%

int yywrap(void) {

return 1;

}

int main(int argc, char \*argv[]) {

yyin = fopen(argv[1], "r");

yylex();

printf("Number of macros defined = %d\n", nmacro);

printf("Number of header files included = %d\n", nheader);

fclose(yyin);

}

>>>>IP CODE

#define PI 3.14

#include<stdio.h>

#include<conio.h>

void main()

{

int a,b,c = 30;

printf("hello");

}

>>>>COUNT NUMBER OF WORDS, LINES, AND CHRS

%{

int nchar, nword, nline;

%}

%%

\n { nline++; nchar++; }

[^ \t\n]+ { nword++, nchar += yyleng; }

. { nchar++; }

%%

int yywrap(void) {

return 1;

}

int main(int argc, char \*argv[]) {

yyin = fopen(argv[1], "r");

yylex();

printf("Number of characters = %d\n", nchar);

printf("Number of words = %d\n", nword);

printf("Number of lines = %d\n", nline);

fclose(yyin);

}

IP CODE://

#include <stdio.h>

int main()

{

int number1, number2, sum;

printf("Enter two integers: ");

scanf("%d %d", &number1, &number2);

sum = number1 + number2;

printf("%d + %d = %d", number1, number2, sum);

return 0;

}