**DAY 3**

**EXPERIMENT-17**: LEX specification file to take input C program from a .c file and count the number of characters, number of lines & number of words.

%{

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();

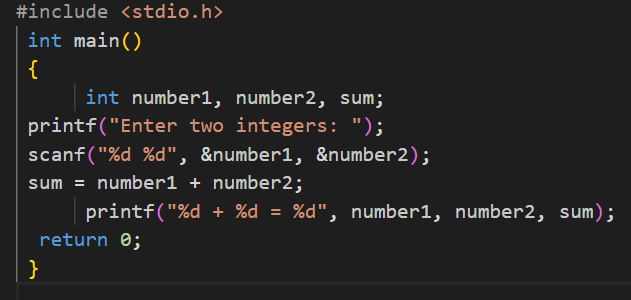
printf("Number of characters = %d\n", nchar);

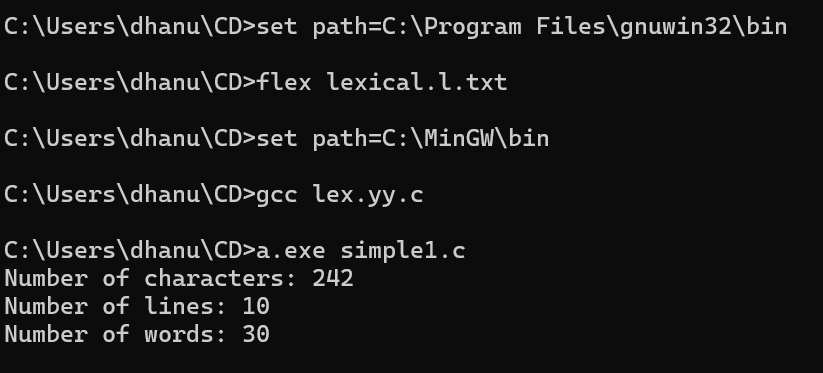
printf("Number of words = %d\n", nword);

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

fclose(yyin);

}





**EXPERIMENT-18**: LEX program to print all the constants in the given C source program file.

digit [0-9]

%{

int cons=0;

%}

%%

{digit}+ { cons++; printf("%s is a constant\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");

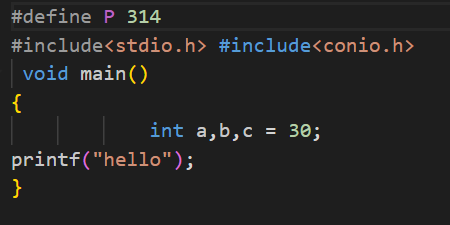
yyin = f;

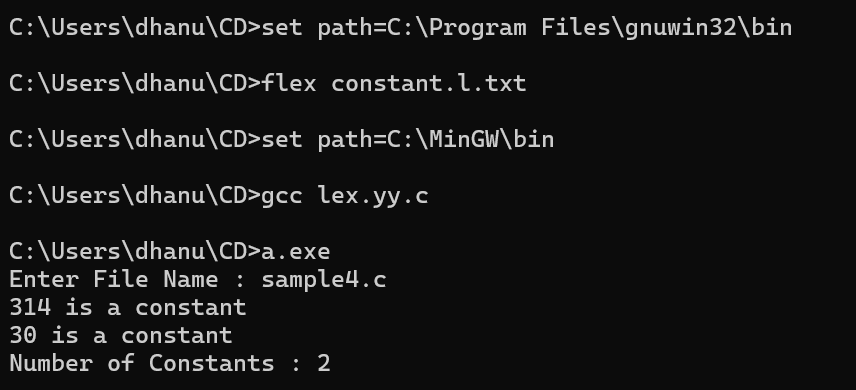
yylex();

printf("Number of Constants : %d\n", cons);

fclose(yyin);

}





**EXPERIMENT-19**: LEX program to count the number of Macros defined and header files included in the C program.

%{

int nmacro, nheader;

%}

%%

^#define { nmacro++; }

^#include { nheader++; }

.|\n { }

%%

int yywrap(void) {

return 1;

}

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

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

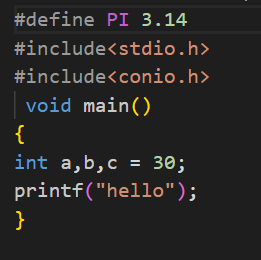
yylex();

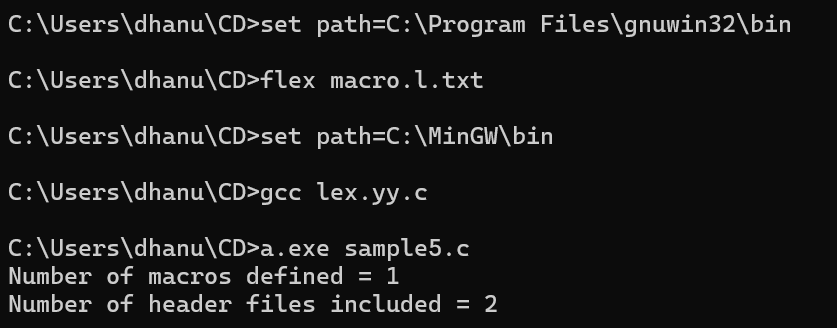
printf("Number of macros defined = %d\n", nmacro);

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

fclose(yyin);

}





**EXPERIMENT-20**: LEX program to print all HTML tags in the input file.

%{

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");

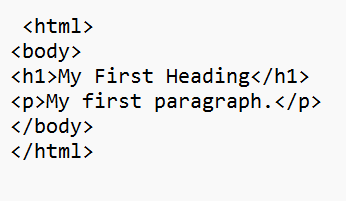
yyin = f;

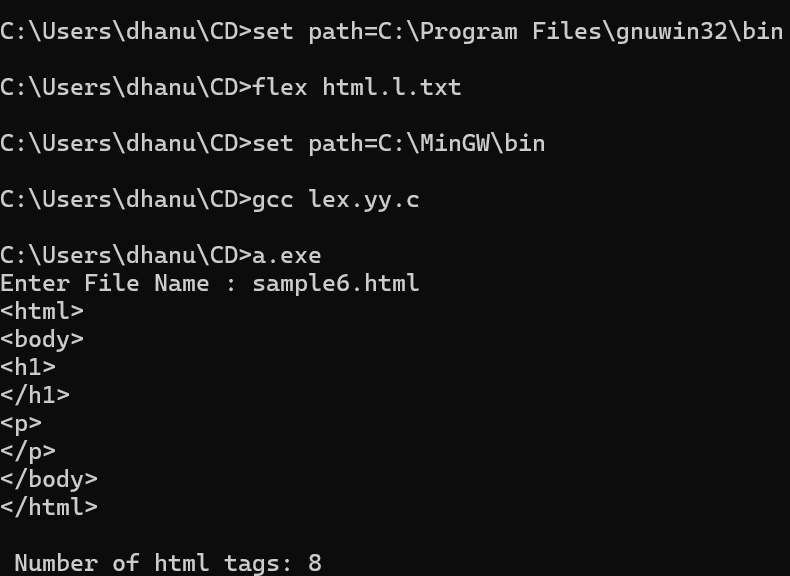
yylex();

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

fclose(yyin);

}





**EXPERIMENT-21**: LEX program which adds line numbers to the given C program file and display the same in the standard output.

%{

int ln=0;

%}

%%

.\* {ln++; fprintf(yyout,"\n%d:%s",ln,yytext);}

%%

int yywrap(){}

int main()

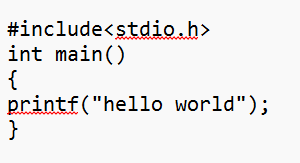
{

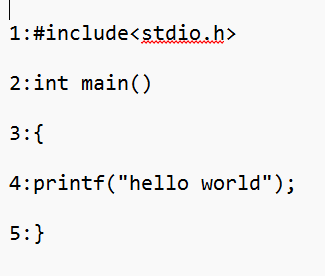
yyin=fopen("simple.txt","r");

yyout=fopen("out.txt","w");

yylex();

}





**EXPERIMENT-22**: LEX program to count the number of comment lines in a given C program and eliminate them and write into another file.

%{

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);

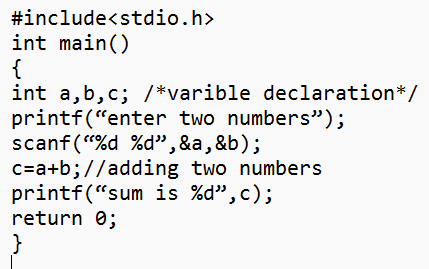
}

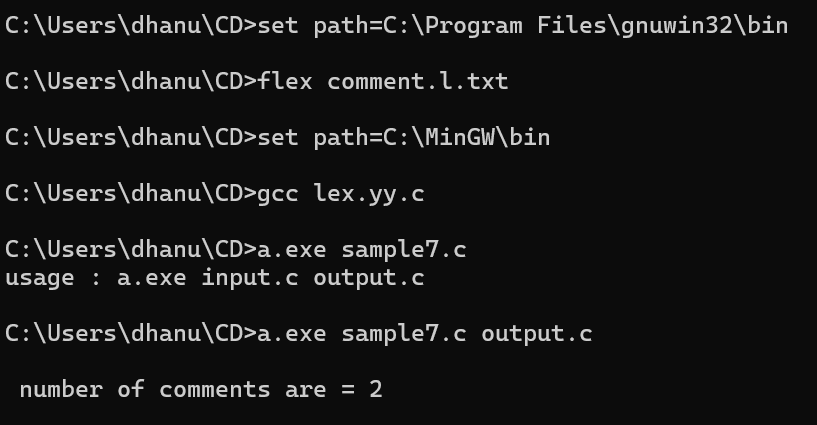
int yywrap()

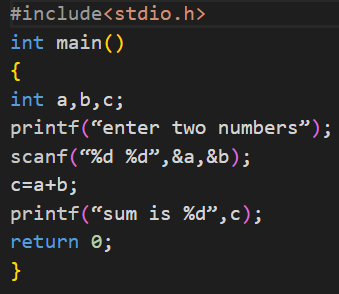
{

return 1;

}







**EXPERIMENT-23**: LEX program should separate the tokens in the given C program and display with appropriate caption.

%{

int n=0;

%}

%%

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

(void|main|printf|int|while|if|else|float) {n++; printf("%s is a keyword\n",yytext);}

[a-zA-Z\_][a-zA-Z0-9\_]\* {n++; printf("%s is a identifier\n", yytext);}

[0-9]+ {n++; printf("%s is a integer\n", yytext);}

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

"<="|"=="|"="|"+"|"-"|"\*"|"/"|"++" {n++; printf("%s is a operator\n", yytext);}

[(){}|,;] {n++;printf("%s is a separator\n", yytext);}

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

.|\n { }

%%

int yywrap(void)

{

return 1;

}

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

{

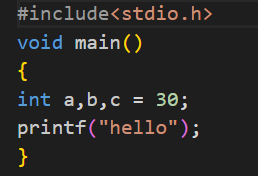
yyin = fopen(argv[1], "r");

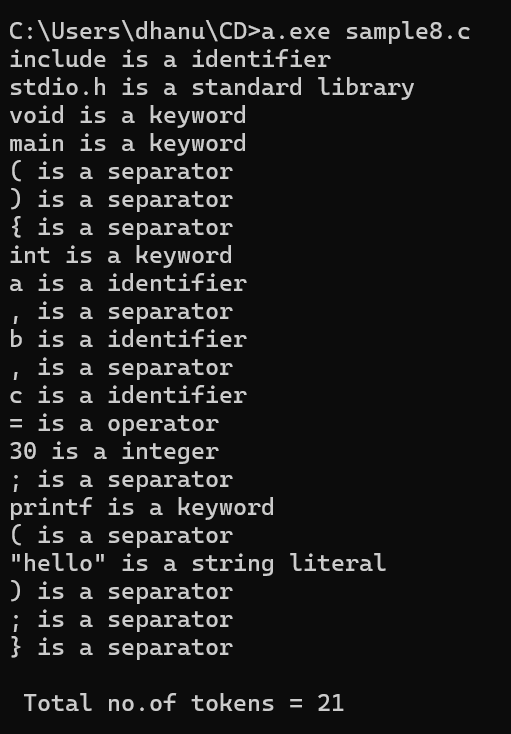
yylex();

printf("\n Total no.of tokens = %d\n", n);

fclose(yyin);

}





**EXPERIMENT-24**: LEX specification file to take input C program from a .c file and count the number of characters, number of lines & number of words.

%{

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);

}

