# Assignment – 1

Objective: Design a LEX Code to count the number of lines, space, tab-meta character and rest of characters in a given Input pattern.

Code:

%{

#include<stdio.h>

int lc=0, sc=0, tc=0, chc=0;

%}

%%

\n lc++; [ ] sc++;

\t tc++;

. chc++;

%%

int yywrap(void) {}

int main()

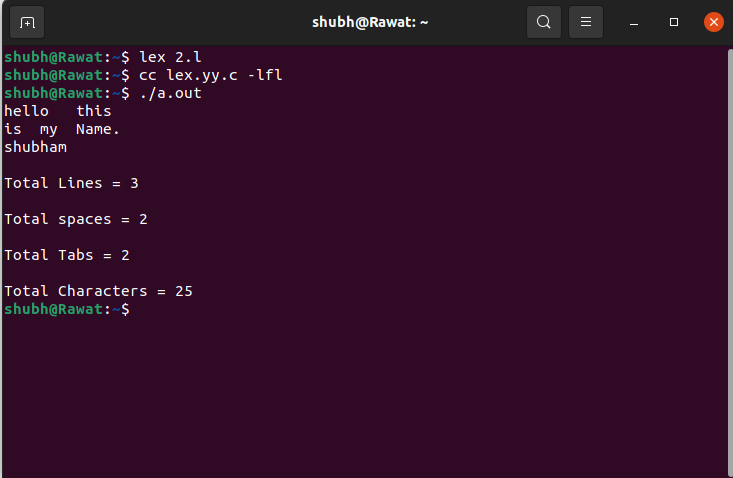
{

yylex();

printf("\nTotal Lines = %d\n",lc); printf("\nTotal spaces = %d\n",sc); printf("\nTotal Tabs = %d\n",tc); printf("\nTotal Characters = %d\n",chc); return 0;

}

Output:



# Assignment – 2

Objective: Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.

Code:

%{ #include<stdio.h>int c=0;%}

%%

[a-zA-Z\_][a-zA-Z0-9]\* {c++; printf("%d",yytext);}

. ;

%%

int main(){ yylex();

printf("\nTotal number of valid Identifier = %d \n",c);

}

Output:

$ lex 2-valid\_Identifier.l

$ cc lex.yy.c -lfl

$ ./a.out

count ad\_samsung w12

valid Identifier = count valid Identifier = ad\_samsung valid Identifier = w12 123 3\_er gh\_

valid Identifier = gh\_

//Press <CTRL>d to stop giving input. Total number of valid Identifier = 4

$

# Assignment – 3

Objective: Design a LEX Code to identify and print integer and float value in given Input pattern.

Code:

%{

#include<stdio.h>

%}

%%

[0-9]+"."[0-9] {printf("\nDecimal Number\n");}

[0-9]+ {printf("\nInteger Number\n");}

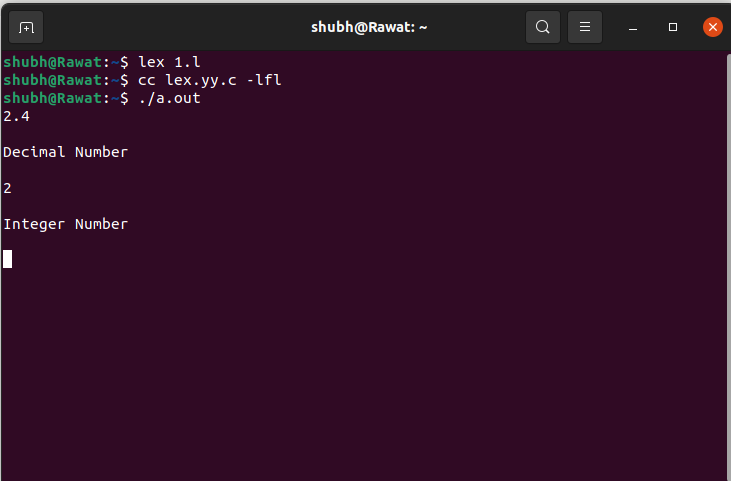
%%

int yywrap(void){} int main()

{

yylex();return 0;}

Output:



# Assignment – 4

Objective: Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPERATORS, KEYWORDS, IDENTIFERS) the following C-fragment:

int p=1, d=0, r=4; float m=0.0, n=200.0; while (p <= 3)

{ if(d==0)

{m= m+n\*r+4.5; d++;} else

{ r++; m=m+r+1000.0; }

p++; }

Code:

%{

int n=0;

%}

%%

"while"|"if"|"else" {n++; printf("\t Keywords : %s",yytext);} "int"|"float" {n++; printf("\t Keywords : %s",yytext);}

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

"<="|"=="|"="|"++"|"+"|"-"|"\*"|"/" {n++; printf("\t Operator : %s",yytext);}

"("|")"|"{"|"}"|","|";" {n++; printf("\t Seperator : %s",yytext);} [0-9]\*"."[0-9]+ {n++; printf("\t Float %s",yytext);}

[0-9]+ {n++; printf("\t Integer : %s",yytext);}

,;

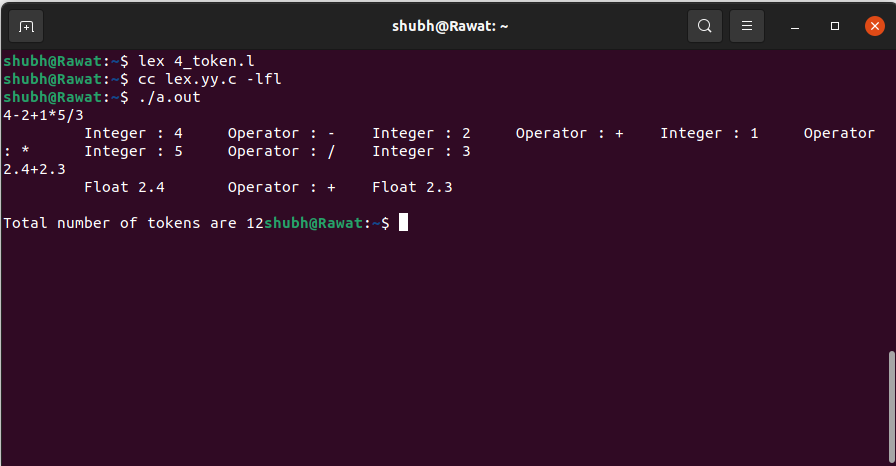
%%

int main() { yylex();

printf("\nTotal number of tokens are %d",n);

}

Output:



# Assignment – 5

Objective: Design a LEX Code to count and print the number of total characters, words, white spaces in given ‘Input.txt’ file.

Code:

%{

int n,w,c;

%}

%%

\n n++;

[^ \n\t]+ {w++; c=c+yyleng;}

. c++;

%%

int main()

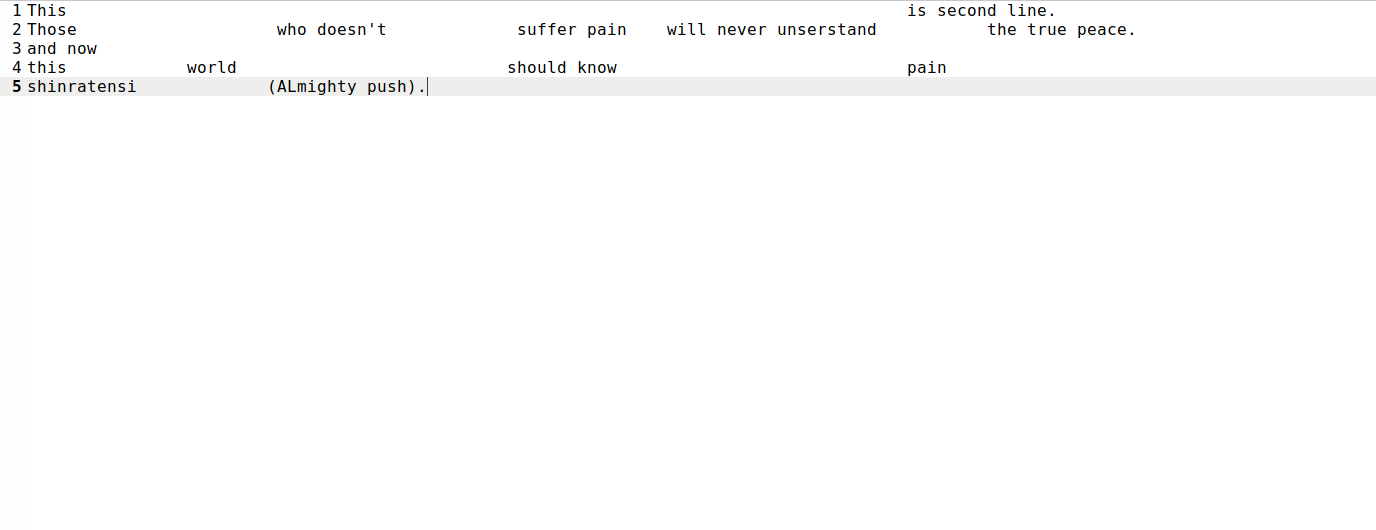
{

extern FILE \*yyin; yyin = fopen("file","r"); yylex();

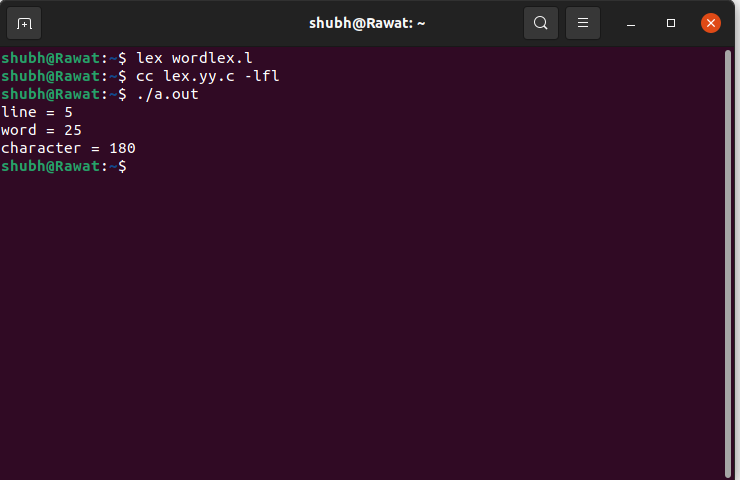
printf("line = %d\nword = %d\ncharacter = %d\n",n,w,c);

}

Input:



Output:



# Assignment – 6

Objective: Design a LEX Code to replace white spaces of ‘Input.txt’ file by a single blank character into ‘Output.txt’ file.

Code:

%{

%}

%%

[\t\n]+ fprintf(yyout," ");

. fprintf(yyout,"%s",yytext);

%%

int main()

{

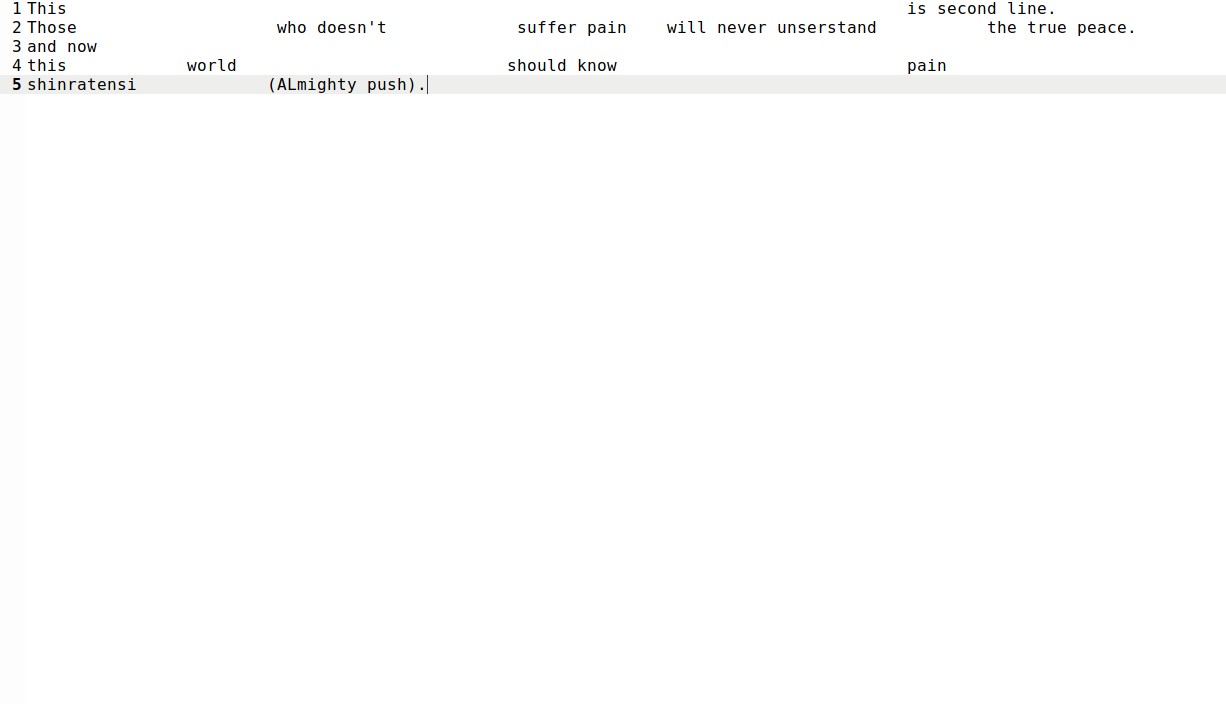
extern FILE \*yyin, \*yyout;

yyin = fopen("file","r"); //r for read.

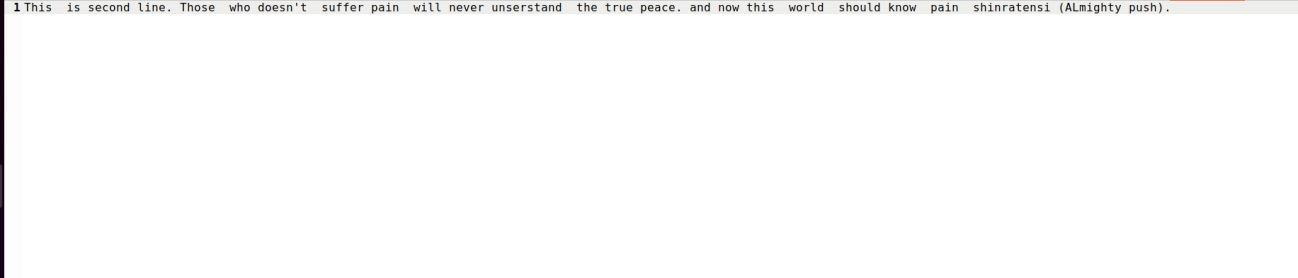
yyout = fopen("output","w"); //w for write. yylex();

}

Input:



Output:



# Assignment – 7

Objective: Design a LEX Code to remove the comments from any C-Program given at run-time and store into ‘out.c’ file.

Code:

%{

#include<stdio.h>

%}

%%

\/\/(.\*) {};

\/\\*(.\*\n)\*.\*\\*\/ {};

%%

int yywrap()

{

return 1;

}

int main()

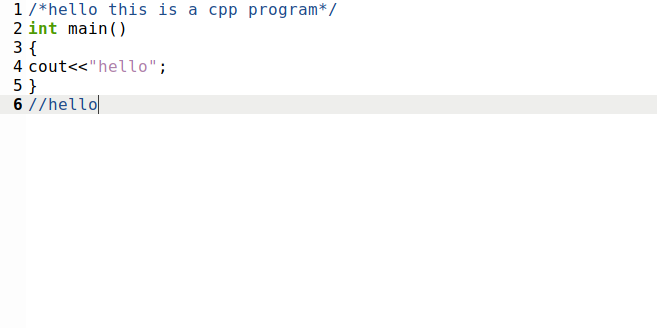
{

yyin = fopen("input8.c","r"); yyout = fopen("output8.txt","w"); yylex();

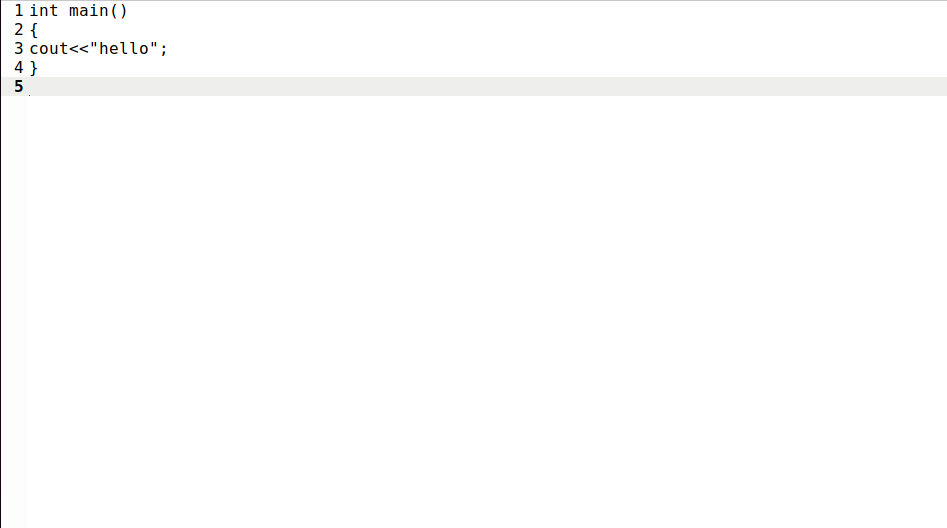
return 0;

}

Input:



Output:



# Assignment – 8

Objective: Design a LEX Code to extract all html tags in the given HTML file at run time and store into Text file given at run time.

Code:

%{

#include<stdio.h>

%}

%%

\<[^>]\*\> fprintf(yyout,"%s\n",yytext);

.|\n;

%%

int yywrap()

{

return 1;

}

int main()

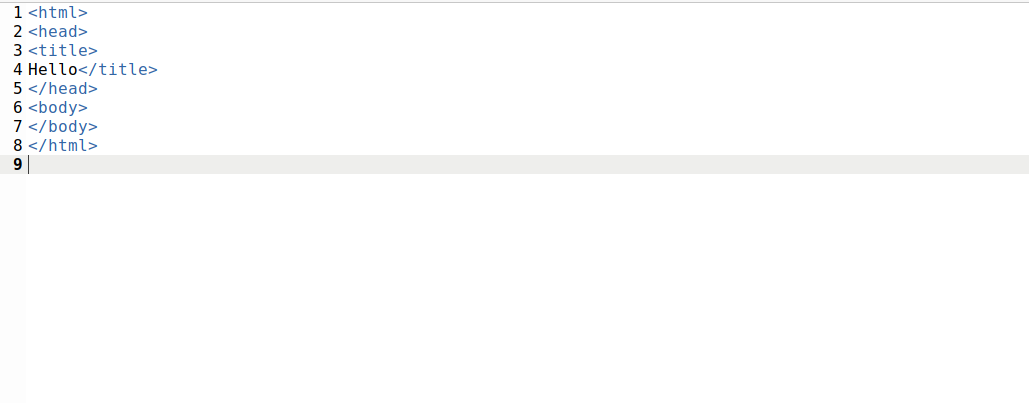
{

yyin = fopen("input7.html","r"); yyout = fopen("output7.txt","w"); yylex();

return 0;

}

Input:



Output:

