**OBJECTIVE 2:** Design a LEX Code to count the number of lines, space, tab-meta character and rest of characters in a given Input pattern.

**PROGRAM:**

%{

#include <stdio.h>

int line\_count = 0, space\_count = 0, tab\_count = 0, other\_count = 0;

%}

%%

\n { line\_count++; }

" " { space\_count++; }

\t { tab\_count++; }

. { other\_count++; }

%%

int main() {

printf("Enter input :\n");

yylex();

printf("Lines: %d\n", line\_count);

printf("Spaces: %d\n", space\_count);

printf("Tabs: %d\n", tab\_count);

printf("Other Characters: %d\n", other\_count);

return 0;

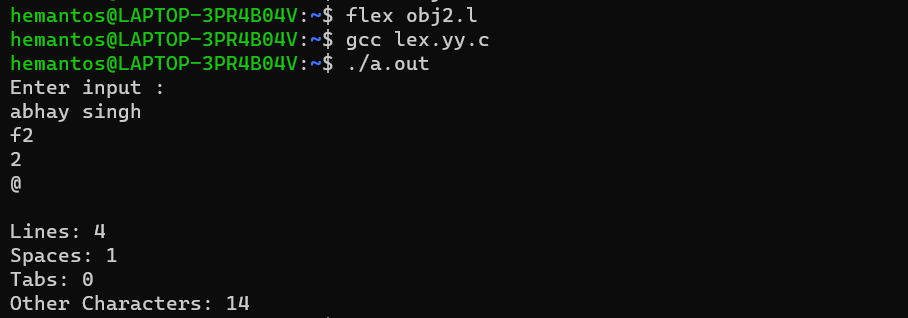
}

int yywrap() {

return 1;

}

**OUTPUT:**



**OBJECTIVE 3:** Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.

**PROGRAM:**

%{

#include <stdio.h>

#include <ctype.h>

%}

%%

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

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

%%

int main() {

printf("Enter input:\n");

yylex();

return 0;

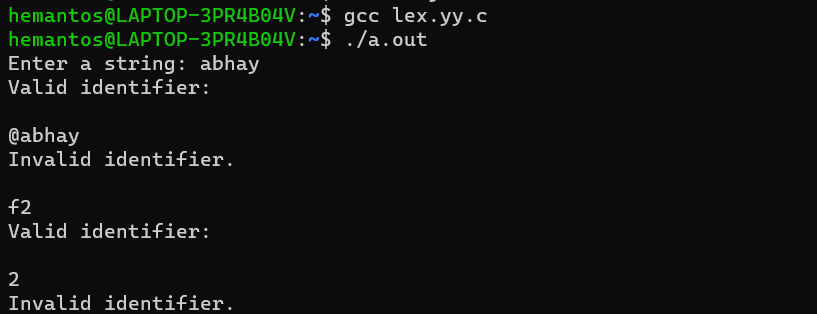
}

int yywrap() {

return 1;

}

**OUTPUT:**



**OBJECTIVE 4:** Design a LEX Code to identify and print integer and float value in given Input pattern.

**PROGRAM:**

%{

#include <stdio.h>

%}

%%

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

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

. { printf("Not a number: %s\n", yytext); }

%%

int main() {

printf("Enter input:\n");

yylex();

return 0;

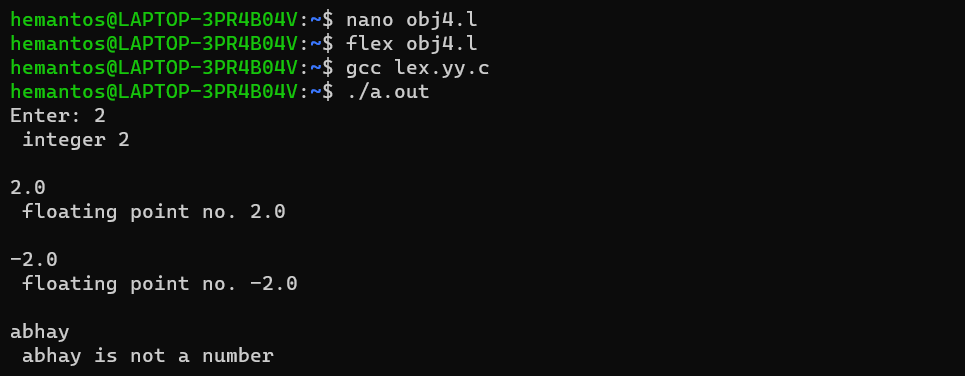
}

int yywrap() {

return 1;

}

**OUTPUT:**



**OBJECTIVE 5:** Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPERATORS, KEYWORDS, and IDENTIFIERS) in the given input.

**PROGRAM:**

%{

#include<stdio.h>

%}

%%

auto|break|case|count|continue|default|do|double|enum|extern|float|for|goto|if|long|register|return|short|signed|sizeof|static|struct|switch|typeof|union|unsigned|else|char|void|volatile|while| int {printf("keyword:\n"); }

[{|}|(|)|;|,] {printf("sperater:\n"); }

[+|\-|\*|/|=|%] {printf("operator:\n"); }

^[a-zA-Z\_][a-zA-Z0-9\_]\* { printf("identifer:\n"); }

.|\n ;

%%

int yywrap(){

return 1;

}

int main(){

printf("Enter the input to check:\n");

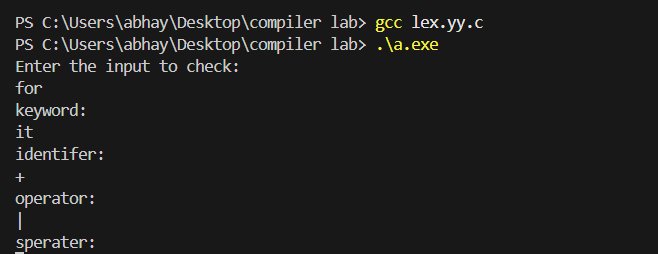
yylex();

printf("------Exit-----\n");

return 0;

}

**OUTPUT:**



**OBJECTIVE 6:** Design a Lex code to count and print the number of total characters, words and whitespaces in given “Input.txt” file.

**PROGRAM:**

%{

#include<stdio.h>

int tword=0, tspace=0 , tchar=0;

%}

%%

[a-zA-Z0-9]+ { tword++; tchar+=yyleng; }

[ \t\n] { tspace++; tchar++; } // Fixed tchar++

%%

int yywrap(){

return 1;

}

int main(){

extern FILE \*yyin;

yyin = fopen("file1.txt", "r");

if (yyin == NULL) {

printf("Error opening file!\n");

return 1;

}

yylex();

fclose(yyin); // Always close files

printf("Total words: %d\n", tword);

printf("Total spaces: %d\n", tspace);

printf("Total characters: %d\n", tchar);

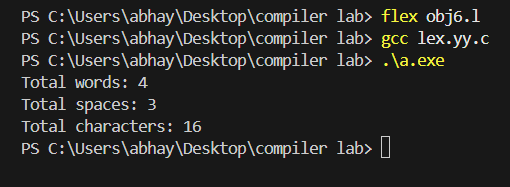
return 0

}

**OUTPUT:**

A black screen with white text

AI-generated content may be incorrect.



**OBJECTIVE 7:** Design a Lex code to replace whitespaces of “Input.txt” file by a single blank character into “Output.txt” file.

**PROGRAM:**

%{

    #include<stdio.h>

%}

%%

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

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

%%

int yywrap()

{

    return 1;

}

int main()

{

    extern  FILE \*yyin,\*yyout;

    yyin=fopen("input.txt","r");

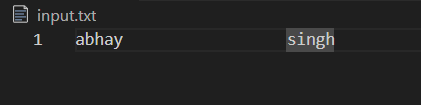
    yyout=fopen("output.txt","w");

    yylex();

    return 0;

}

**OUTPUT:**



A screen shot of a computer screen

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

**OBJECTIVE 8:** Design a LEX Code to remove the comments from any C Program given at runtime and store into ‘out.c’ file.

**PROGRAM:**

**%{**

**#include<stdio.h>**

**%}**

**%%**

**"//".\*\n {fprintf(yyout," ");}**

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

**%%**

**int yywrap()**

**{**

**return 1;**

**}**

**int main()**

**{**

**extern  FILE \*yyin,\*yyout;**

**yyin=fopen("input.c","r");**

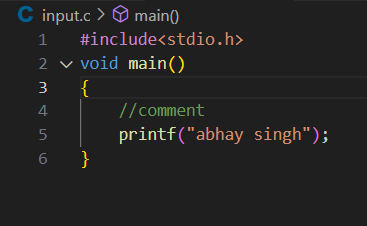
**yyout=fopen("output.c","w");**

**yylex();**

**return 0;**

**}**

**OUTPUT:**



A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**OBJECTIVE 9:** 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.

**PROGRAM:**

**%{**

**#include<stdio.h>**

**%}**

**%%**

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

**.|\n ;**

**%%**

**int yywrap()**

**{**

**return 1;**

**}**

**int main()**

**{**

**extern  FILE \*yyin,\*yyout;**

**yyin=fopen("input.html","r");**

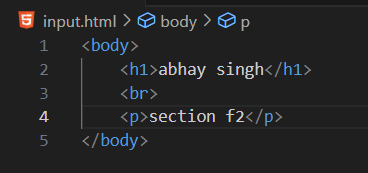
**yyout=fopen("output.html","w");**

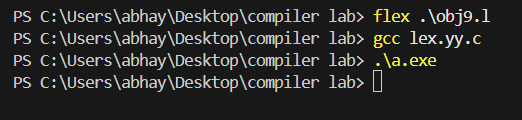
**yylex();**

**return 0;**

**}**

**OUTPUT:**







**OBJECTIVE 10:** **Design a LEX Code to recognize and print the following tokens: a) string b) keywords c) constants d) identifiers e) literals**

**PROGRAM:**

%{

#include<stdio.h>

%}

%%

^["].\*["] {printf("string");}

^['].\*['] {printf("literals");}

auto|break|case|count|continue|default|do|double|enum|extern|float|for|goto|if|long|register|return|short|signed|sizeof|static|struct|switch|typeof|union|unsigned|else|char|void|volatile|while|int {printf("keywords");}

[0-9]+|[0-9]\*[.][0-9]+ {printf("constants");}

^[a-zA-Z\_][a-zA-Z0-9\_]\* {printf("identifiers");}

.\* {printf("not valid token");}

%%

int yywrap()

{

    return 0;

}

int main()

{

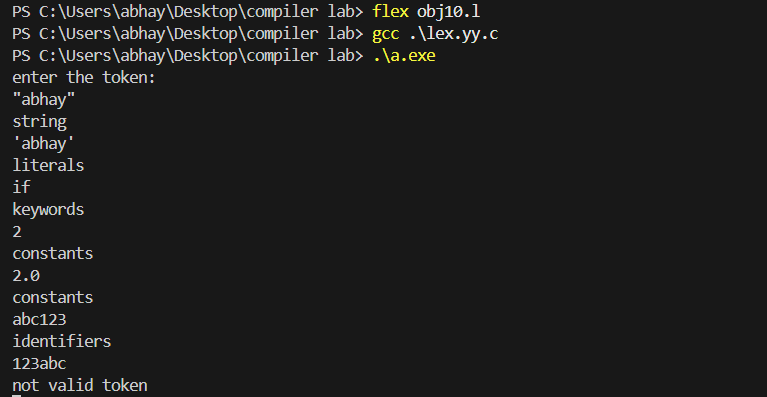
printf("enter the token:\n");

yylex();

return 0;

}

**OUTPUT:**

****

**OBJECTIVE 11:** Design a LEX Code to take check whether the given number is even or odd.

**PROGRAM:**

%{

#include<stdio.h>

%}

%%

[0-9]\*[02468]+ {printf("even number");}

[0-9]\*[13579]+ {printf("odd number");}

.\* {printf("not a valid number");}

%%

int yywrap()

{

return 1;

}

int main()

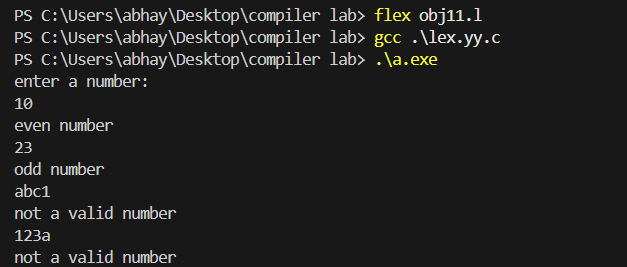
{

printf("enter a number:\n");

yylex();

return 0;

}

**OUTPUT:**

**OBJECTIVE 12:** Design a LEX Code to count number of vowels and consonants in a given pattern.

**PROGRAM:**

%{

    #include<stdio.h>

    int v=0,c=0;

%}

%%

[aeiouAEIOU] {v++;}

[a-zA-Z] {c++;}

.|\n ;

%%

int yywrap()

{

    return 1;

}

int main()

{

    printf("enter a string:\n");

    yylex();

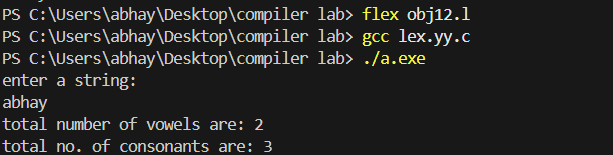
    printf("total number of vowels are: %d\n",v);

    printf("total no. of consonants are: %d\n",c);

    return 0;

}

**OUTPUT:**



**OBJECTIVE 13:** Design a LEX Code to check for a valid E-mail Id.

**PROGRAM:**

%{

    #include<stdio.h>

%}

%%

[a-zA-Z0-9\_]([a-zA-Z0-9\_-]\*[.]?[a-zA-Z0-9\_]+)\*(@gmail.com) {printf("valid email");}

.\* {printf("not a valid email");}

%%

int yywrap()

{

    return 1;

}

int main()

{

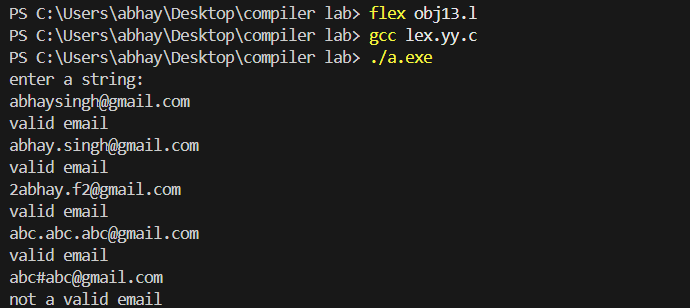
    printf("enter a string:\n");

    yylex();

    return 0;

}

**OUTPUT:**

****

**OBJECTIVE 14:** Design a DFA in LEX Code which accepts all possible set of string containing even number of ‘a’ over input alphabet Σ = {a, b}.

**PROGRAM:**

%{

#include<stdio.h>

%}

%s A DEAD

%%

<INITIAL>a BEGIN A;

<INITIAL>b BEGIN INITIAL;

<INITIAL>\n printf("accepted\n"); BEGIN INITIAL;

<INITIAL>[^ab\n] BEGIN DEAD;

<A>a BEGIN INITIAL;

<A>b BEGIN A;

<A>\n printf("not accepted\n"); BEGIN INITIAL;

<A>[^ab\n] BEGIN DEAD;

<DEAD>\n printf("invalid input\n"); BEGIN INITIAL;

<DEAD>[^\n] BEGIN DEAD;

%%

int yywrap()

{

return 1;

}

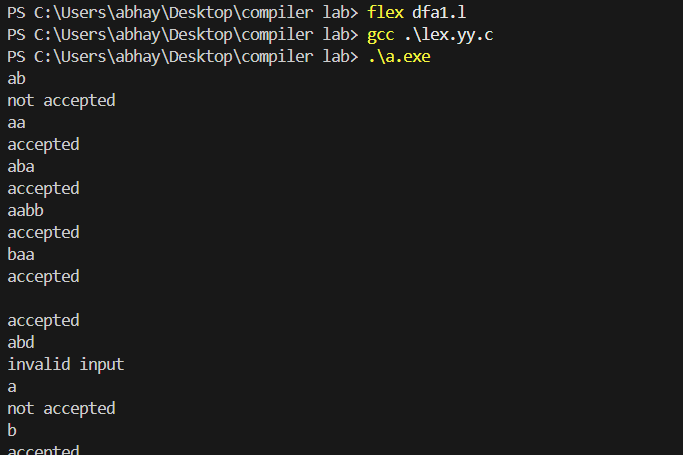
int main(){

yylex();

return 0;

}

**OUTPUT:**

****