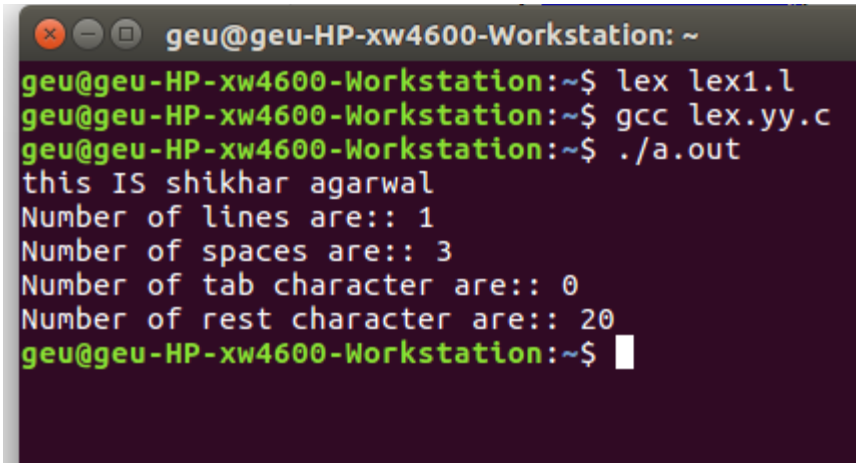
NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 1:
Design a LEX Code to count the number of lines, space, tab-meta character, and rest of characters in each Input pattern.

INPUT :

```
%{
#include<stdio.h>
#include<stdlib.h>
int count=0,space=0,tcount=0,rcount=0;
%}

%%

\n count++;

" " space++;

\t tcount++;

[^\t" "\n] rcount++;
. ;
%%

int yywrap()
{return 1;}

int main(void)
{


yylex();
printf("Number of lines are:: %d\n",count);
printf("Number of spaces are:: %d\n",space);
printf("Number of tab character are:: %d\n",tcount);
printf("Number of rest character are:: %d\n",rcount);
return 0;
}
```

OUTPUT :


geu@geu-HP-xw4600-Workstation: ~
geu@geu-HP-xw4600-Workstation:~$ lex lex1.l
geu@geu-HP-xw4600-Workstation:~$ gcc lex.yy.c
geu@geu-HP-xw4600-Workstation:~$ ./a.out
this IS shikhar agarwal
Number of lines are:: 1
Number of spaces are:: 3
Number of tab character are:: 0
Number of rest character are:: 20
geu@geu-HP-xw4600-Workstation:~$

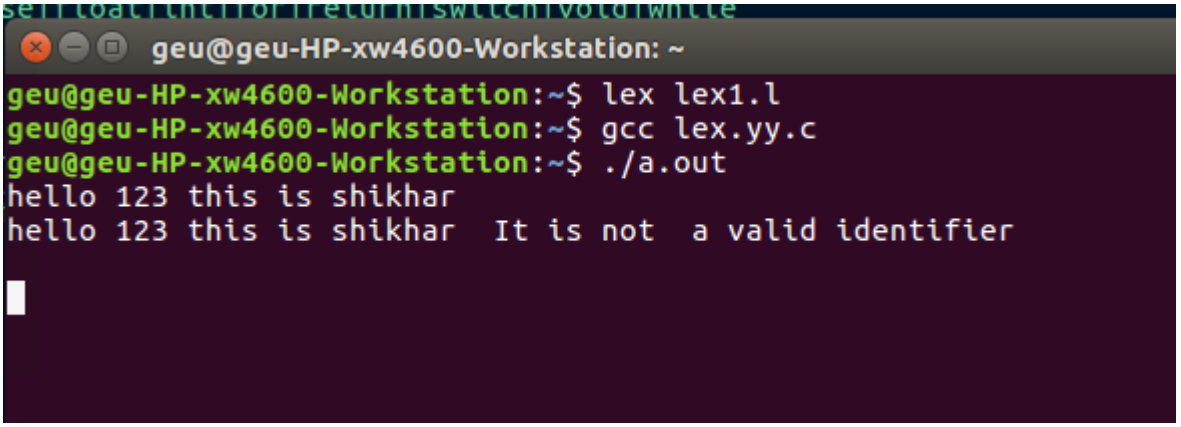
NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 2:
Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.

INPUT :

```
%{
   #include<stdio.h>
   #include<string.h>


%}

keyw  auto|break|case|char|constant|default|else|float|int|for|return|switch|void|while
idf   [0-zA-Z]+[a-zA-Z0-9]*
nidf .*

%%
{keyw} {printf("%s It is a keyword\n" ,yytext);}
{idf}  {printf("%s It is a valid identifier\n" ,yytext);}
{nidf}    {printf("%s It is not  a valid identifier\n" ,yytext);}
%%

int yywrap(){
   return 1;
}

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

OUTPUT :


geu@geu-HP-xw4600-Workstation: ~
geu@geu-HP-xw4600-Workstation:~$ lex lex1.l
geu@geu-HP-xw4600-Workstation:~$ gcc lex.yy.c
geu@geu-HP-xw4600-Workstation:~$ ./a.out
hello 123 this is shikhar
hello 123 this is shikhar It is not a valid identifier

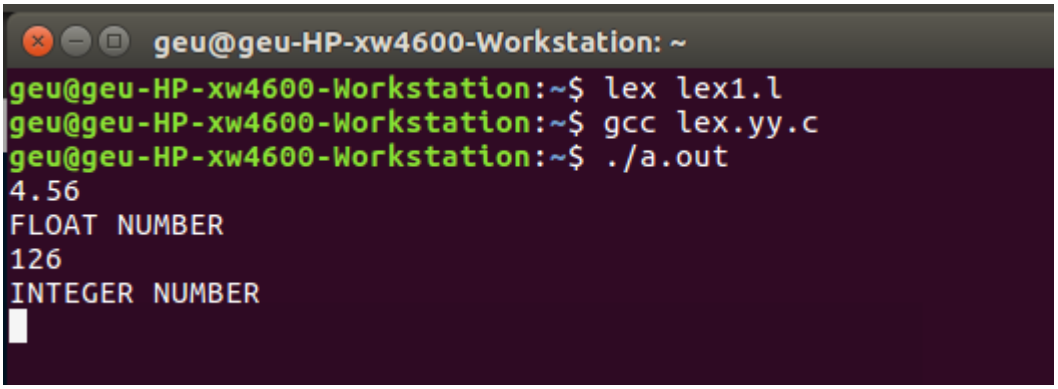
NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 3:
Design a LEX Code to identify and print integer and float value in given Input pattern.

INPUT :

```
%{
   #include<stdio.h>

%}


%%
[-+]*[0-9]+[.][0-9]+ {printf("FLOAT NUMBER" );}
[-+]*[0-9]+ {printf("INTEGER NUMBER" );}

%%

int yywrap(){
   return 1;
}

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

OUTPUT :


geu@geu-HP-xw4600-Workstation: ~
geu@geu-HP-xw4600-Workstation:~$ lex lex1.l
geu@geu-HP-xw4600-Workstation:~$ gcc lex.yy.c
geu@geu-HP-xw4600-Workstation:~$ ./a.out
4.56
FLOAT NUMBER
126
INTEGER NUMBER

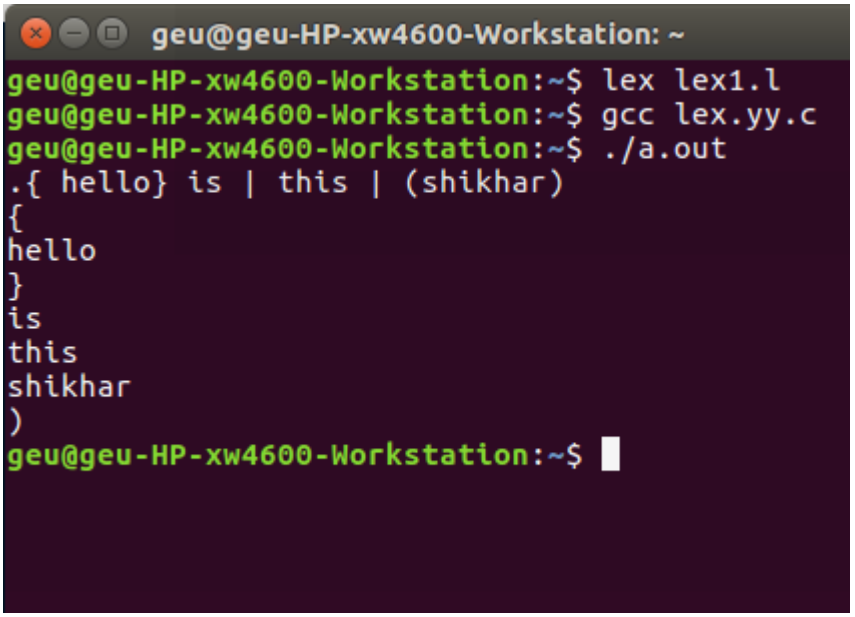
NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 4:
Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPARATORS, KEYWORDS, IDENTIFIERS) in the C-fragment:

INPUT :

```
%{
#include <stdio.h>
#include <stdlib.h>

%}

%%
"while"|"if"|"else"|"int"|"float" {
  printf("%s\n", yytext);
}
[a-zA-Z_][a-zA-Z0-9_]* {
  printf("%s\n", yytext);
}
"<="|"=="|"="|"++"|"-"|"*"|"+""("|")"|"," {
  printf("%s\n", yytext);
}

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

.|\n {
  // Ignore whitespace and newlines
}

%%

int yywrap(){
        return 1;
}


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

OUTPUT :


geu@geu-HP-xw4600-Workstation: ~
geu@geu-HP-xw4600-Workstation:~$ lex lex1.l
geu@geu-HP-xw4600-Workstation:~$ gcc lex.yy.c
geu@geu-HP-xw4600-Workstation:~$ ./a.out
.{ hello} is | this | (shikhar)
{
hello
}
is
this
shikhar
)
geu@geu-HP-xw4600-Workstation:~$

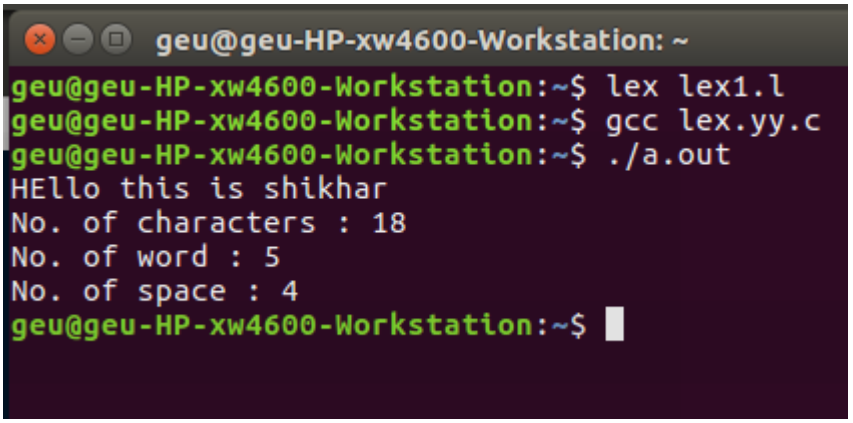
NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 5:
Design a LEX Code to count and print the number of total characters, words, white spaces in the given ‘Input.txt’ file.

INPUT :

```
%{
  #include<stdio.h>
  int tchar=0,tword=0,tspace=0;
%}

%%
" " {tspace++;tword++;}
[\t\n] {tword++;}
[^\n\t] {tchar++;}
%%

int yywrap(){
   return 1;
}

int main(){
   yyin=fopen("input.txt","r");
   yylex();
   printf("No. of characters : %d\n" ,tchar);
   printf("No. of word : %d\n",tword);
   printf("No. of space : %d\n",tspace);
   return 0;
}
```

OUTPUT :


geu@geu-HP-xw4600-Workstation: ~
geu@geu-HP-xw4600-Workstation:~$ lex lex1.l
geu@geu-HP-xw4600-Workstation:~$ gcc lex.yy.c
geu@geu-HP-xw4600-Workstation:~$ ./a.out
HEllo this is shikhar
No. of characters : 18
No. of word : 5
No. of space : 4
geu@geu-HP-xw4600-Workstation:~$

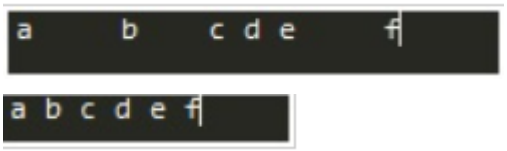
NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 6:
Design a LEX Code to replace white spaces of ‘Input.txt’ file by a single blank character into ‘Output.txt’ file.

INPUT :

```
%{
#include<stdio.h>
%}

%%

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

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


int yywrap()
{
return 1;
}

int main(void)
{
yyin=fopen("input.txt","r");
yyout=fopen("output.txt","w");

yylex();
return 0;
}
```

OUTPUT :



NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

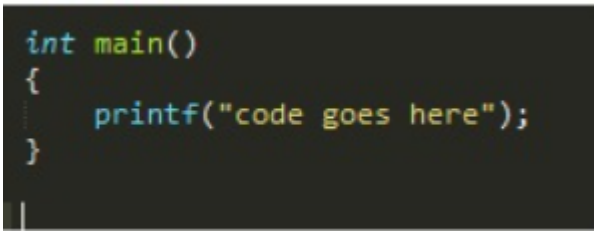
Problem Statement 7:
Design a LEX Code to remove the comments from any C-Program given at run-time and store into ‘out.c’ file.

INPUT :

```
%{
#include<stdio.h>
%}


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

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


int yywrap()
{
return 1;
}


int main()
{
yyin=fopen("input5.c","r");
yyout=fopen("out.c","w");

yylex();
return 0;
}
```

OUTPUT :


//Single line comment
/*multi
line
comment*/
int main()
{
printf("code goes here");
}


int main()
{
printf("code goes here");
}

NAME : SHIKHAR AGARWAL
UNIV. ROLL NO. : 2019096
SECTION : G

Problem Statement 8:
Design a LEX Code to extract all html tags in the given HTML file at run time and store into a Text file given at run time.

INPUT :

```
%{
#include<stdio.h>
%}

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

int yywrap()
{
return 1;
}

int main()
{
yyin=fopen("input5.html","r");
yyout=fopen("output5.txt","w");
yylex();
return 0;
}
```

OUTPUT :

```
<HTML>
<HEAD>
    <TITLE>Page</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

```
<HTML>

<HEAD>

<TITLE>
</TITLE>

</HEAD>

<BODY>

</BODY>

</HTML>
```