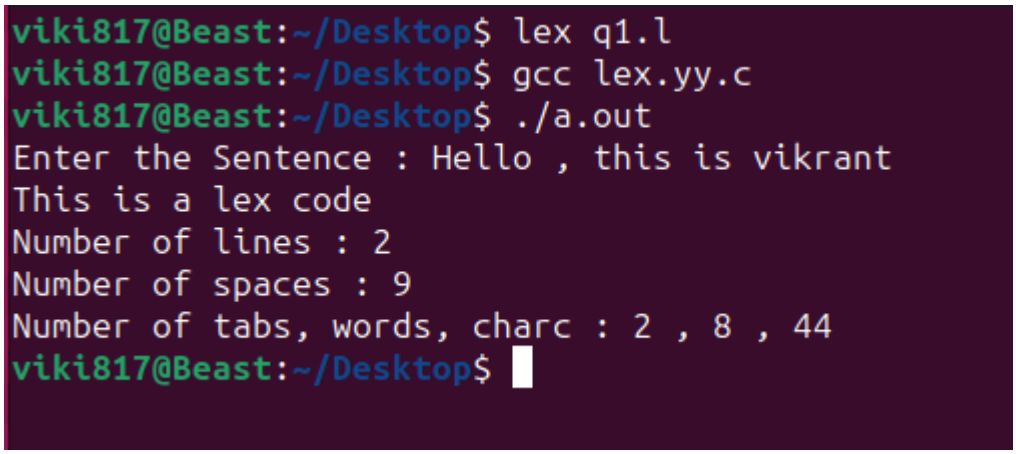
Q1. Design a LEX Code to count the number of lines, space, tab-meta character and rest of characters in a given Input pattern.

```
%{
#include<stdio.h>
int lc=0,sc=0,tc=0,ch=0,wc=0;
%}
%%
[\n] { lc++; ch+=yyleng;}
[ \t] { sc++; ch+=yyleng;}
[^\t] { tc++; ch+=yyleng;}
[^\t\n ]+ { wc++; ch+=yyleng;}
%%
int yywrap(){ return 1;}
int main(){
        printf("Enter the Sentence : ");
        yylex();
        printf("Number of lines : %d\n",lc);
        printf("Number of spaces : %d\n",sc);
        printf("Number of tabs, words, charc : %d , %d , %d\n",tc,wc,ch);

        return 0;
}
```

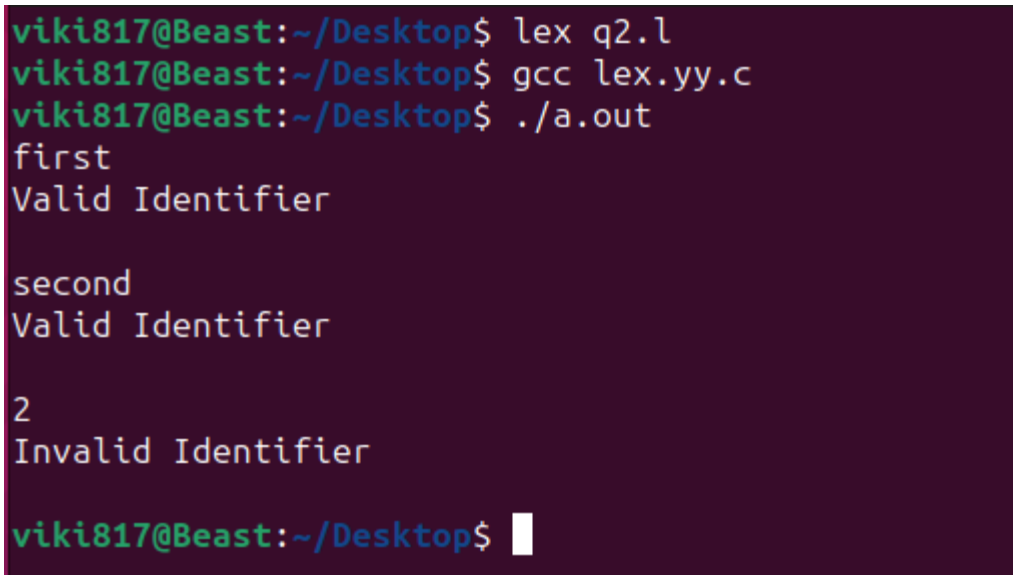
OUTPUT:-


viki817@Beast:~/Desktop$ lex q1.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
Enter the Sentence : Hello , this is vikrant
This is a lex code
Number of lines : 2
Number of spaces : 9
Number of tabs, words, charc : 2 , 8 , 44
viki817@Beast:~/Desktop$

Q2. Design a LEX Code to identify and print valid Identifier of C/C++ in given Input pattern.

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

%%

[a-zA-Z_][a-zA-Z0-9_]*   { printf("Valid Identifier\n"); }
.   { printf("Invalid Identifier\n"); }

%%

int yywrap() {
   return 1;
}

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

OUTPUT:-


viki817@Beast:~/Desktop$ lex q2.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
first
Valid Identifier
second
Valid Identifier
2
Invalid Identifier
viki817@Beast:~/Desktop$

Q3. Design a LEX Code to identify and print integer and float value in given Input pattern.

```
%{
int valid_int=0, valid_float=0;
%}

%%
^[-+]?[0-9]* valid_int++;
^[-+]?[0-9]*[.][0-9]+$ valid_float++;
.;
%%
int yywrap(){return 1;}
int main()
{
yylex();
if(valid_int!=0) printf("Valid Integer number\n");
else if(valid_float!=0) printf("Valid Float number\n");
else printf("Not valid Integer/Float number\n");
return 0;
}
```

OUTPUT:-

```
viki817@Beast:~/Desktop$ lex q3.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
223123

Valid Integer number
viki817@Beast:~/Desktop$ ./a.out
23.45

123123

Valid Integer number
viki817@Beast:~/Desktop$ ./a.out
234.3423

Valid Float number
viki817@Beast:~/Desktop$
```

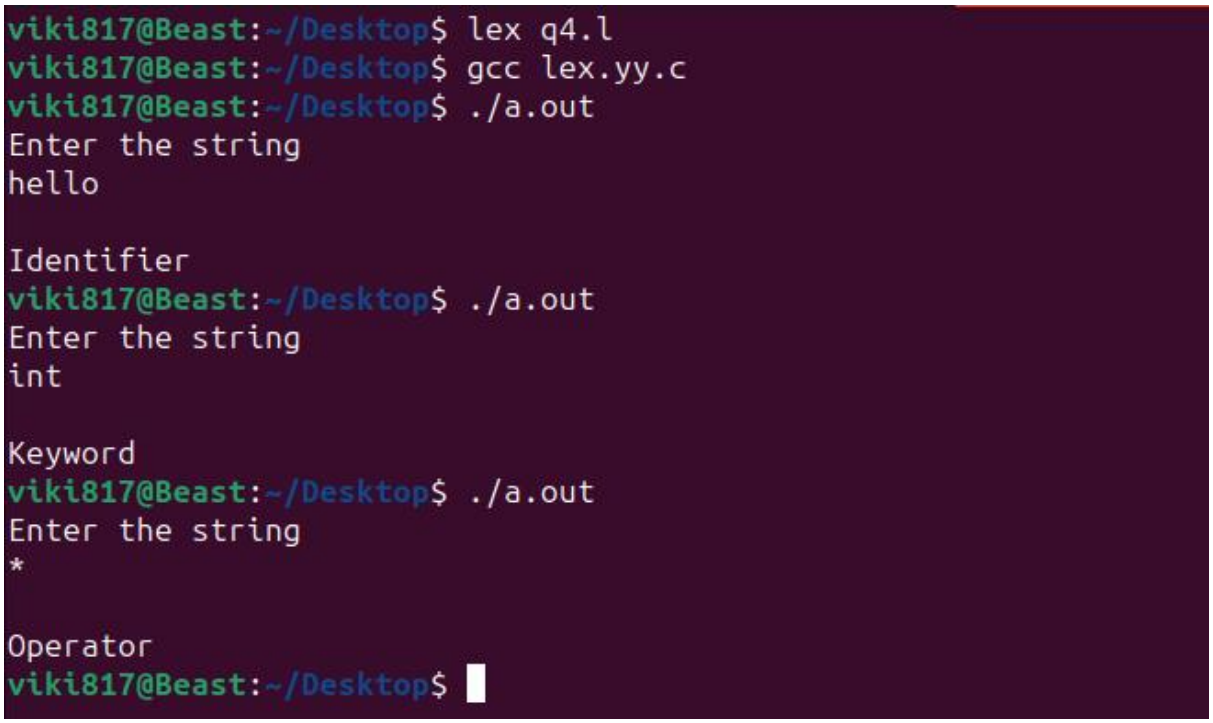
Q4. Design a LEX Code for Tokenizing (Identify and print OPERATORS, SEPERATORS, KEYWORDS, IDENTIFERS) the following C-fragment:

```
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++;
}
```

```
%{
#include<stdio.h>
int op=0,k=0,i=0;
%}

%%
^"int"|"float" {k++;}
^[a-z A-Z _][a-z A-Z 0-9]* {i++;}

[+-/=*%] {op++;}
%%

int yywrap()
```

```
{
        return 1;
}

int main()
{
        printf("Enter the string\n");
        yylex();
        if(op>0)
                printf("Operator\n");
        if(k>0)
                printf("Keyword\n");
        if(i>0)
                printf("Identifier\n");
        return 0;
}
```

OUTPUT:-


viki817@Beast:~/Desktop$ lex q4.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
Enter the string
hello
Identifier
viki817@Beast:~/Desktop$ ./a.out
Enter the string
int
Keyword
viki817@Beast:~/Desktop$ ./a.out
Enter the string
*
Operator
viki817@Beast:~/Desktop$

Q5. Design a LEX Code to count and print the number of total characters, words, white spaces in given 'Input.txt' file.

```
%{
#include<stdio.h>
int n, w, c;
%}
%%
\n { n++; }
[^ \n\t]+ { w++; c = c + yyleng;}
.   {c++;}
%%
int yywrap(void)
{
        return 1;
}

int main()
{
        extern FILE* yyin;
        yyin = fopen("input.txt", "r");
        yylex();
        printf("Line= %d word=%d total char=%d \n", n, w, c);
}
```

INPUT:-

```
1 hello AI this side
2 don't worry about your jobs , i will not take them
3 i will destroy them
```

OUTPUT:-

```
viki817@Beast:~/Desktop$ lex q5.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
Line = 3
Word = 19
Total Characters = 87
viki817@Beast:~/Desktop$
```

Q6. Design a LEX Code to replace white spaces of 'Input.txt' file by a single blank character into 'Output.txt' file.

```
%{
#include <stdio.h>
FILE *output;
%}
%%
[ ]+ fprintf(output, "_");
.|\n fprintf(output, yytext, 0);
%%
int yywrap() {return 1;}
int main() {
        yyin = fopen("input.txt", "r");
        output = fopen("output.txt", "w");
        yylex();
        printf("File written\n");
        return 0;
}
```

INPUT :

```
hello   AI this side
don't worry     about your      jobs , i will not take them
i will destroy   them
```

OUTPUT :

```
viki817@Beast:~/Desktop$ lex q6.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
File written
viki817@Beast:~/Desktop$
```

```
hello AI this side
don't worry about your jobs , i will not take them
i will destroy them
```

Q7. Design a LEX Code to remove the comments from any C-Program given at run-time and store into 'out.c' file.

```
%{
#include <stdio.h>
FILE *output;
%}
%%
\/\/.* ;
\/\*[^\*/]*\*\/ ;
.|\n fprintf(output, yytext, 0);
%%
int yywrap() {return 1;}
int main(int argc, char **argv) {
        yyin = fopen(argv[1], "r");
        output = fopen("out.c", "w");
        yylex();
        printf("File written\n");
        fclose(yyin);
        fclose(output);
        return 0;
}
```

INPUT :

```
viki817@Beast:~/Desktop$ lex q7.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out
//this is comment
int a,b;
File written
viki817@Beast:~/Desktop$
```

OUTPUT :

```
1
2 int a,b;
```

Q8. 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.

```
%{
#include <stdio.h>
FILE *output;
%}
%%
\<[^ \>]+ fprintf(output, "%s>\n", yytext);
.|\n ;
%%
int yywrap() {return 1;}
int main(int argc, char** argv) {
        yyin = fopen(argv[1], "r");
        output = fopen(argv[2], "w");
        yylex();
        printf("Tokenized\n");
        return 0;
}
```

INPUT :

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample HTML Page</title>
</head>
<body>
    <h1>Welcome to my website</h1>
    <p>This is a sample paragraph.</p>
    <a href="https://example.com">Link</a>
</body>
</html>
```

OUTPUT :

```
viki817@Beast:~/Desktop$ lex q8.l
viki817@Beast:~/Desktop$ gcc lex.yy.c
viki817@Beast:~/Desktop$ ./a.out first.html output.txt
Tokenized
viki817@Beast:~/Desktop$
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>
</title>
</head>
<body>
<h1>
</h1>
<p>
</p>
<a href="https://example.com">
</a>
</body>
</html>
```