

### 1.Mobile number validation:

```
%{  
%}  
%%  
[6-9][0-9]{9} {printf("\nMobile Number valid\n");}  
.+ {printf("\nMobile Number Invalid\n");}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter Mobile Number:");  
yylex();  
printf("\n");  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex mobile.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
Enter Mobile Number:7995895745  
Mobile Number valid  
  
799859  
Mobile Number Invalid  
#1|
```

2.digit:

```
%{  
%}  
%%  
[0-9]+| [0-9]*\.[0-9]+ {printf("\ngiven is a digit\n");}  
.+ {printf("\ngiven is not a digit\n");}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter a digit:");  
yylex();  
printf("\n");  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex digit.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
Enter a digit:4  
given is a digit  
d  
given is not a digit  
4.5  
given is a digit  
#2|
```

3.identifier:

```
%{  
%}  
%%  
[a-zA-Z][a-zA-Z0-9]* {printf("\ngiven is a identifier\n");}  
.+ {printf("\ngiven is not a identifier\n");}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter a identifier:");  
yylex();  
printf("\n");  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex identifier.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
Enter a identifier:a  
given is a identifier  
3  
given is not a identifier  
a3  
given is a identifier  
#3|
```

4,vowels and consonents:

```
%{  
int vow_count=0;  
int const_count=0;  
%}  
%%  
[aeiouAEIOU] {vow_count++;}  
[a-zA-Z] {const_count++;}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter a string of vowels and consonents:");  
yylex();  
printf("enter no.of vowels::%d\n",vow_count);  
printf("enter no.of consonents::%d\n",const_count);  
return 0;  
}
```

```
C:  
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin
```

```
C:\Users\palic>flex vowels.l.txt
```

```
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin
```

```
C:\Users\palic>gcc lex.yy.c
```

```
C:\Users\palic>a
```

```
Enter a string of vowels and consonents:duneesh
```

```
enter no.of vowels::3
```

```
enter no.of consonents::4
```

5.email is valid or not:

```
%{  
%}  
%%  
[a-z.0-9_]+@[a-z]+\."com"|"."in" {printf("\nemail is valid\n");}  
.+ {printf("\nemail is not valid\n");}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter a email:");  
yylex();  
printf("\n");  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex mail.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
Enter a email:p.duneeshreddy5421@gmail.com  
email is valid  
p,dun@735@125.com  
email is not valid  
#5
```

6.keyword and identifier:

```
%{  
%}  
%%  
if|else|while|for|return|main|int|char|switch|float|break {printf("\ngiven is a keyword\n");}  
[a-zA-Z][a-zA-Z0-9]* {printf("\ngiven is a identifier\n");}  
.+ {printf("\ngiven is not a identifier and not a keyword\n");}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter the string:");  
yylex();  
printf("\n");  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex keywordidentifier.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
Enter the string:else  
given is a keyword  
bhhg  
given is a identifier  
2bbh  
given is not a identifier and not a keyword  
#6|
```

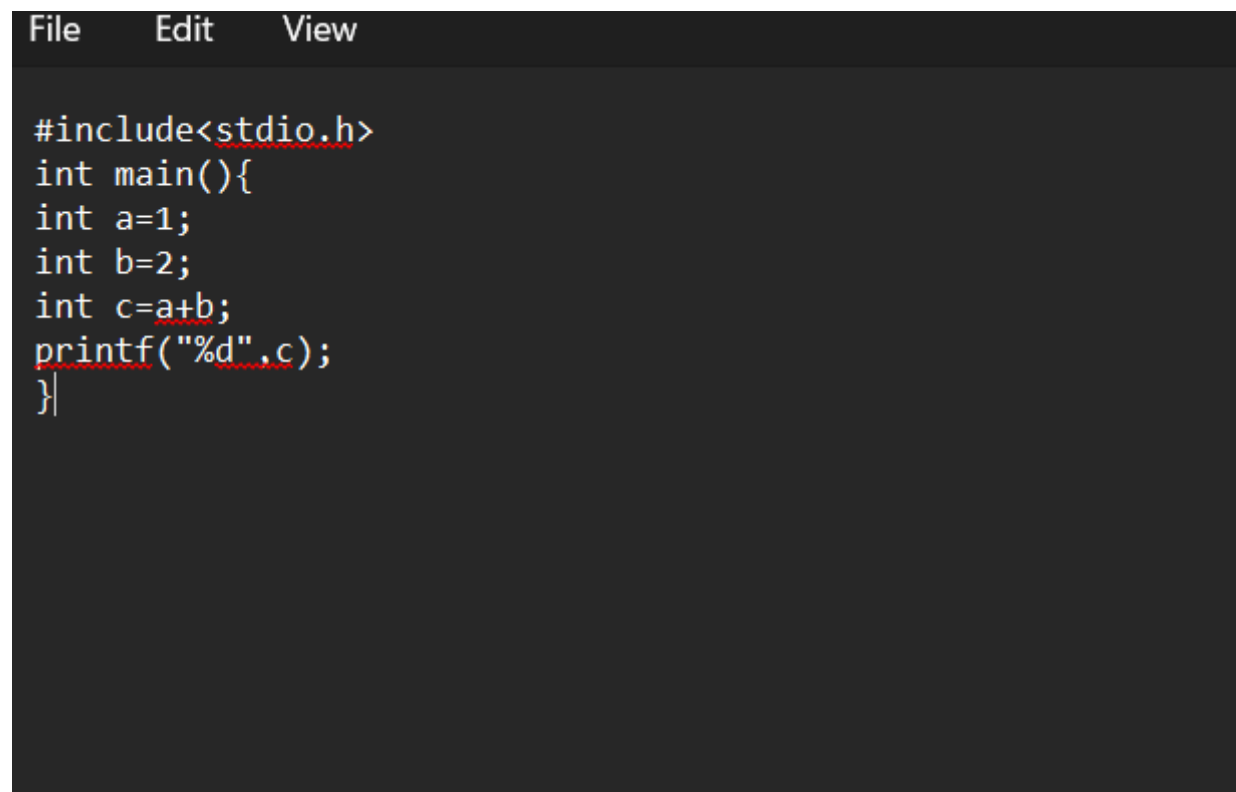
7.capitals:

```
%{  
%}  
%%  
[A-Z]+ {printf("\ngiven string is capitals\n");}  
.+ {printf("\ngiven string not all letters are capitals\n");}  
%%  
int yywrap(void) {}  
int main()  
{  
printf("\nEnter a string:");  
yylex();  
printf("\n");  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex capital.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
Enter a string:Duneesh  
in given string not all are capitals  
DUNEESH  
in given string is capitals  
#7
```

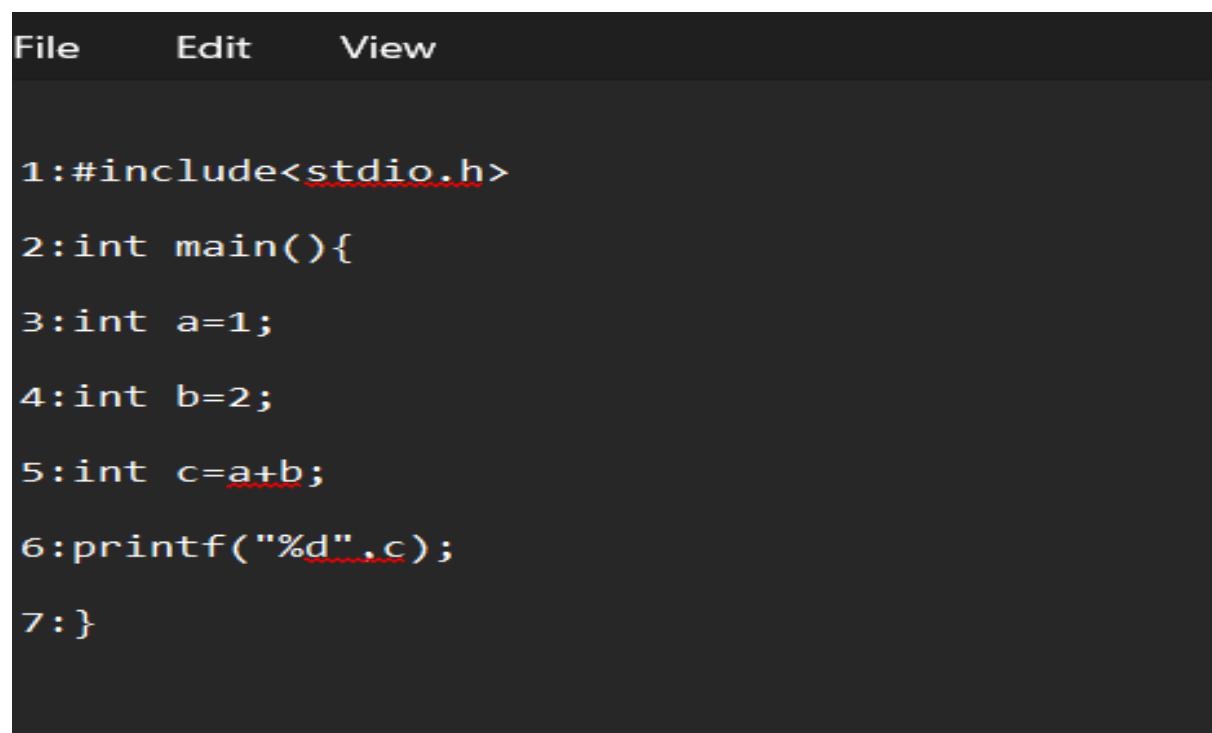
8.add line numbers:

Input:

A screenshot of a code editor with a dark background. The menu bar at the top shows 'File', 'Edit', and 'View'. The code is as follows:

```
#include<stdio.h>
int main(){
int a=1;
int b=2;
int c=a+b;
printf("%d",c);
}
```

Output:

A screenshot of a code editor with a dark background. The menu bar at the top shows 'File', 'Edit', and 'View'. The code is as follows:

```
1:#include<stdio.h>
2:int main(){
3:int a=1;
4:int b=2;
5:int c=a+b;
6:printf("%d",c);
7:}
```



9.longest word:

```
%{  
int counter=0;  
%}  
%%  
[a-zA-Z]+ {  
if(yyleng>counter) {  
counter=yyleng;  
}  
}  
%%  
int yywrap(){}  
int main()  
{  
printf("enter the sentence: ");  
yylex();  
printf("longest word is:%d",counter);  
printf("\n");  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex longestword.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
enter the sentence: saveetha is a marveouls college  
  
longest word is:9  
C:\Users\palic>#9|
```

10.valid url:

```
%{  
%}  
%%  
[http://]+[www.]+[a-z]+".com" {printf("\n valid url\n");}  
.+ {printf("\n in valid url\n");}  
%%  
int yywrap(){}  
int main()  
{  
printf("\n enter the url:");  
yylex();  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin
```

```
C:\Users\palic>flex url.l.txt
```

```
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin
```

```
C:\Users\palic>gcc lex.yy.c
```

```
C:\Users\palic>a
```

```
enter the url:http://www.example.com
```

```
valid url
```

```
fgrwegwerf
```

```
in valid url
```

```
#10
```

11.date of birth:

```
%{  
%}  
%%  
[0-9][0-9]\/[0-1][0-9]\/[1-2][0-9]{3} {printf("valid date of birt");}  
.+ {printf("invalid date of birt");}  
%%  
int yywrap(){}  
int main()  
{  
printf("enter the date of birth: ");  
yylex();  
printf("\n");  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex dob.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
enter the date of birth: 20/06/2005  
valid date of birt  
3/89/9090  
invalid date of birt
```

12:relational operators:

```
%{  
%}  
%%  
[a-z]+ {printf("word: %s\n",yytext);}   
">"|"<"|"<="|">="|"=="|"!=" {printf("relational operator: %s\n",yytext);}   
%%  
int yywrap(){}  
int main()  
{  
printf("enter the combination of words and relops: ");  
yylex();  
printf("\n");  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex words.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
enter the combination of words and relops: duneesh<priya  
word: duneesh  
relational operator:m<  
word: priya  
|
```

13.no of charecters:

```
%{  
#include <stdio.h>  
int i=0, l=0, c=0;  
%}  
%%  
\n { i++; }  
[a-zA-Z0-9]+ { l++; c += yyleng; }  
. { c++; }  
%%  
int yywrap() {}  
int main() {  
    printf("enter the string: ");  
    yylex();  
    printf("no of lines: %d\n", i);  
    printf("no of words: %d\n", l);  
    printf("no of characters: %d\n", c);  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex noofchars.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
enter the string: duneesh  
new player of kabbadi  
^Z  
no of lines: 2  
no of words: 5  
no of characters: 28
```

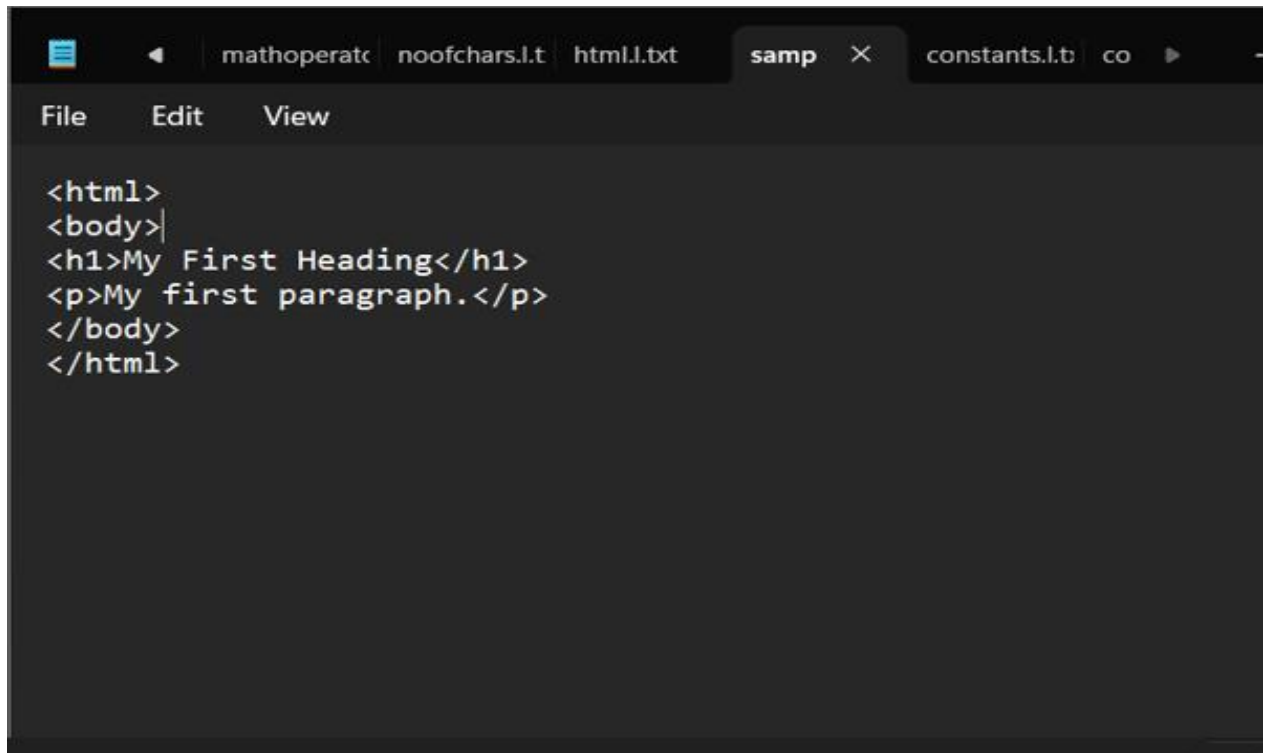
14.comment lines:

```
%{  
#include<stdio.h>  
  
int n=0;  
%}  
%%  
"/"[a-zA-Z0-9 \n\t]+"/" {n++;}  
"//[a-zA-Z0-9 \n\t]+"/" {n++;}  
%%  
  
int yywrap()  
{  
  
int main()  
{  
  
printf("enter the input:");  
  
yylex();  
  
printf("count no of comment lines:%d",n);  
  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex commentlines.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
enter the input:// new code//  
  
/ c program/  
  
# include stdio.h  
# include stdio.h  
/repeat/  
  
//donet  
^Z  
//donet  
count no of comment lines:3  
C:\Users\palic>
```

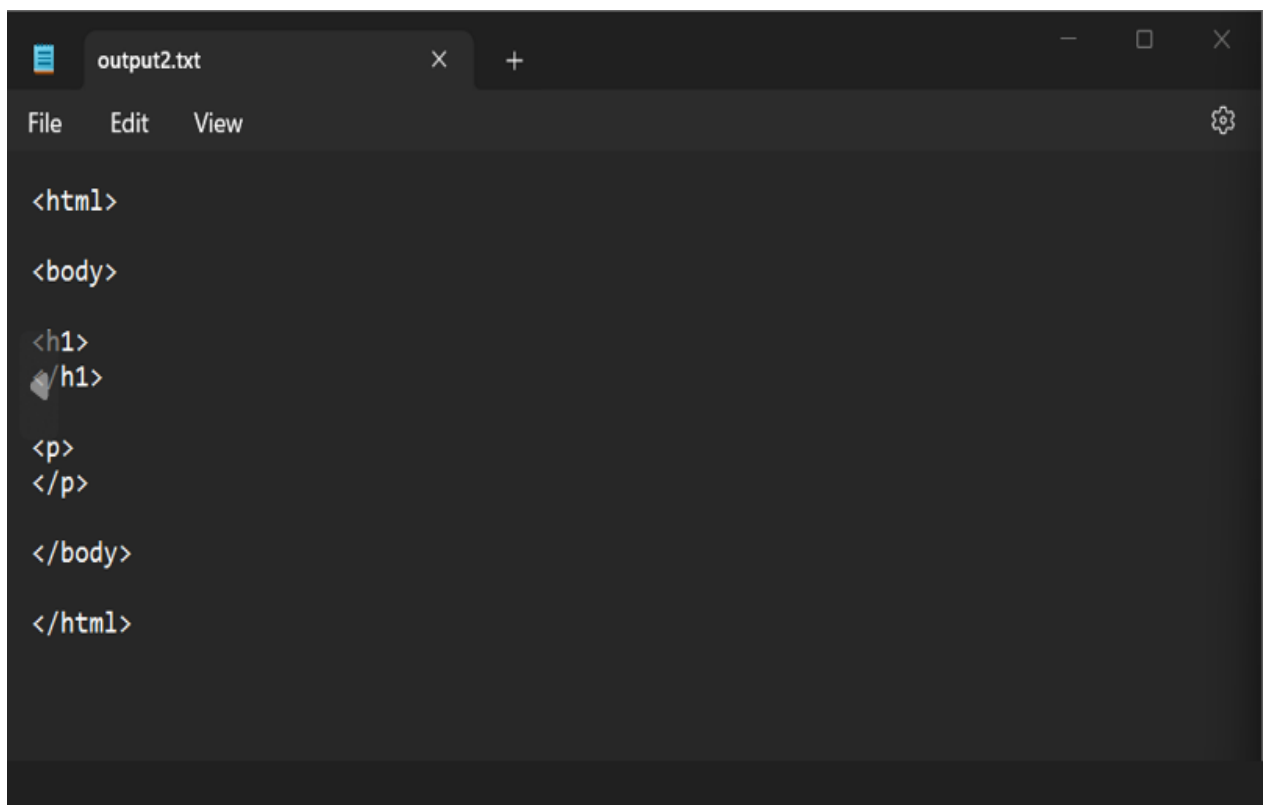
15.html:

Input:

A screenshot of a code editor window with a dark theme. The title bar shows several tabs: 'mathoperatc', 'noofchars.lt', 'html.l.txt', 'samp', 'constants.lt', and 'co'. The 'samp' tab is active. The editor has a menu bar with 'File', 'Edit', and 'View'. The code in the editor is as follows:

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Output:

A screenshot of a code editor window with a dark theme. The title bar shows a single tab 'output2.txt'. The editor has a menu bar with 'File', 'Edit', and 'View', and a settings icon on the right. The code in the editor is as follows:

```
<html>

<body>

<h1>
/h1>

<p>
</p>

</body>

</html>
```

16.constants:

```
%{  
int cons = 0;  
%}  
digit [0-9]  
%%  
{digit}+ "." {digit}+ { cons++; printf("%s is a floating-point constant\n", yytext); }  
{digit}+ { cons++; printf("%s is an integer constant\n", yytext); }  
.\| \n { }  
%%  
int yywrap() {  
}  
int main() {  
printf("Enter the code:");  
yylex();  
printf("Number of Constants: %d\n", cons);  
return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin
```

```
C:\Users\palic>flex constants.l.txt
```

```
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin
```

```
C:\Users\palic>gcc lex.yy.c
```

```
C:\Users\palic>a
```

```
Enter the code:int i =1,2,3,4;
```

```
1 is an integer constant
```

```
2 is an integer constant
```

```
3 is an integer constant
```

```
4 is an integer constant
```

```
^Z
```

```
Number of Constants: 4
```



17.positive or negative numbers:

```
%{  
int positive_no=0,negative_no=0;  
%}  
%%  
[-][0-9]+ {negative_no++;  
    printf("negative number=%s\n",yytext);}  
[0-9]+ {positive_no++;  
    printf("positive number=%s\n",yytext);}  
%%  
int yywrap(){}  
int main()  
{  
    yylex();  
    printf("number of posive integers=%d,"  
        "number of negativenumbers=%d\n",  
        positive_no,negative_no);  
    return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex positive.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
-8 -9 9 0  
negative number=-8  
negative number=-9  
positive number=9  
positive number=0  
  
^Z  
number of posive integers=2,number of negativenumbers=2
```

18.substring:

```
%{  
#include <ctype.h>  
%}  
%%  
[a-z] { printf("%c", toupper(yytext[0])); }  
.|\\n { printf("%s", yytext); }  
%%  
int yywrap()  
{  
}  
int main() {  
    yylex();  
    return 0;  
}
```

(c) Microsoft Corporation. All rights reserved.

C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin

C:\Users\palic>flex substring.l.txt

C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin

C:\Users\palic>gcc lex.yy.c

C:\Users\palic>a

i AM a A student

I AM A A STUDENT

duneesh

DUNEESH

|

19.macros:

```
%{  
int nmacro = 0, nheader = 0; // Initialize counters  
%}  
%%  
"#define" { nmacro++; } // Increment macro counter for "#define"  
"#include" { nheader++; } // Increment header counter for "#include"  
.|\\n { } // Ignore all other characters  
%%  
int yywrap() {  
    return 1;  
}  
int main() {  
    printf("Enter the string:\\n");  
    yylex(); // Start lexical analysis  
    // Corrected printf statement  
    printf("Number of macros defined = %d\\n", nmacro);  
    printf("Number of header files included = %d\\n", nheader);  
    return 0;  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex macros.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
  
C:\Users\palic>a  
Enter the string:  
#define MAX 100  
#include <stdio.h>  
#define MIN 0  
#include <stdlib.h>  
  
^Z  
Number of macros defined = 2  
Number of header files included = 2
```


20.frequency:

```
%{  
#include<stdio.h>  
#include<string.h>  
char word [] = "geeks";  
int count = 0;  
%}  
%%  
[a-zA-Z]+ { if(strcmp(yytext, word)==0) count++; }  
.  
%%  
int yywrap()  
{  
return 1;  
}  
int main()  
{  
extern FILE *yyin, *yyout;  
yyin=fopen("input.txt", "r");  
yylex();  
printf("%d", count);  
}
```

```
C:\Users\palic>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Users\palic>flex frequency.l.txt  
C:\Users\palic>set path=C:\Program Files (x86)\MinGW\bin  
C:\Users\palic>gcc lex.yy.c  
C:\Users\palic>a  
geeks are cool, geeks love coding.  
geeks enjoy solving problems.  
^Z  
3
```

21.math operators:

```
%{  
float op1, op2;  
%}  
%%  
"+" { printf("sum = %f\n", op1 + op2); }  
"-" { printf("diff = %f\n", op1 - op2); }  
"*" { printf("mul = %f\n", op1 * op2); }  
"/" {  
    if (op2 != 0)  
        printf("div = %f\n", op1 / op2);  
    else  
        printf("Error: Division by zero is not allowed.\n");  
}  
. { printf("Invalid operator. Please enter a valid operator.\n"); }  
%%  
int yywrap() {}  
int main() {  
    printf("Enter number 1: ");  
    scanf("%f", &op1);  
    printf("Enter number 2: ");  
    scanf("%f", &op2);  
    printf("Enter the Operator (+, -, *, /): ");  
    yylex();  
    return 0;  
}
```

 C:\Users\Owner\Documents\arithmetic.exe

Enter a string: +,-,\*,/

Character '+' is an operator: PLUS

Character ',' is invalid

Character '-' is an operator: MINUS

Character ',' is invalid

Character '\*' is an operator: MULTIPLY

Character ',' is invalid

Character '/' is an operator: DIVIDE

-----

Process exited after 10.12 seconds with return value 0

Press any key to continue . . .

22.replace:

```
%{
#include<stdio.h>
#include<string.h>
char replace_with [100];
char replace [100];
}%
%%
[a-zA-Z]+ { if(strcmp(yytext, replace)==0)
    fprintf(yyout, "%s", replace_with);
    else
    fprintf(yyout, "%s", yytext);}
. fprintf(yyout, "%s", yytext);
%%
int yywrap()
{
    return 1;
}
int main()
{
    printf("enter replacing string:\n");
    scanf("%s",&replace);
    printf("\nenter replacing with string:\n");
    scanf("%s",&replace_with);
    printf("enter a input string:\n");
    yylex();
}
```