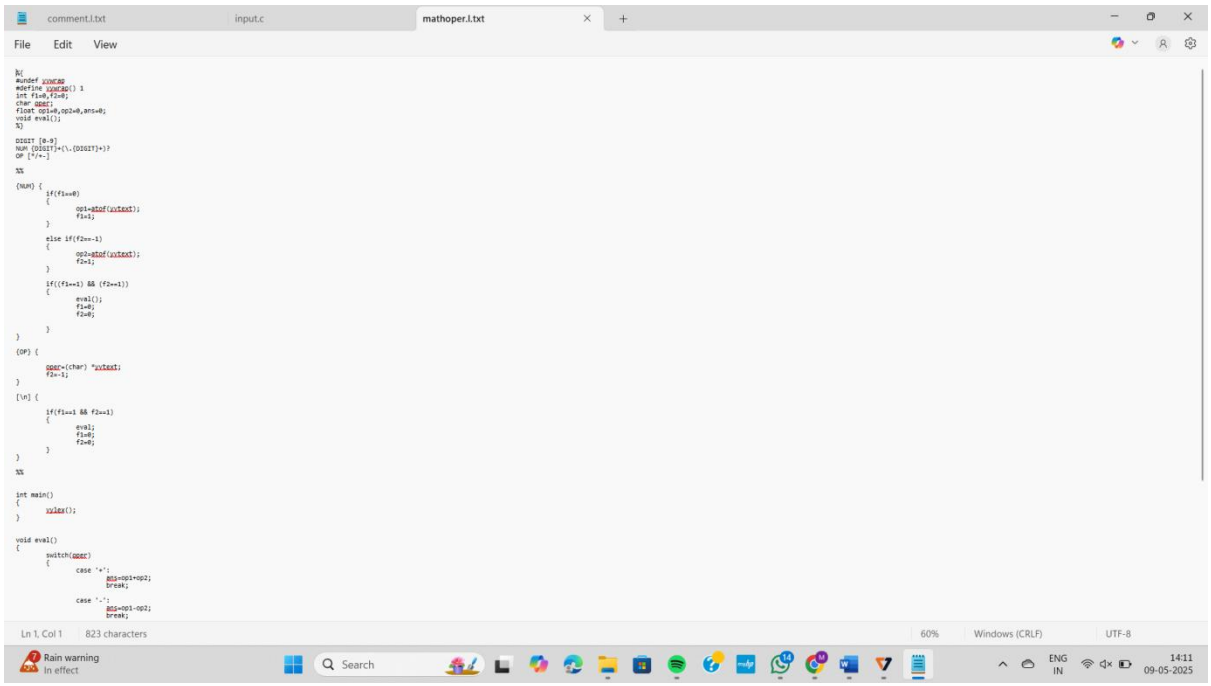
Name: ch.ishwarya
reg no: 192372104
course: csa1409 – compiler design

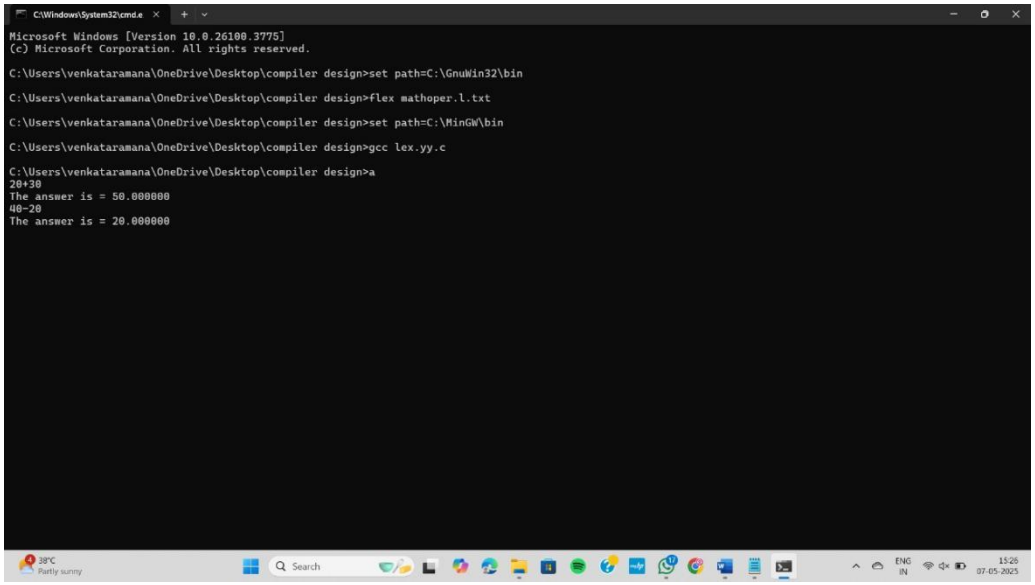
1) Write a LEX program to implement basic mathematical operations.

Program:



```
%{
#undef yywrap
#define yywrap() 1
int f1=0,f2=0;
char oper;
float op1=0,op2=0,ans=0;
void eval();
%}

DIGIT [0-9]
NUM {DIGIT}+(\.{DIGIT}+)?
OP [*/+-]

%%

{NUM} {
        if(f1==0)
        {
                op1=atof(yytext);
                f1=1;
        }
        else if(f2==-1)
        {
                op2=atof(yytext);
                f2=1;
        }
        if((f1==1) && (f2==1))
        {
                eval();
                f1=0;
                f2=0;

        }
}
{OP} {
        oper=(char) *yytext;
        f2=-1;
}
[\n] {
        if(f1==1 && f2==1)
        {
                eval;
                f1=0;
                f2=0;
        }
}
%%

int main()
{
        yylex();
}

void eval()
{
        switch(oper)
        {
                case '+':
                        ans=op1+op2;
                        break;

                case '-':
                        ans=op1-op2;
                        break;
```

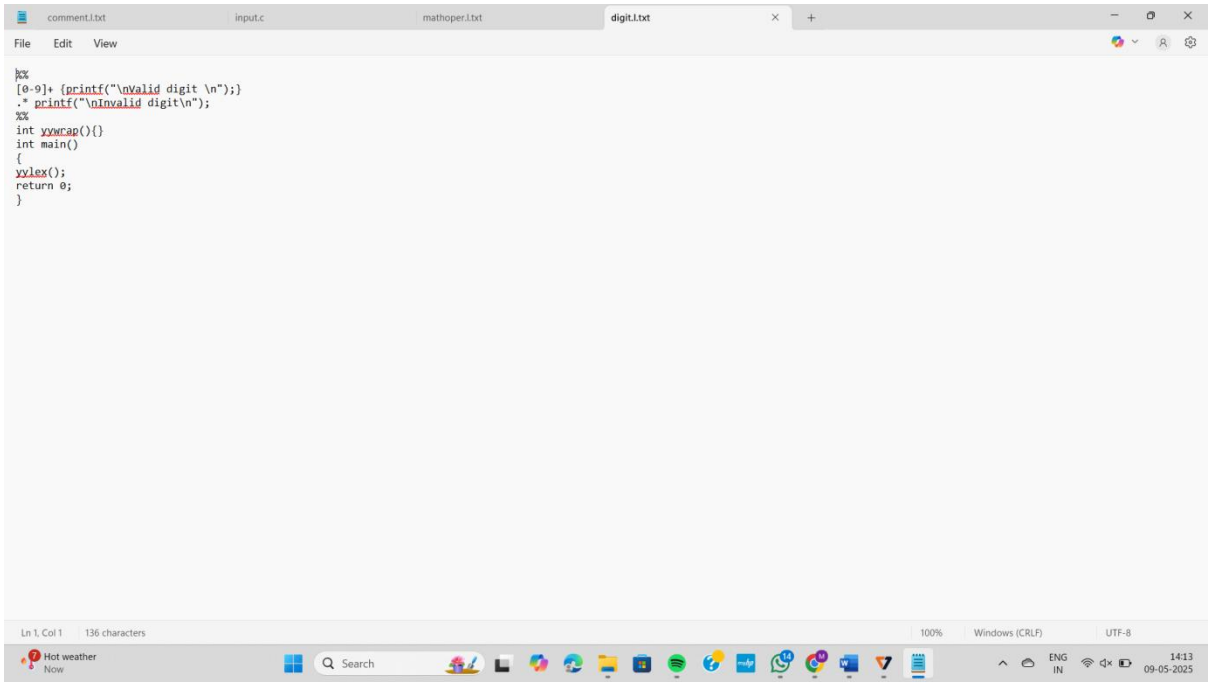
Output:



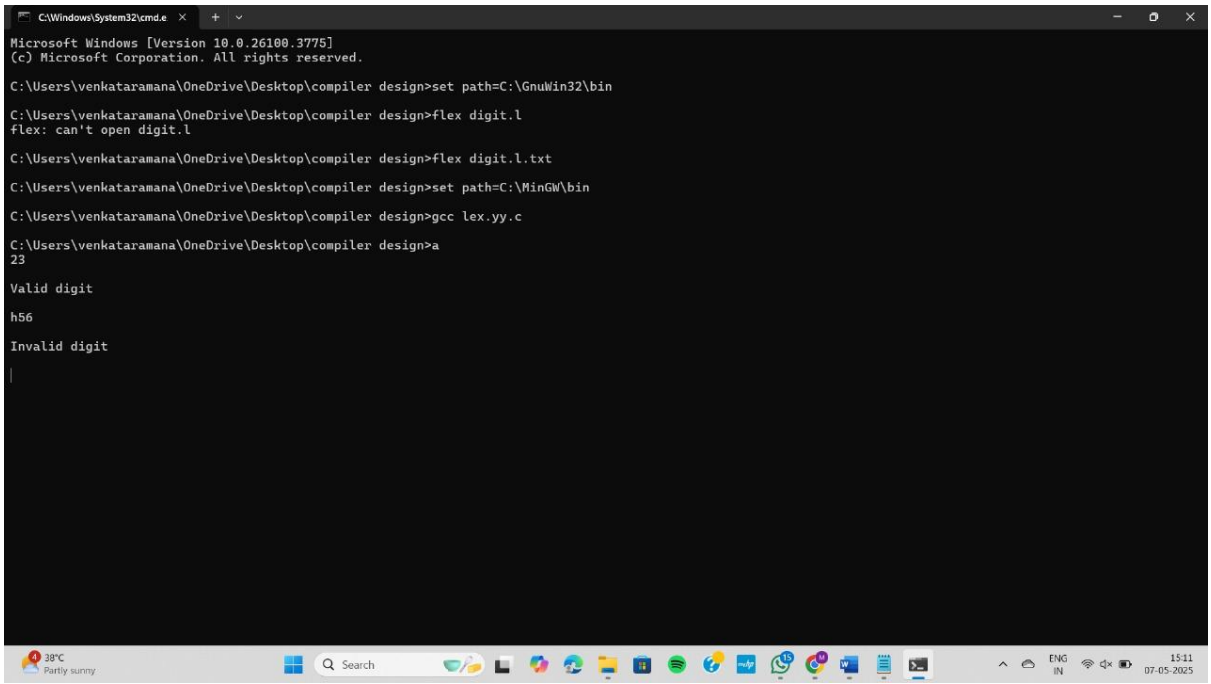
```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex mathoper.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
20+30
The answer is = 50.000000
40-20
The answer is = 20.000000
```

2) Write a LEX program to check whether the given input is digit or not.

Program:

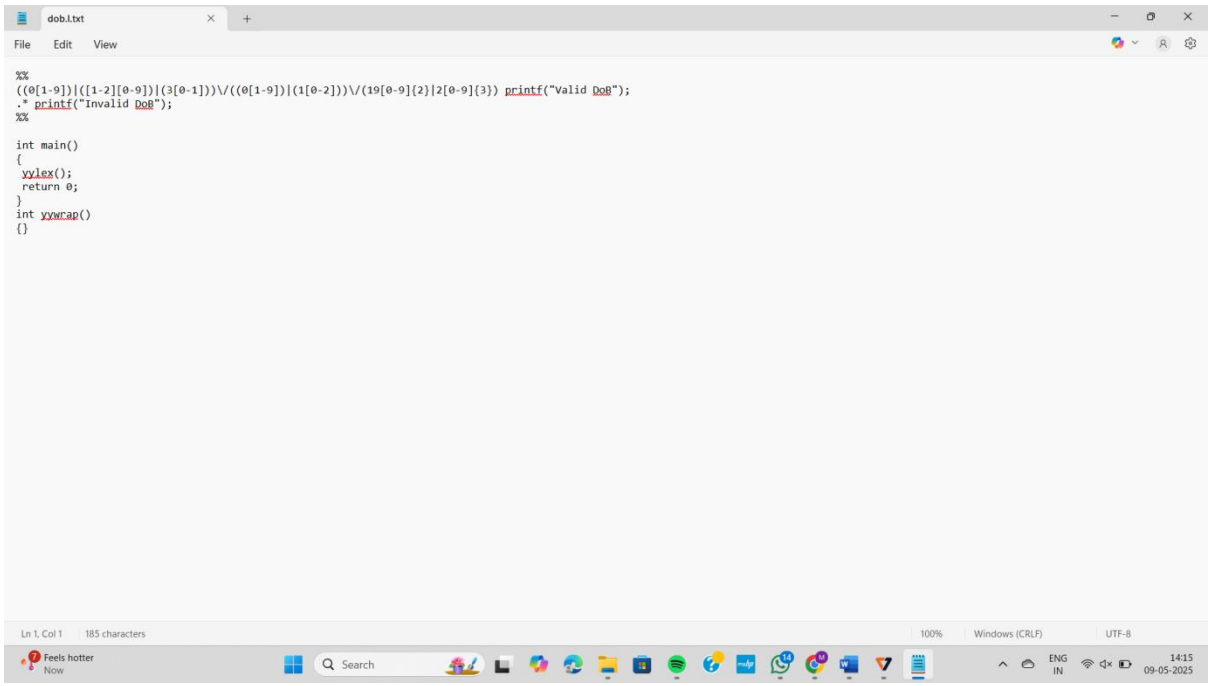

comment.l.txt
input.c
mathoper.l.txt
digit.l.txt
File
Edit
View
%%
[0-9]+ {printf("\nValid digit \n");}
.* printf("\nInvalid digit\n");
%%
int yywrap(){}
int main()
{
yylex();
return 0;
}
Ln 1, Col 1
136 characters
100%
Windows (CRLF)
UTF-8
Hot weather
Now
Search
ENG
IN
14:13
09-05-2025

Output:

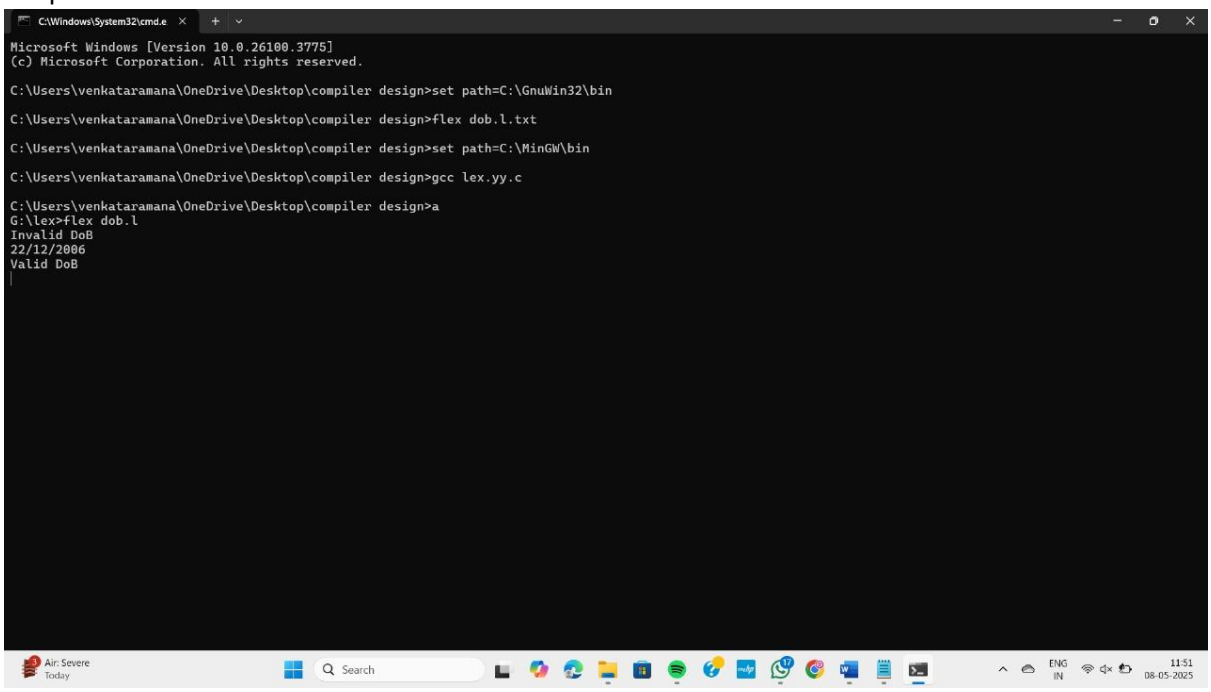

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.
C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin
C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex digit.l
flex: can't open digit.l
C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex digit.l.txt
C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin
C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c
C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
23
Valid digit
h56
Invalid digit
38°C
Partly sunny
Search
ENG
IN
15:11
07-05-2025

3) Write a LEX program to validate DOB of students.

Program:



Output:



4) Write a LEX program to validate the URL.

Program:

```
%%
((http)|(ftp))s?:\/\/[a-zA-Z0-9](.[a-z])+(.[a-zA-Z0-9+=?]*)* {printf("\nURL Valid\n");}

.+ {printf("\nURL Invalid\n");}

%%
void main()
{
        printf("\nEnter URL : ");
        yylex();
        printf("\n");
}
int yywrap()
{
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex url.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a

Enter URL : https://www.sse.in

URL Valid
```

5) Write a LEX program to identify and count positive and negative numbers.

Program:

```
%{
int positive_no = 0, negative_no = 0;
%}
%%
^[-][0-9]+ {negative_no++;
                        printf("negative number = %s\n",
                                yytext);} // negative number

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

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex posneg.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
20
positive number = 20

-21
negative number = -21
```

6) Write a LEX program to recognise numbers and words in a statement.

Program:

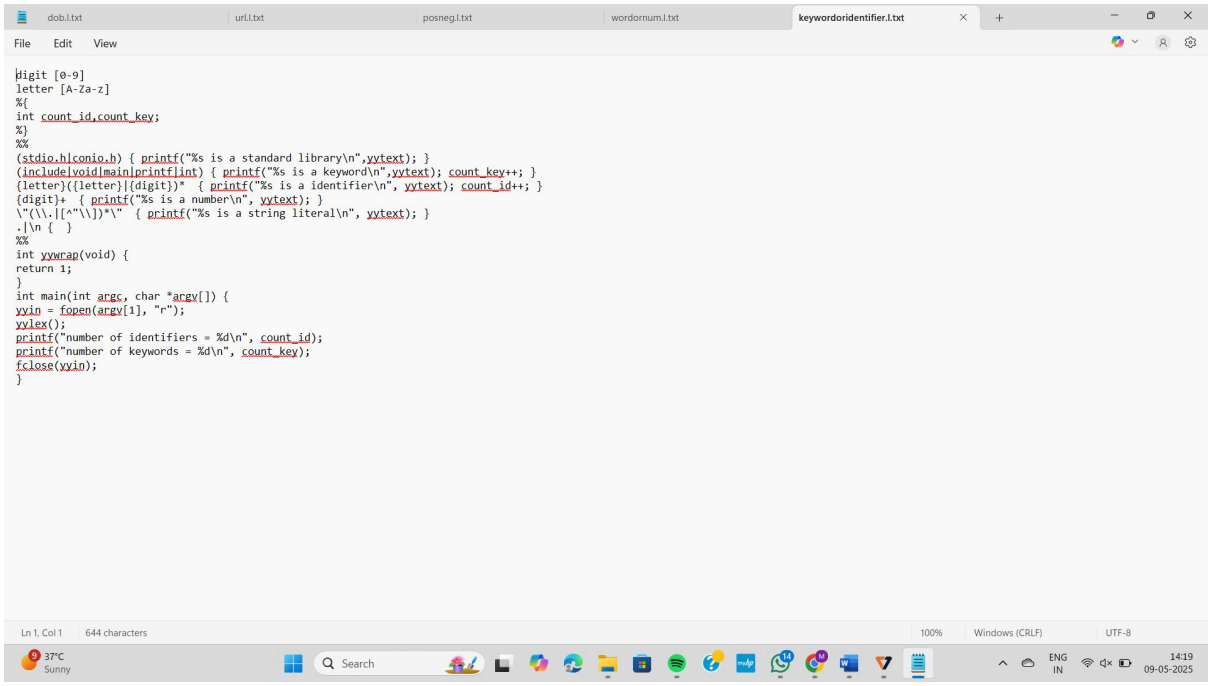
```
%%
[\t ]+ ;
[0-9]+|[0-9]*\.[0-9]+ { printf("\n%s is NUMBER", yytext);}
#.* { printf("\n%s is COMMENT", yytext);}
[a-zA-Z]+ { printf("\n%s is WORD", yytext);}
\n { ECHO;}
%%
int main()
{
        while( yylex());
}

int yywrap( )
{
        return 1;
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex wordornum.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>ycc lex.yy.c
'ycc' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
variable contains 10 and 20

variable is WORD
contains is WORD
10 is NUMBER
and is WORD
20 is NUMBER
```

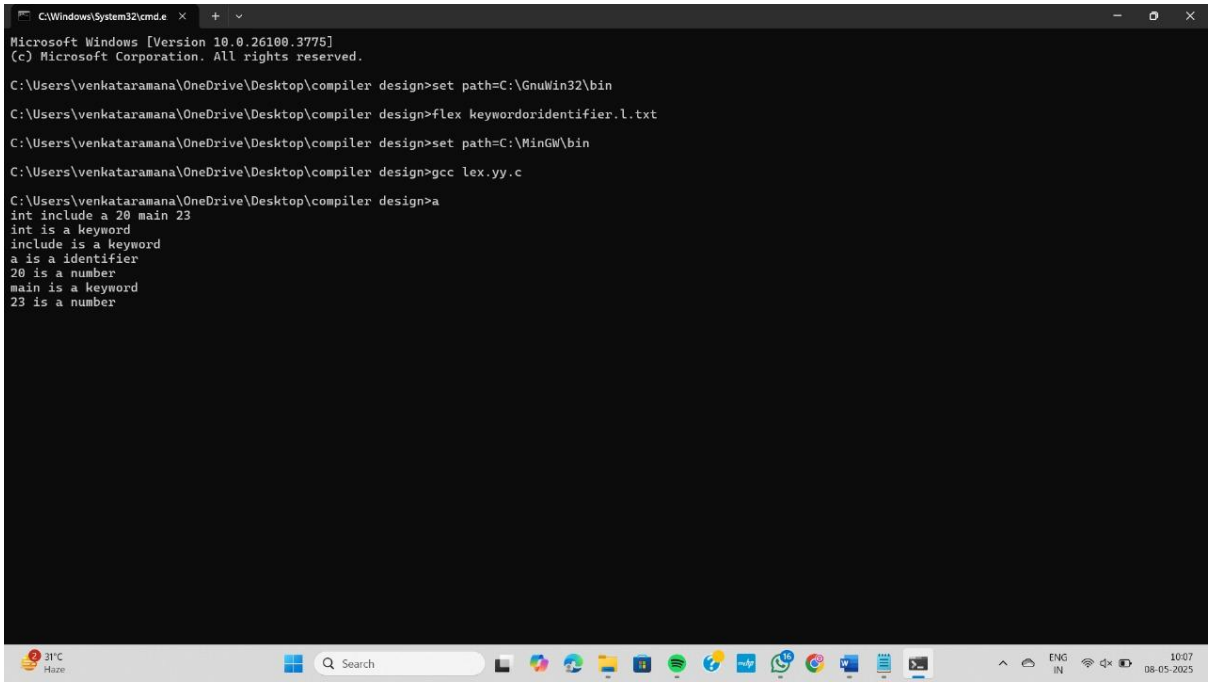
7) Write a LEX program to separate the keywords and identifiers.

Program:



```
digit [0-9]
letter [A-Za-z]
%{
int count_id,count_key;
%}
%%
(stdio.h|conio.h) { printf("%s is a standard library\n",yytext); }
(include|void|main|printf|int) { printf("%s is a keyword\n",yytext); count_key++; }
{letter}({letter}|{digit})*  { printf("%s is a identifier\n", yytext); count_id++; }
{digit}+  { printf("%s is a number\n", yytext); }
\"(\\.|[^"\\])*\"  { printf("%s is a string literal\n", yytext); }
.|\n {  }
%%
int yywrap(void) {
return 1;
}
int main(int argc, char *argv[]) {
yyin = fopen(argv[1], "r");
yylex();
printf("number of identifiers = %d\n", count_id);
printf("number of keywords = %d\n", count_key);
fclose(yyin);
}
```

Output:



```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex keywordoridentifier.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
int include a 20 main 23
int is a keyword
include is a keyword
a is a identifier
20 is a number
main is a keyword
23 is a number
```

8) Write a LEX program to count the number of vowels in the given sentence.

Program:

```
%{
    int vow_count=0;
    int const_count =0;
%}

%%
[aeiouAEIOU] {vow_count++;}
[a-zA-Z] {const_count++;}
%%
int yywrap(){}
int main()
{
    printf("Enter the string of vowels and consonants:");
    yylex();
    printf("Number of vowels are:  %d\n", vow_count);
    printf("Number of consonants are:  %d\n", const_count);
    return 0;
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex vowelsandconsonants.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
Enter the string of vowels and consonants:compiler

^Z
Number of vowels are:  3
Number of consonants are:  5

C:\Users\venkataramana\OneDrive\Desktop\compiler design>
```

9) Implement a LEX program to check whether the mobile number is valid or not.

Program:

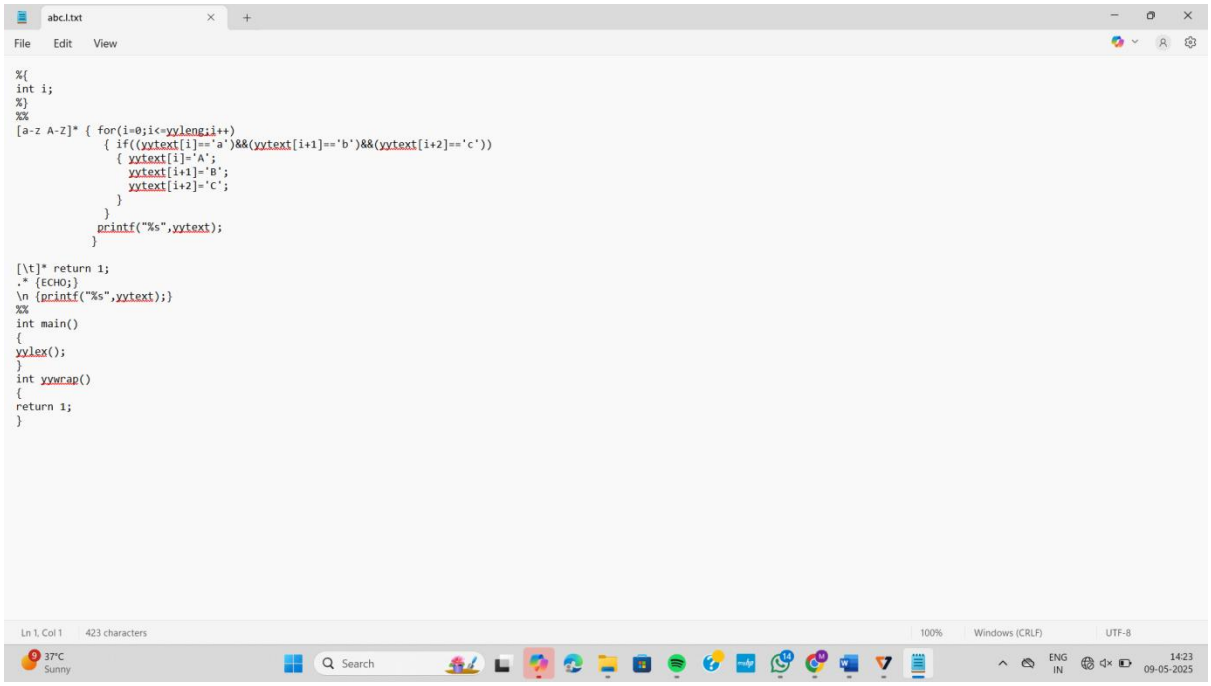
```
%%
[1-9][0-9]{9} {printf("\nMobile Number Valid\n");}
.+ {printf("\nMobile Number Invalid\n");}
%%
int main()
{
        printf("\nEnter Mobile Number : ");
        yylex();
        printf("\n");
        return 0;
}
int yywrap()
{ }
```

Output:

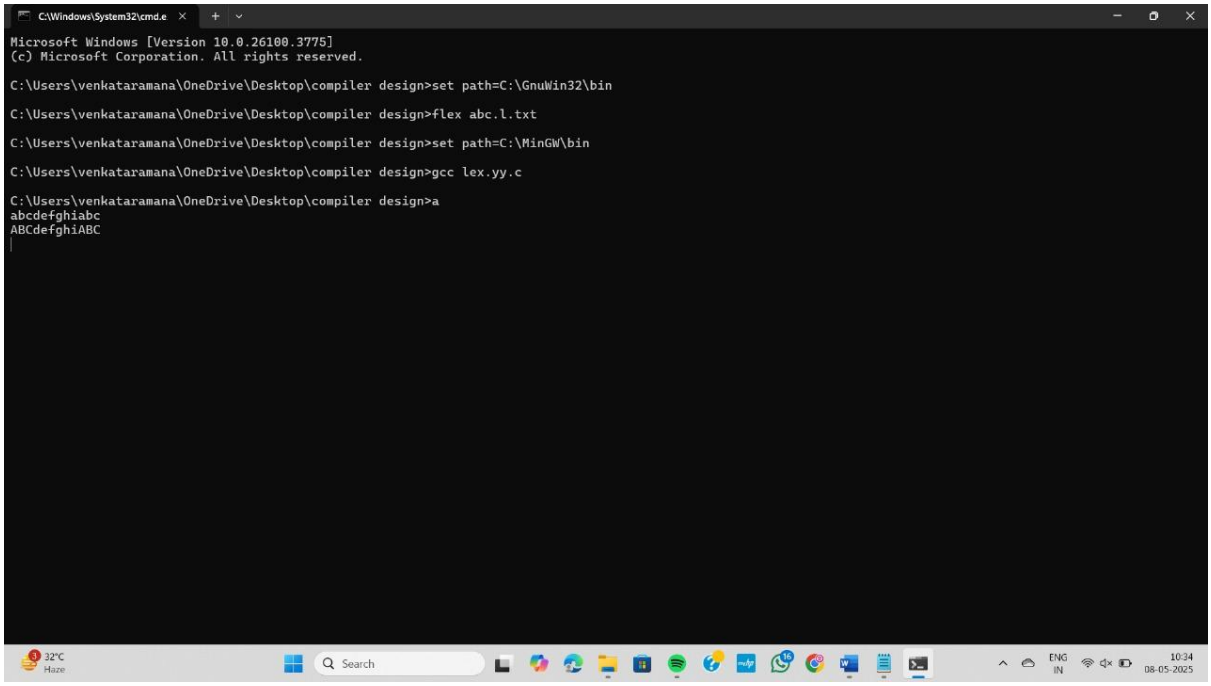
```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex mobile.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a

Enter Mobile Number : 1234

Mobile Number Invalid

1234567890

Mobile Number Valid
```

10) Write a LEX Program to convert the substring abc to ABC from the given input string

Program:



Output:



11) Write a LEX Program to check the email address is valid or not.

Program:

email.l.txt

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

%%

[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.com      { flag = 1; }
[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.in       { flag = 1; }
.|\n                                       { /* ignore other characters */ }

%%

int main()
{
    printf("Enter an email address:\n");
    yylex();
    if(flag == 1)
        printf("Accepted\n");
    else
        printf("Not Accepted\n");
    return 0;
}

int yywrap() {
    return 1;
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex email.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c -o email.exe

C:\Users\venkataramana\OneDrive\Desktop\compiler design>email.exe
Enter an email address:
sse123@gmail.com
^z
^Z
Accepted

C:\Users\venkataramana\OneDrive\Desktop\compiler design>
```

12) Write a LEX program to identify the capital words from the given input.

Program:

```
%%
[A-Z]+[\t\n ] { printf("%s is a capital word\n",yytext); }
. ;
%%

int main( )
{
        printf("Enter String :\n");
        yylex();
}
int yywrap( )
{
        return 1;
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Documents>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Documents>flex cap.l.txt

C:\Users\venkataramana\OneDrive\Documents>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Documents>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Documents>a
Enter String :
mayuRI
RI
 is a capital word
```

13) Write a LEX program to count the number of comment lines in a given C program and eliminate them and write into another file.

Program:

comment.l.txt

```
%{
int com=0;
%}
%s COMMENT
%%
"/*" {BEGIN COMMENT;}
<COMMENT>"*/" {BEGIN 0; com++;}
<COMMENT>\n {com++;}
<COMMENT>. {;}
\/\/.* {; com++;}
.|\n {fprintf(yyout,"%s",yytext);}
%%
void main(int argc, char *argv[])
{
if(argc!=3)
{
printf("usage : a.exe input.c output.c\n");
exit(0);
}
yyin=fopen(argv[1],"r");
yyout=fopen(argv[2],"w");
yylex();
printf("\n number of comments are = %d\n",com);
}
int yywrap()
{
return 1;
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex comment.l.txt

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

C:\Users\venkataramana\OneDrive\Desktop\compiler design>gcc lex.yy.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>a
usage : a.exe input.c output.c

C:\Users\venkataramana\OneDrive\Desktop\compiler design>remove_comments.exe input.c output.c

 number of comments are = 2

C:\Users\venkataramana\OneDrive\Desktop\compiler design>
```