

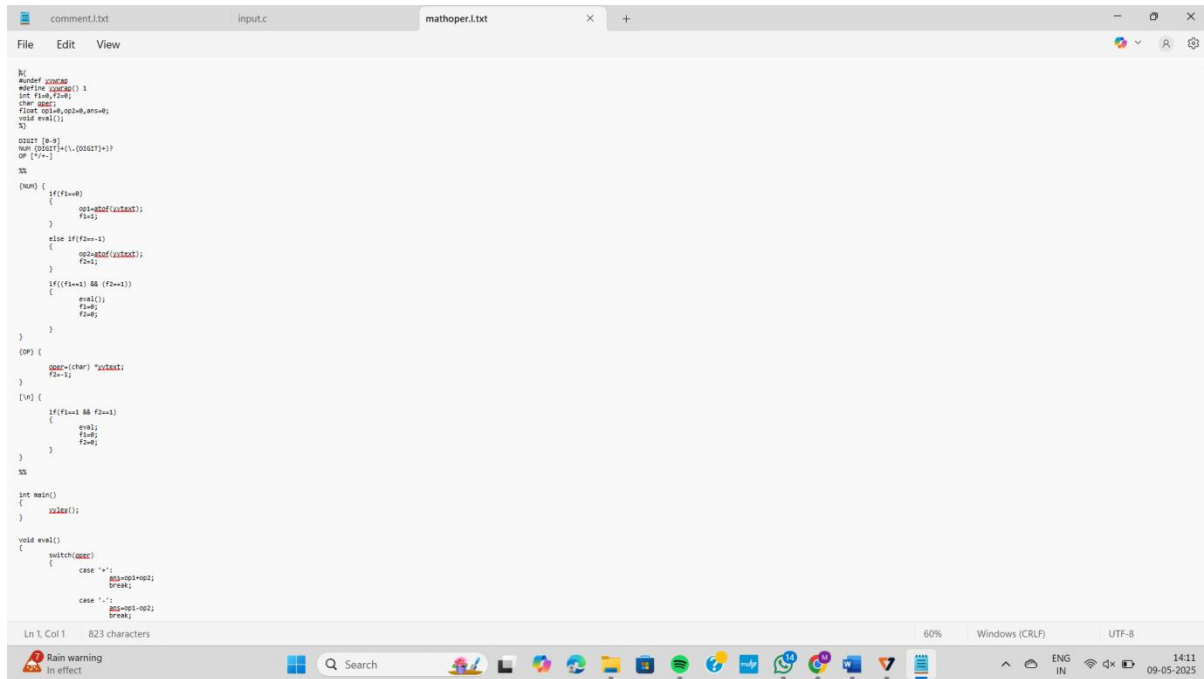
Name: ch.ishwarya

reg no: 192372104

course: csa1409 – compiler design

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

Program:



```

%{
#include <stdio.h>
#define YYSTYPE int
int f1=0,f2=0;
char op;
float op1=0,op2=0,ans=0;
void eval();
%}

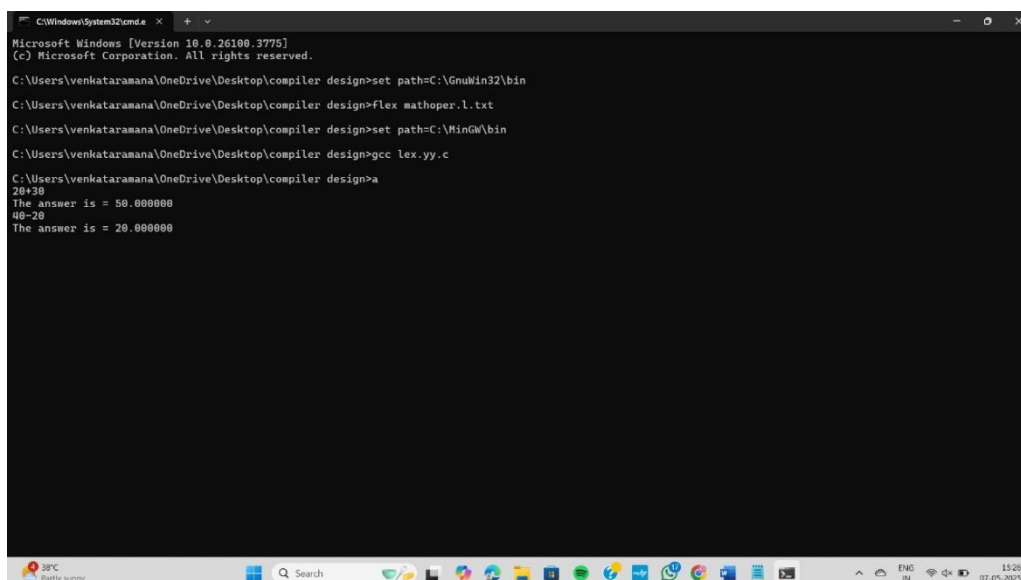
%%
[0-9] {
    if(f1==0)
    {
        op1=atoi(yytext);
        f1++;
    }
    else if(f2==1)
    {
        op2=atoi(yytext);
        f2++;
    }
    if(f1==1 && f2==1)
    {
        eval();
        f1=0;
        f2=0;
    }
}
[+] {
    op=*(char*)yytext;
    f2=1;
}
[\n] {
    if(f1==1 && f2==1)
    {
        eval();
        f1=0;
        f2=0;
    }
}
%%

int main()
{
    scanf("%s");
}

void eval()
{
    switch(op)
    {
        case "+":
            ans=op1+op2;
            break;
        case "-":
            ans=op1-op2;
            break;
    }
}

```

Output:



```

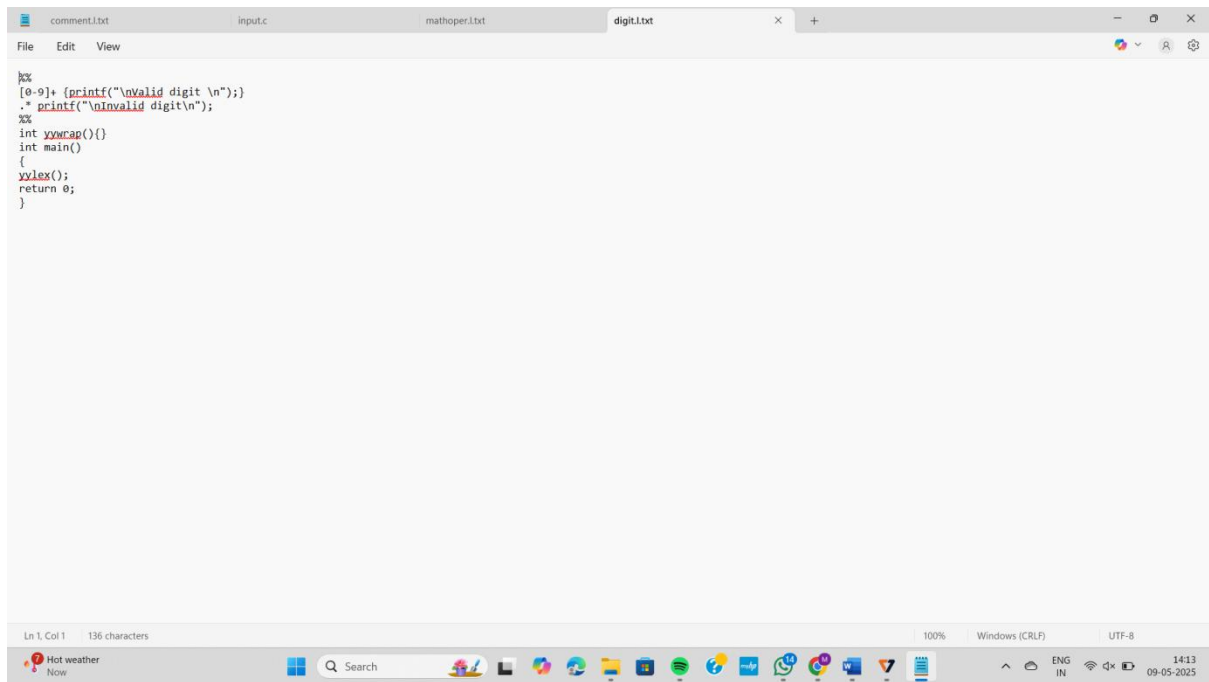
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3779]
(c) Microsoft Corporation. All rights reserved.

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:

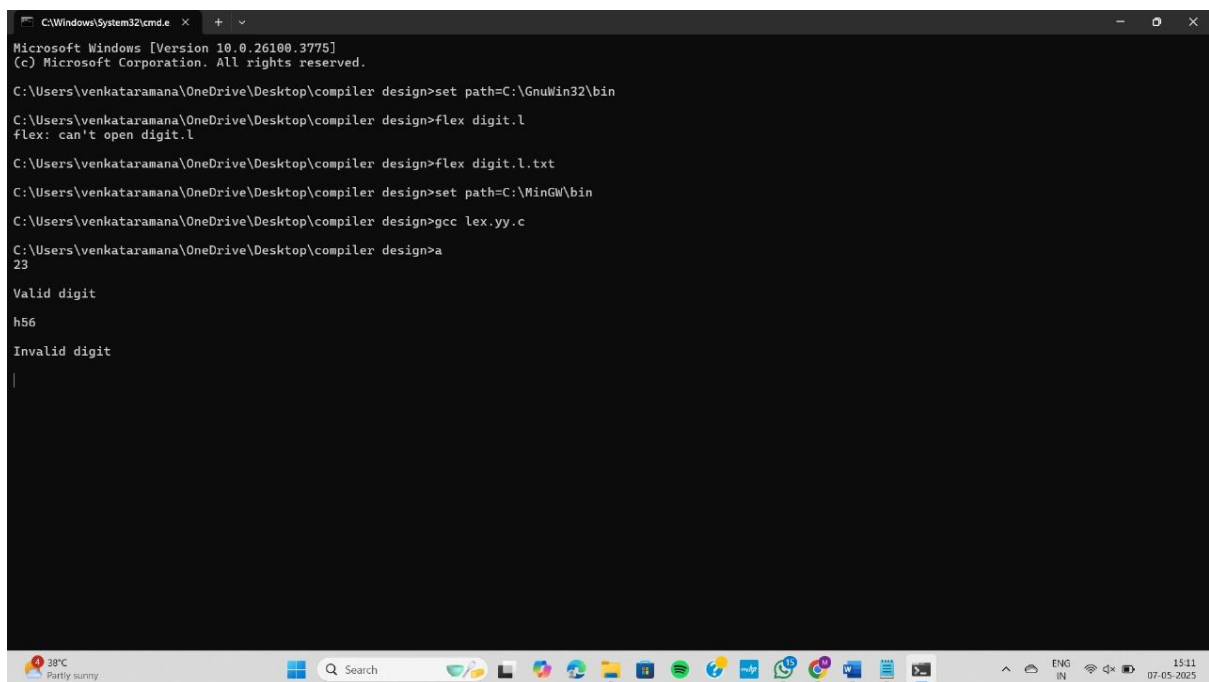


```
comment.txt input.c mathoper.txt digit.l.txt
File Edit View

%%
[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 14:13 09-05-2025
```

Output:



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

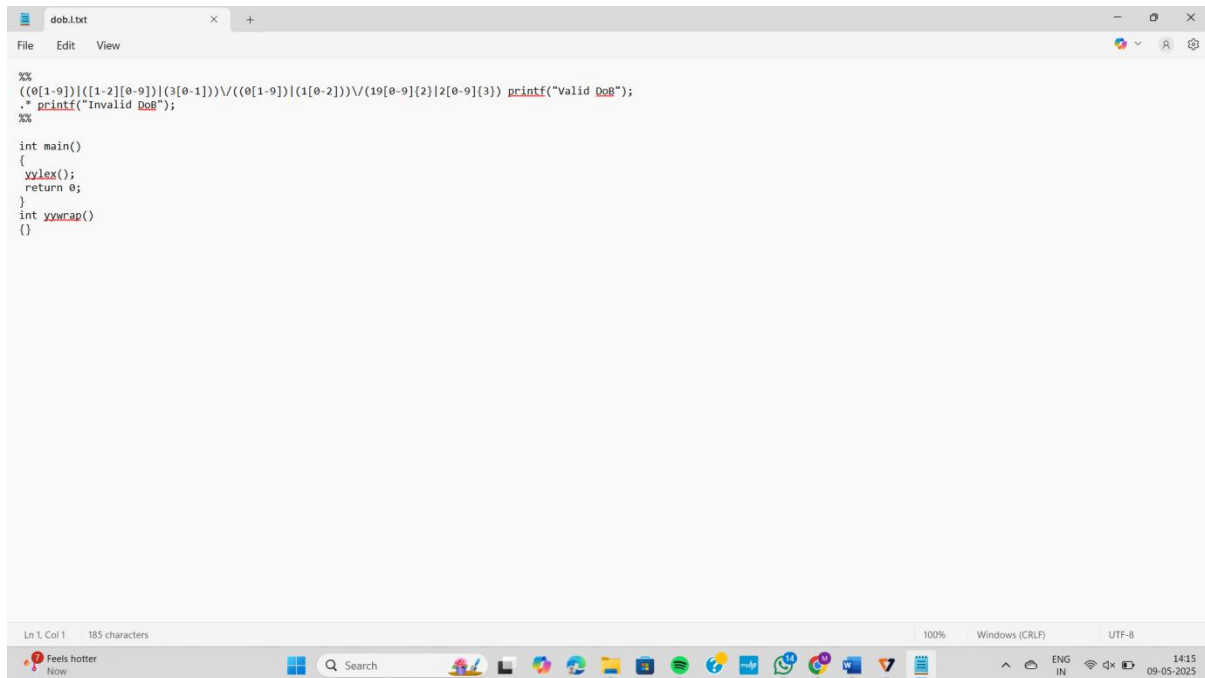
C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin
C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex digit.l
flex: can't open digit.l

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 15:11 07-05-2025
```

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

Program:

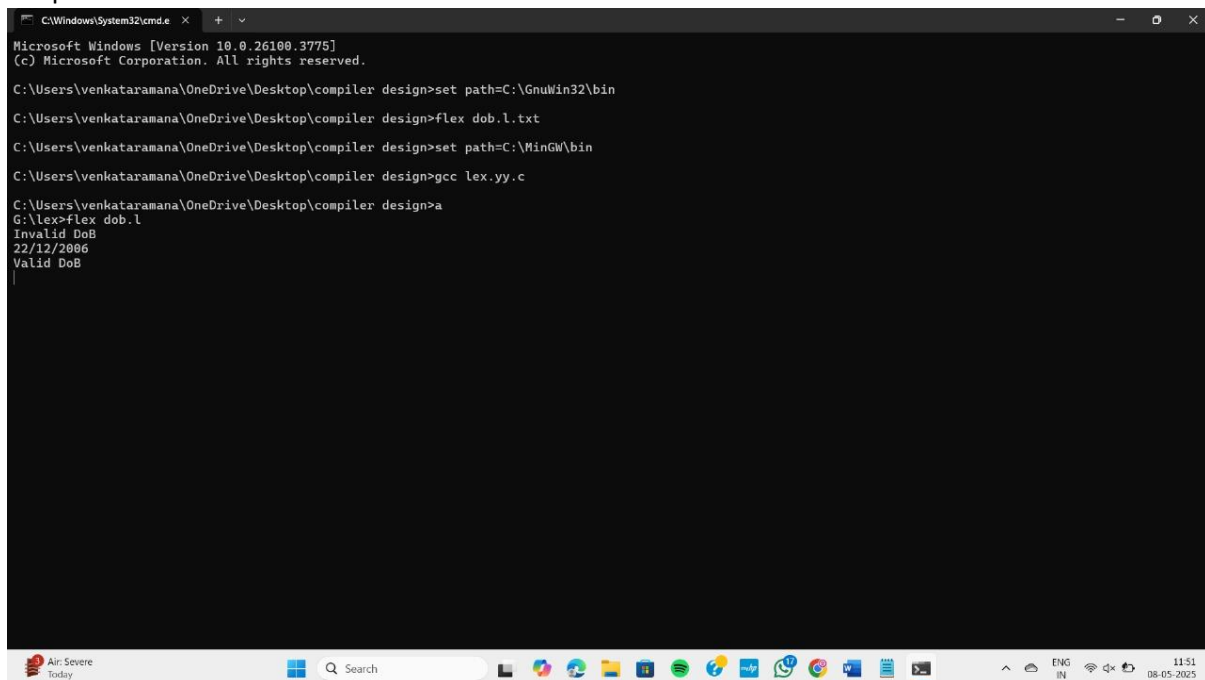


```
dob.l.txt
File Edit View

%%
((0[1-9])|([1-2][0-9])|3[0-1]))\(((0[1-9])|(1[0-2]))\)(19[0-9]{2}|2[0-9]{3}) printf("Valid DoB");
.* printf("Invalid DoB");
%%

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

Output:

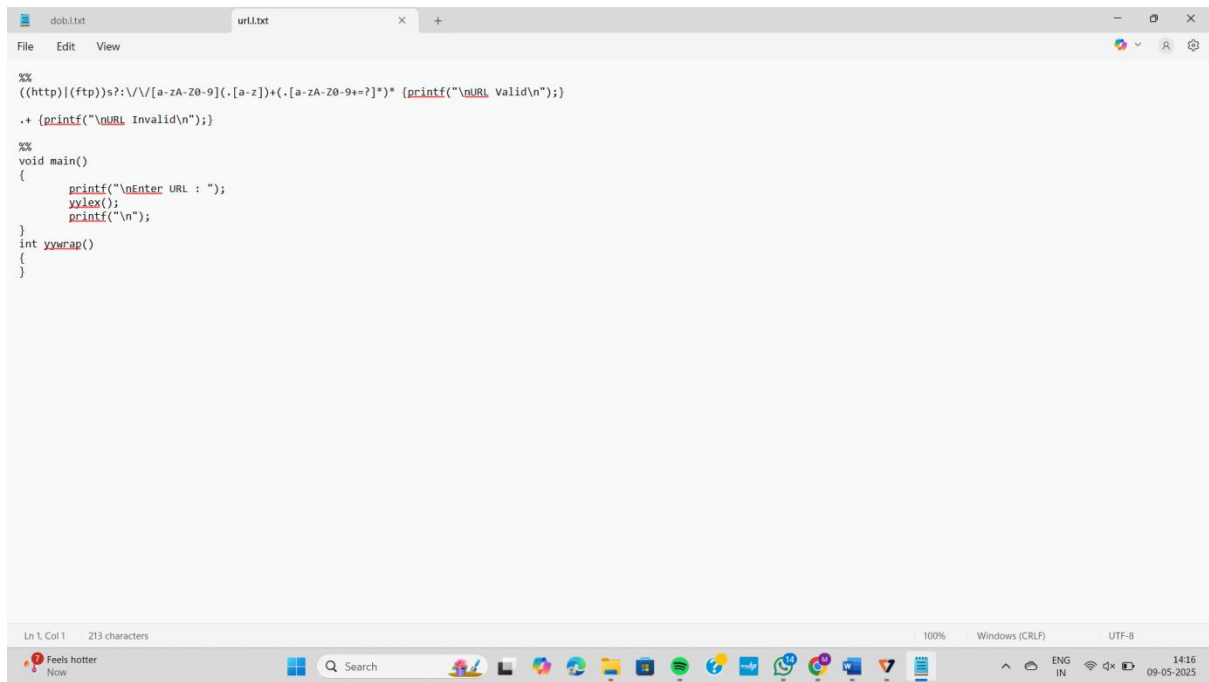


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin
C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex dob.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
G:\lex>flex dob.l
Invalid DoB
22/12/2006
Valid DoB
|
```

4) Write a LEX program to validate the URL.

Program:

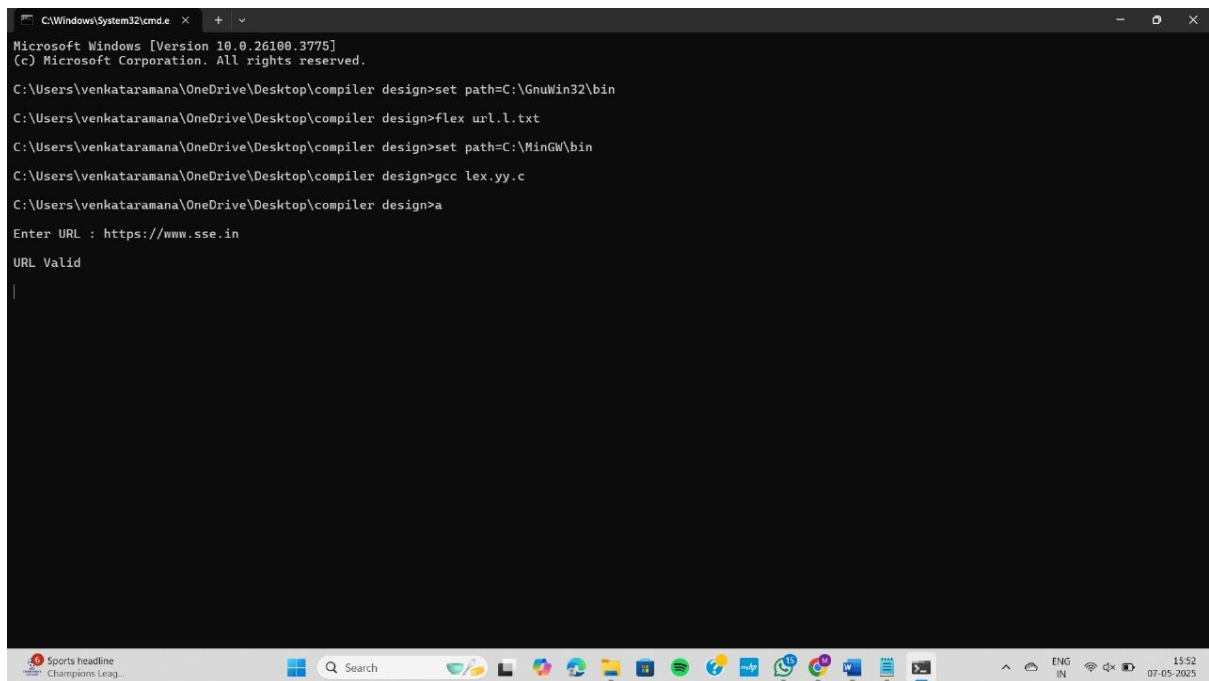


The screenshot shows a text editor with two tabs: 'dob1.txt' and 'url1.txt'. The 'url1.txt' tab is active, displaying a Lex program. The program defines a regular expression for valid URLs and a main function that prompts the user to enter a URL and prints whether it is valid or invalid. The status bar at the bottom indicates 'Ln 1, Col 1' and '213 characters'.

```
%%
((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:



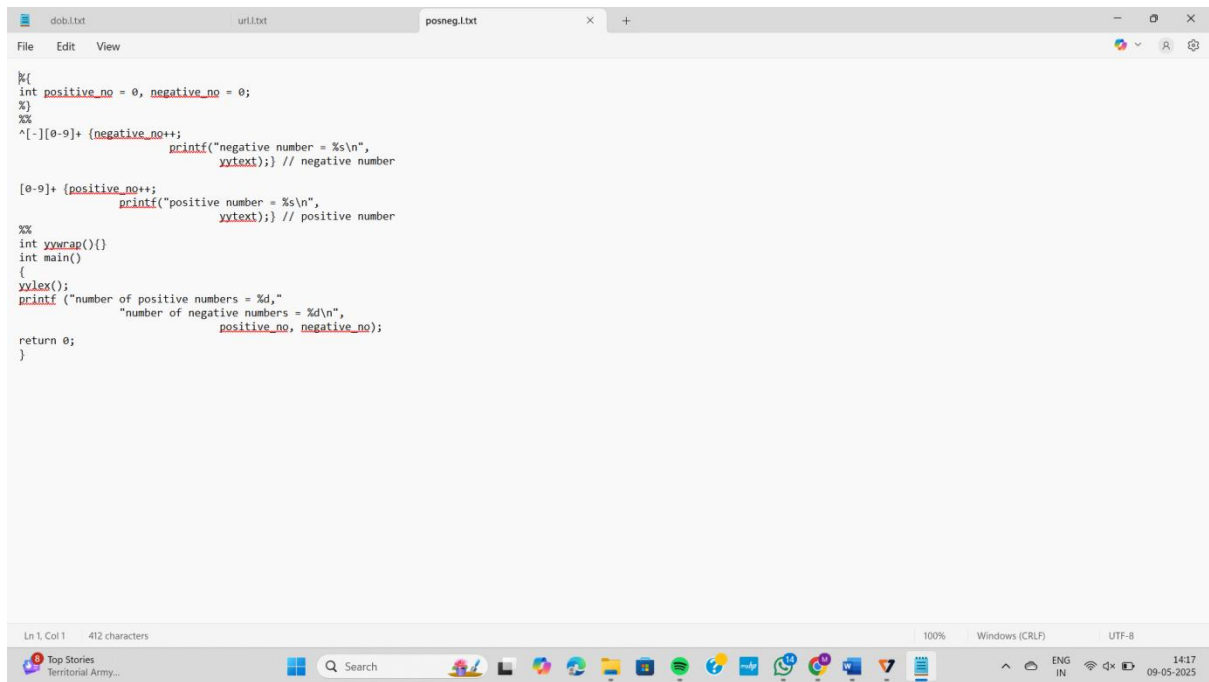
The screenshot shows a Windows command prompt window. The user has set the path to the GNUWin32 bin directory, compiled the 'url.l.txt' file using 'flex', and then compiled the resulting 'lex.yy.c' file using 'gcc'. Finally, they ran the executable 'a'. The program prompts for a URL, and the user enters 'https://www.sse.in', resulting in the output 'URL Valid'.

```
C:\Windows\System32\cmd.exe
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:

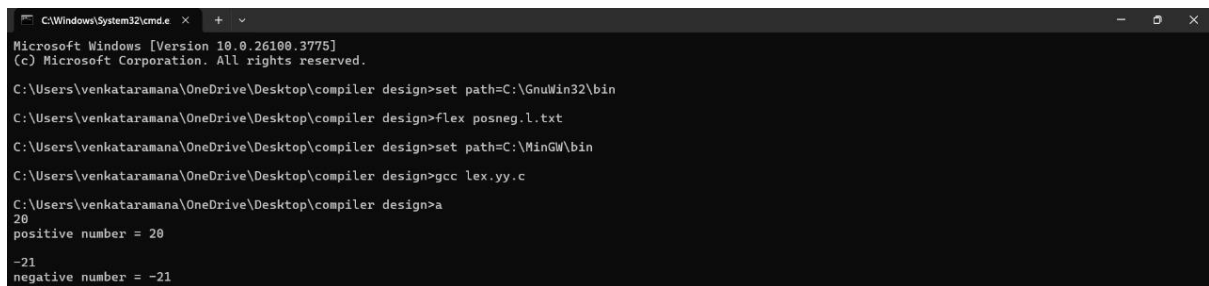


```
%{
int positive_no = 0, negative_no = 0;
}%
%%
^[-][0-9]+ {negative_no++;
              printf("negative number = %s\n",
                    yytext);} // negative number

[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:

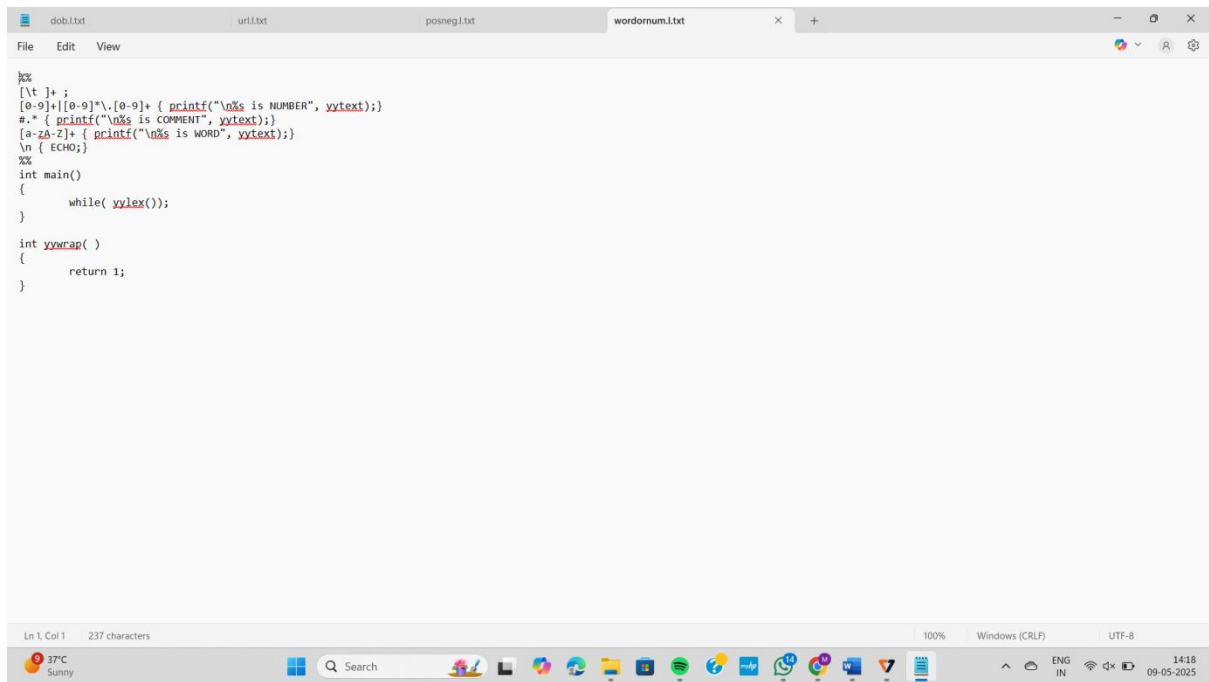


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

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.

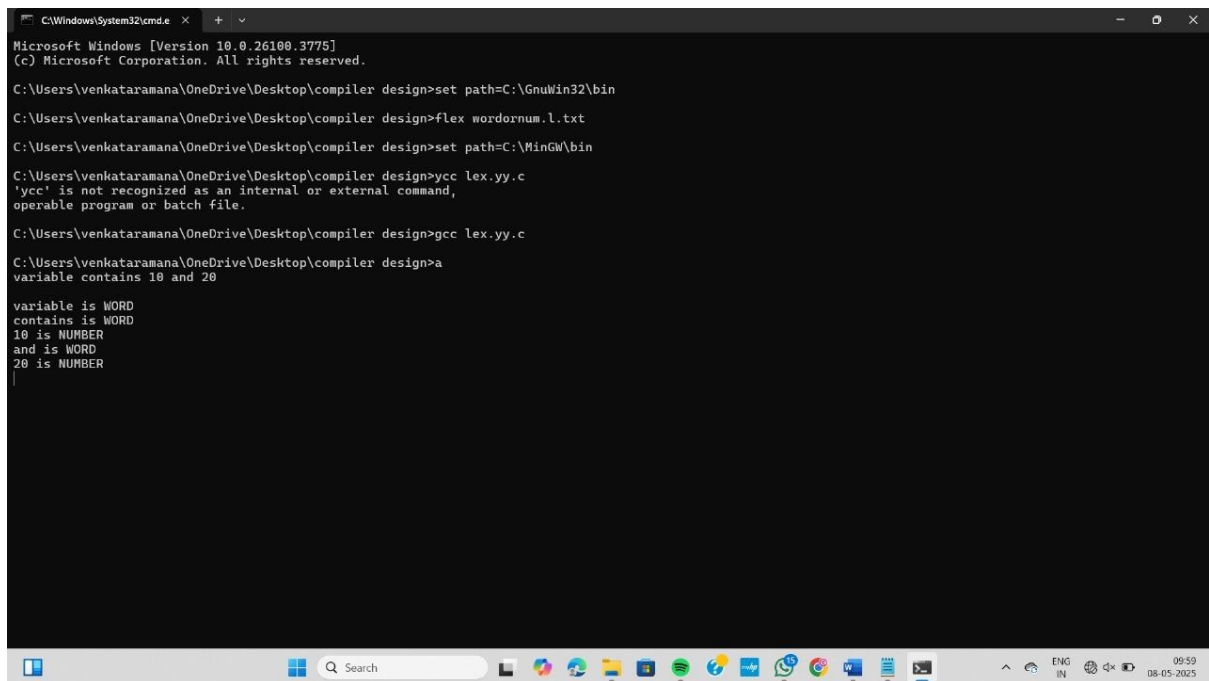
Program:



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

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

Output:



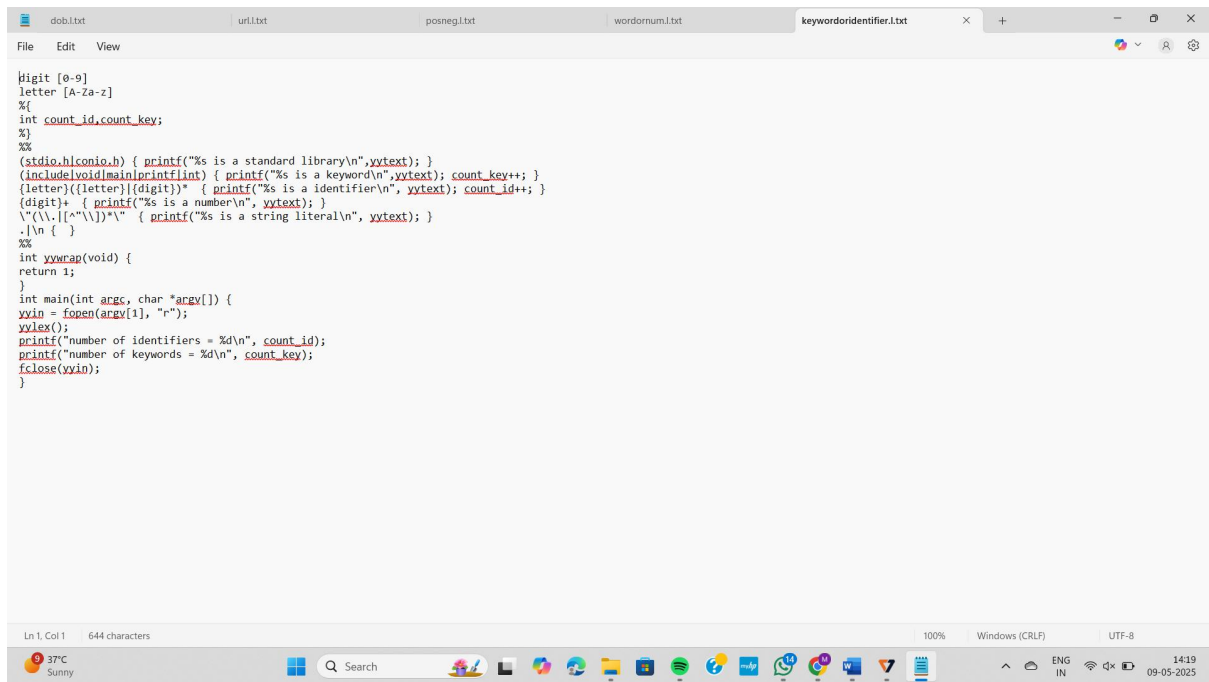
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

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.

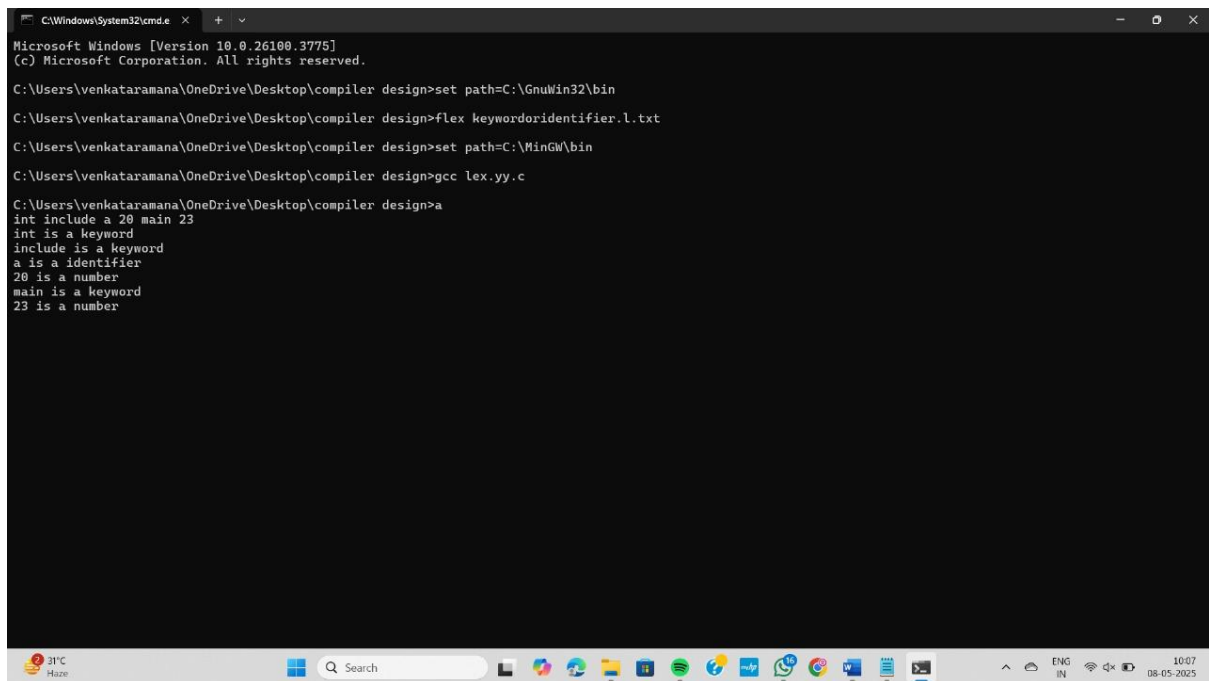
Program:



The screenshot shows a text editor window with the file 'keywordidentifier.l.txt' open. The code is a Lex program designed to identify keywords, identifiers, numbers, and string literals in a source file. It uses standard C preprocessor directives and printf for output. The program counts the number of identifiers and keywords found in the input file 'url.txt'.

```
#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}{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:



The screenshot shows a Windows command prompt window. The user has set the path to the GNUWin32 bin directory and compiled the Lex program 'keywordidentifier.l.txt' using 'flex'. They then set the path to the MinGW bin directory and executed the compiled program 'lex.yy.c'. The output shows the results of the Lex analysis for the input file 'url.txt'.

```
C:\Windows\System32\cmd.exe
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 keywordidentifier.l.txt

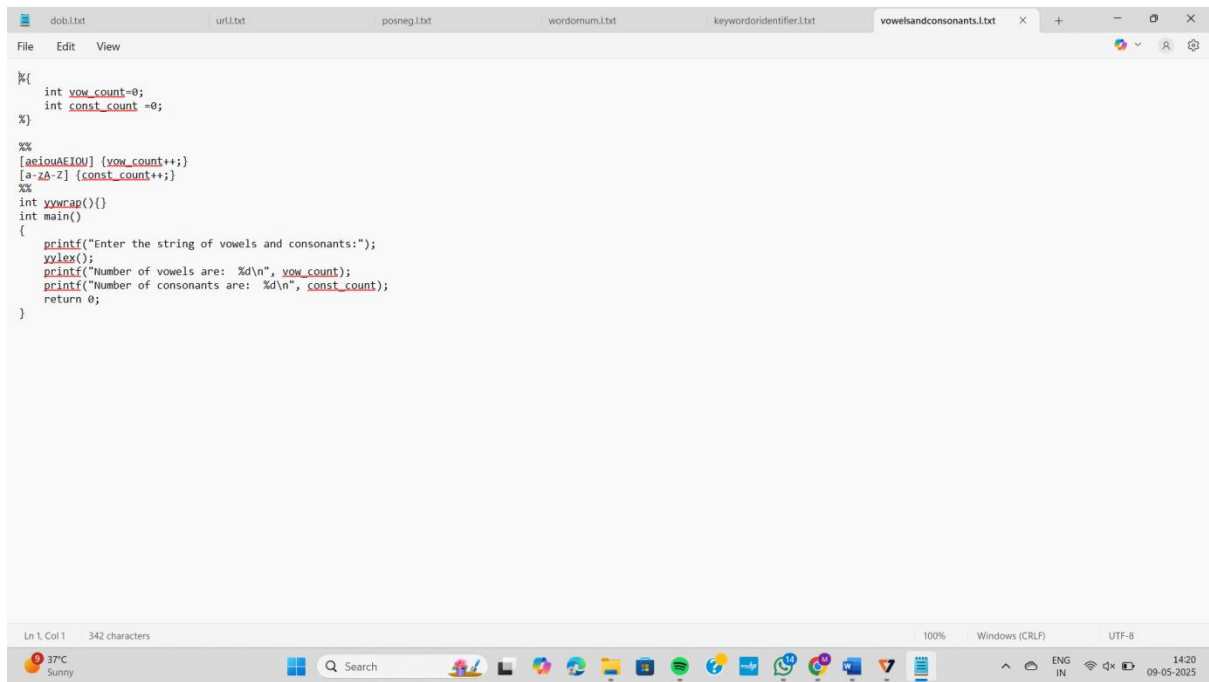
C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\MinGW\bin

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:



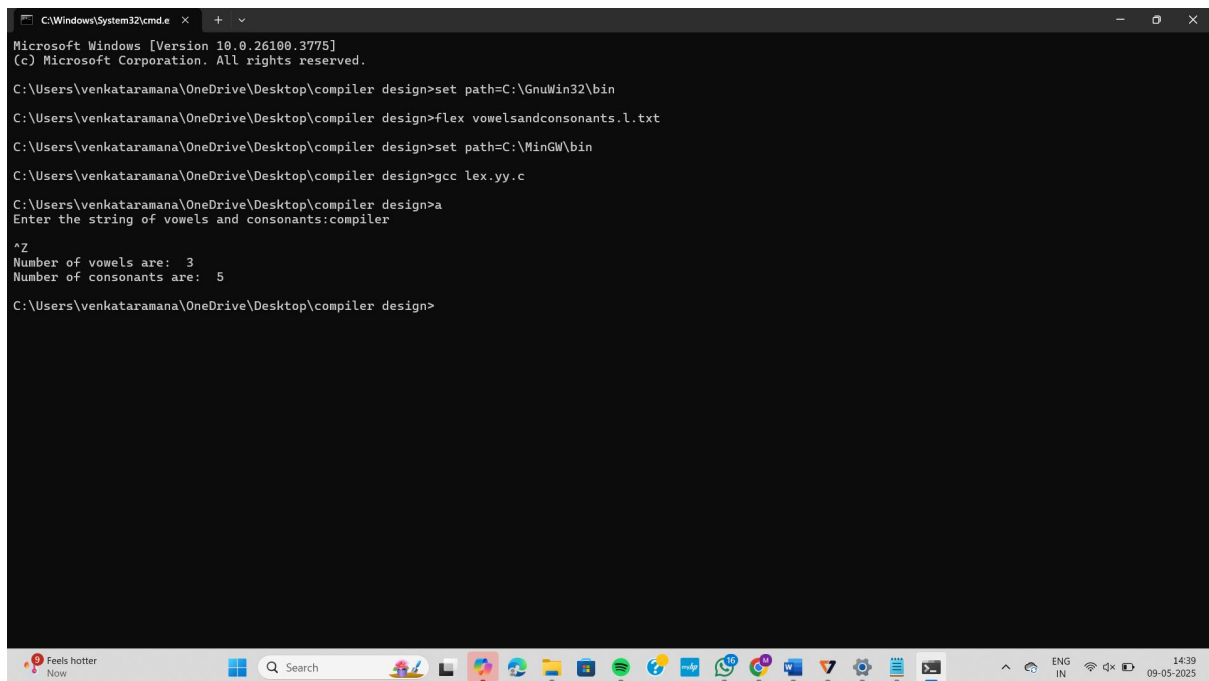
```
%{
    int vow_count=0;
    int const_count=0;
}%

%%
[aouiouAEIOU] {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;
}
```

Ln 1, Col 1 342 characters 100% Windows (CRLF) UTF-8

37°C Sunny

Output:



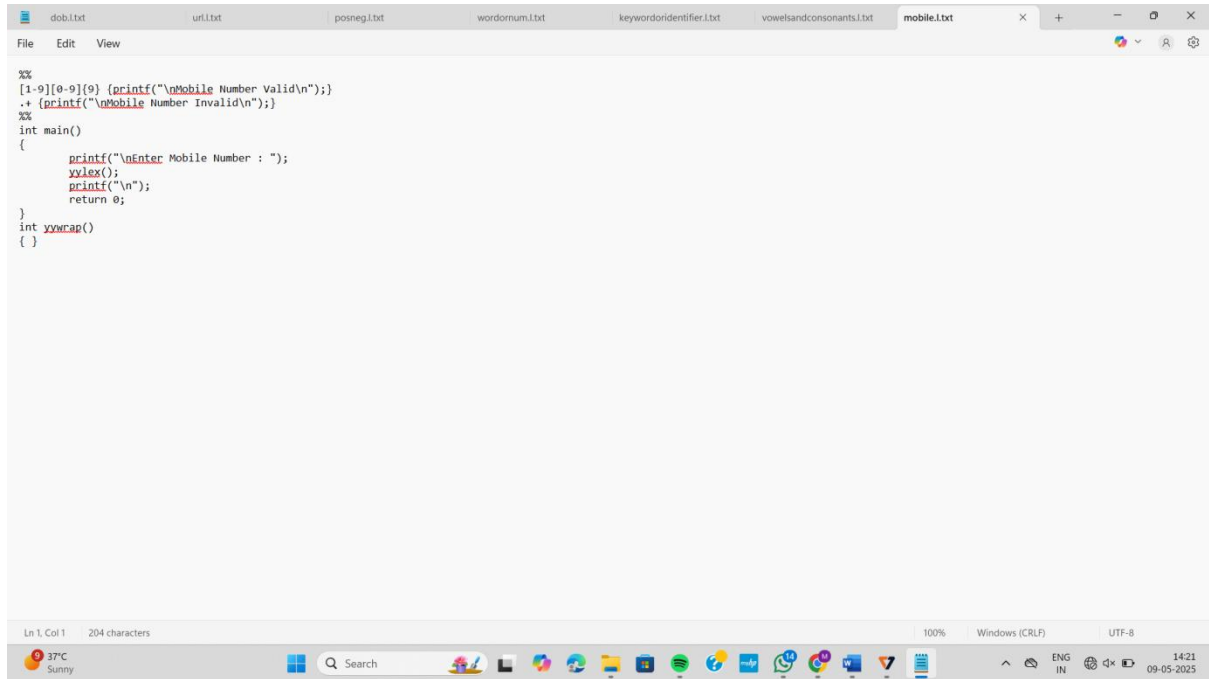
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

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>
```

Feels hotter Now

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

Program:

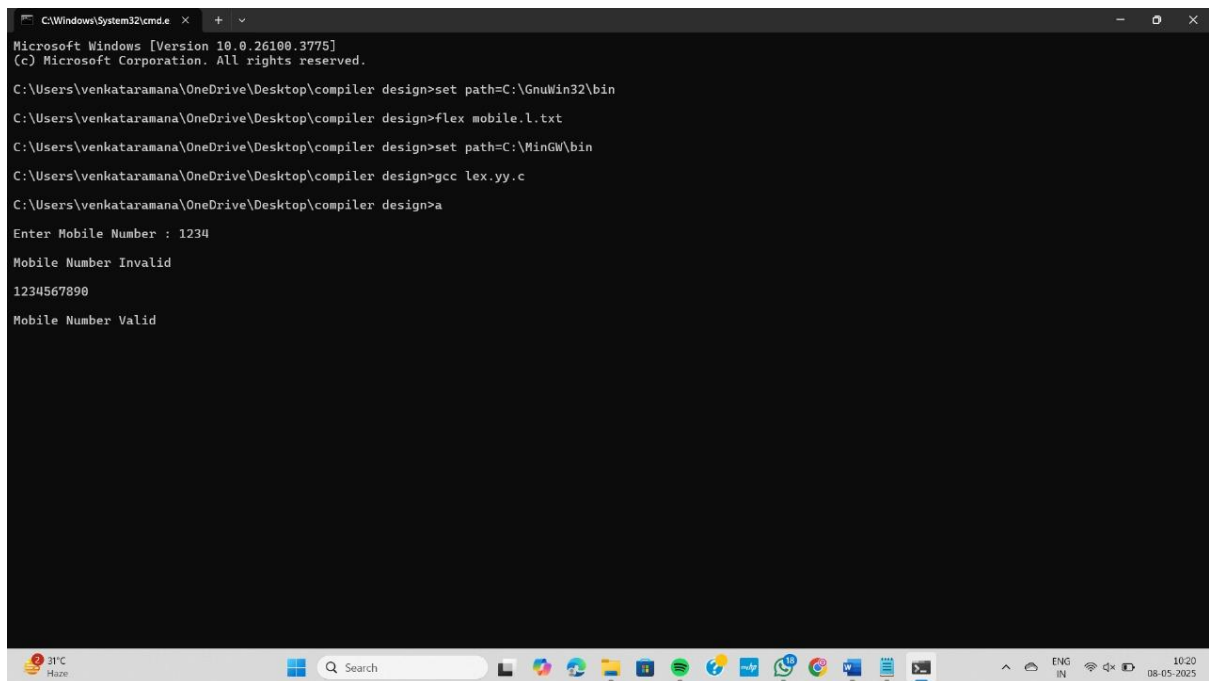


The screenshot shows a text editor with multiple tabs. The active tab is 'mobile.l.txt'. The code in the editor is a Lex program for validating mobile numbers. It uses a regular expression [1-9][0-9]{9} to match valid mobile numbers. The program prints 'Mobile Number Valid' for valid numbers and 'Mobile Number Invalid' for invalid ones. The main function prompts the user to enter a mobile number and calls the yywrap() function.

```
%%
[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()
{
}
```

The status bar at the bottom indicates 'Ln 1, Col 1' and '204 characters'.

Output:



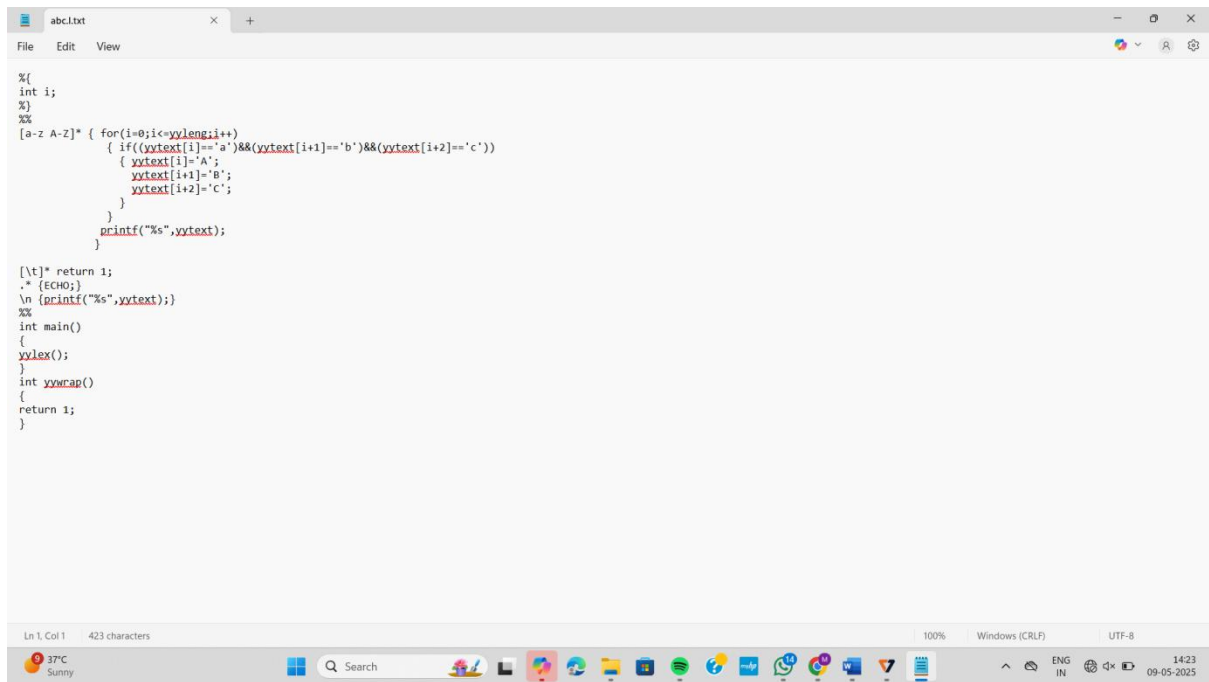
The screenshot shows a Windows command prompt window. The user has set the path to the compiler's bin directory and compiled the Lex program using flex and gcc. The program is then executed, and the user enters the mobile number 1234. The program outputs 'Mobile Number Invalid' because 1234 is not a valid mobile number. The user then enters 1234567890, and the program outputs 'Mobile Number Valid'.

```
C:\Windows\System32\cmd.exe
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

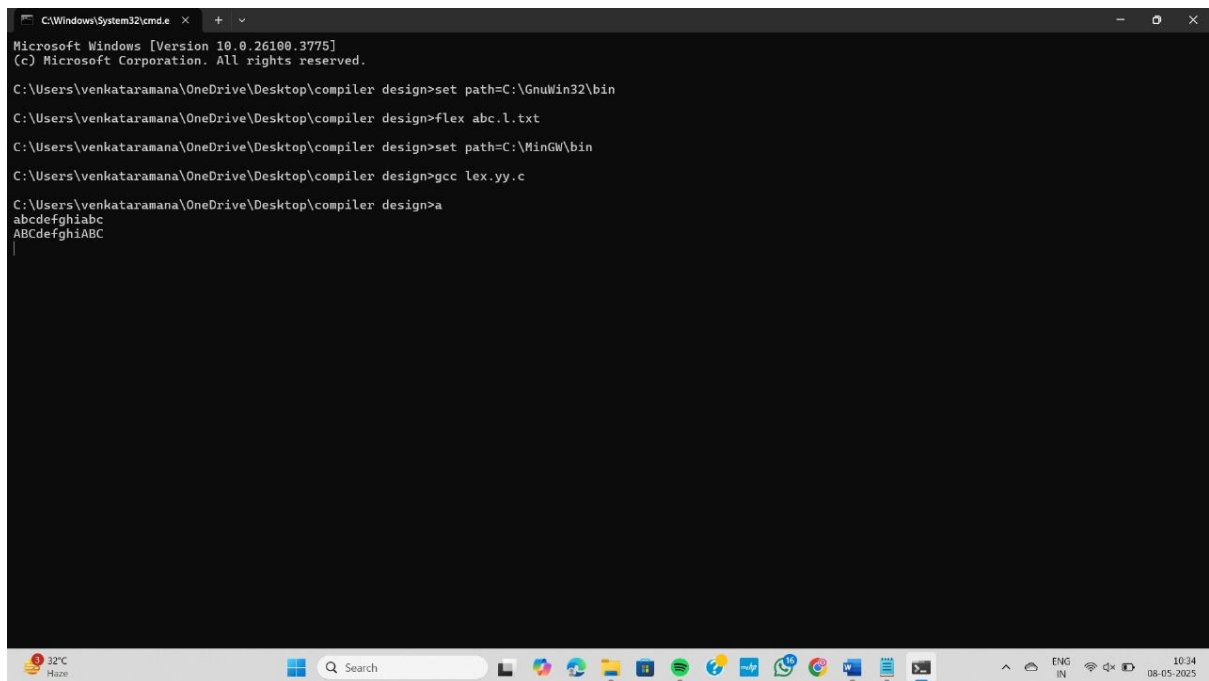
Program:



```
%{
int i;
}%
[a-zA-Z]* { for(i=0;i<yylength;i++)
{ if((yytext[i]!='a')&&(yytext[i+1]!='b')&&(yytext[i+2]!='c'))
{ yytext[i]='A';
yytext[i+1]='B';
yytext[i+2]='C';
}
}
printf("%s",yytext);
}

[\\t]* return 1;
.* {ECHO;}
\\n {printf("%s",yytext);}
}%
int main()
{
yylex();
}
int yywrap()
{
return 1;
}
```

Output:

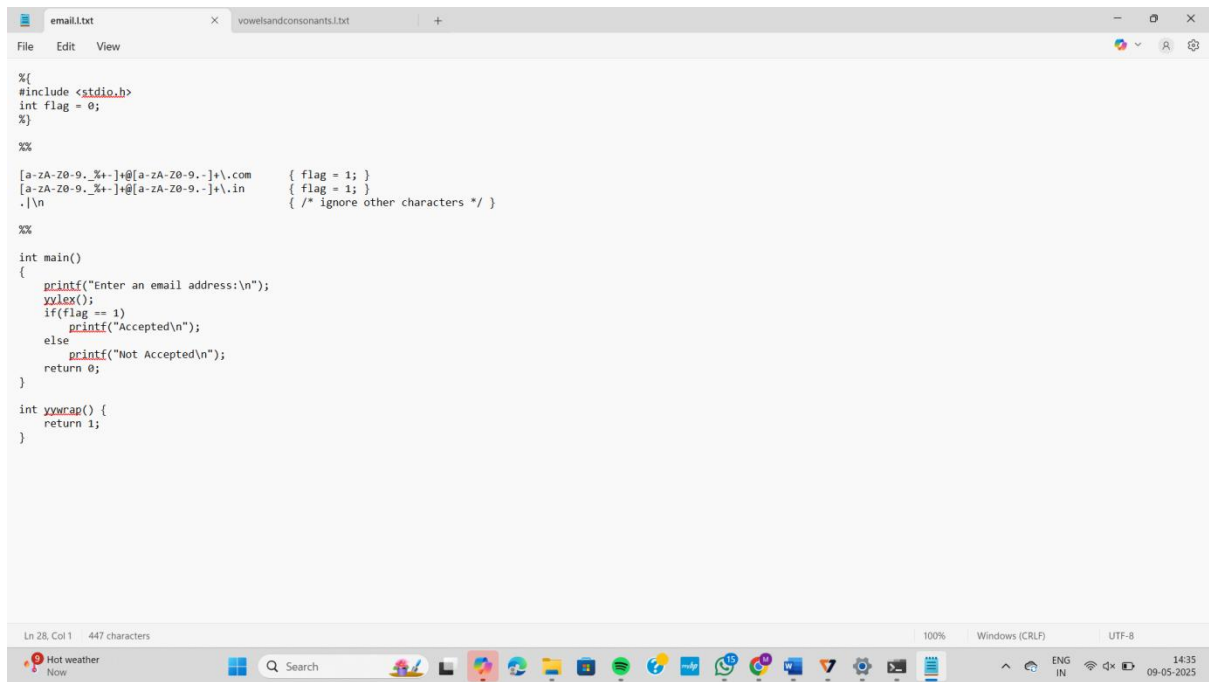


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venkataramana\OneDrive\Desktop\compiler design>set path=C:\GnuWin32\bin
C:\Users\venkataramana\OneDrive\Desktop\compiler design>flex abc.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
abcdefghiabc
ABCdefghiABC
```

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

Program:

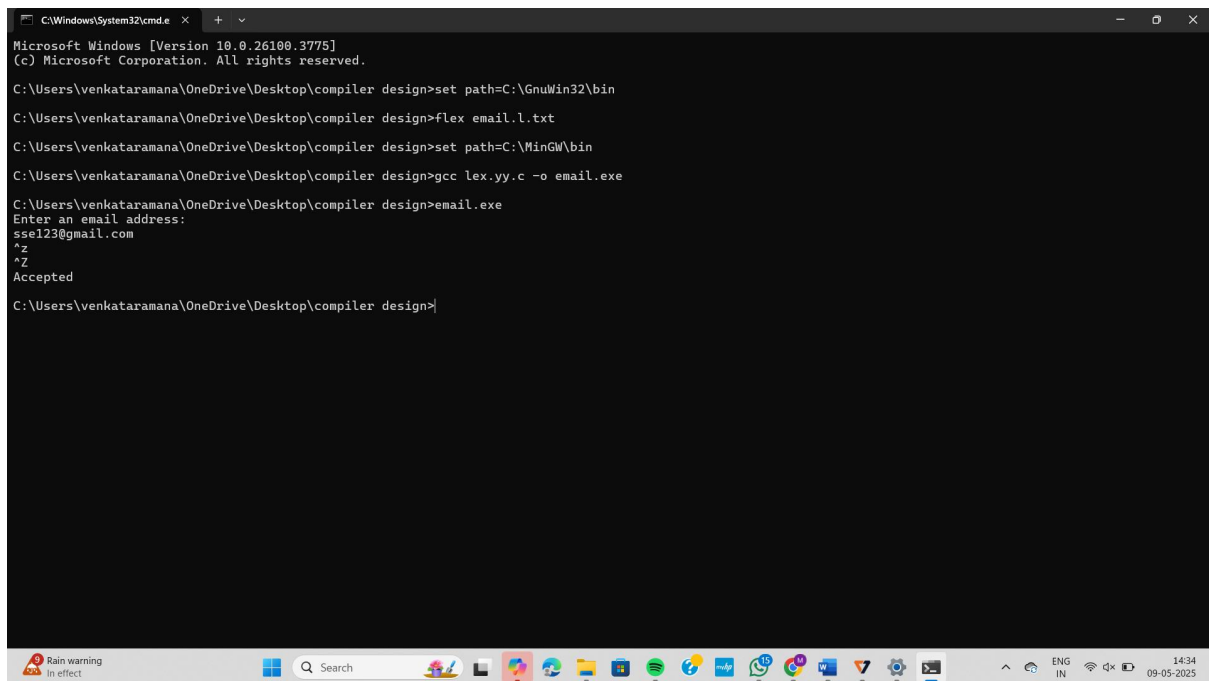


The screenshot shows a text editor with two tabs: 'email.txt' and 'vowelsandconsonants.txt'. The 'email.txt' tab is active, displaying a C program. The program includes `<stdio.h>` and defines a flag. It uses two regular expressions to validate an email address: `[a-zA-Z0-9_%.+]*@[a-zA-Z0-9.-]+\.` for the domain and `[a-zA-Z0-9_%.+]*@[a-zA-Z0-9.-]+\.` for the local part. The program prompts the user to enter an email address and prints 'Accepted' or 'Not Accepted' based on the validation. The status bar at the bottom indicates 'Ln 28, Col 1' and '447 characters'.

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

%%
[a-zA-Z0-9_%.+]*@[a-zA-Z0-9.-]+\.
```

Output:



The screenshot shows a Windows command prompt window. The user sets the path to the GCC compiler and runs the program. The program prompts for an email address, and the user enters 'snel23@gmail.com'. The program outputs 'Accepted'.

```
C:\Windows\System32\cmd.exe
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:
snel23@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:

```
cap.l.txt
File Edit View

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

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

Ln 1, Col 1 156 characters 100% Windows (CRLF) UTF-8

37°C Sunny

Output:

```
C:\Windows\System32\cmd.exe
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
File Edit View

%{
int com=0;
%}
%% 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;
}

Ln 1, Col 1 419 characters 100% Windows (CRLF) UTF-8
37°C Sunny
```

Output:

```
C:\Windows\System32\cmd.exe
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>|

37°C Sunny
```