CSA1447-COMPILER DESIGN FOR SYNTAX SMITH

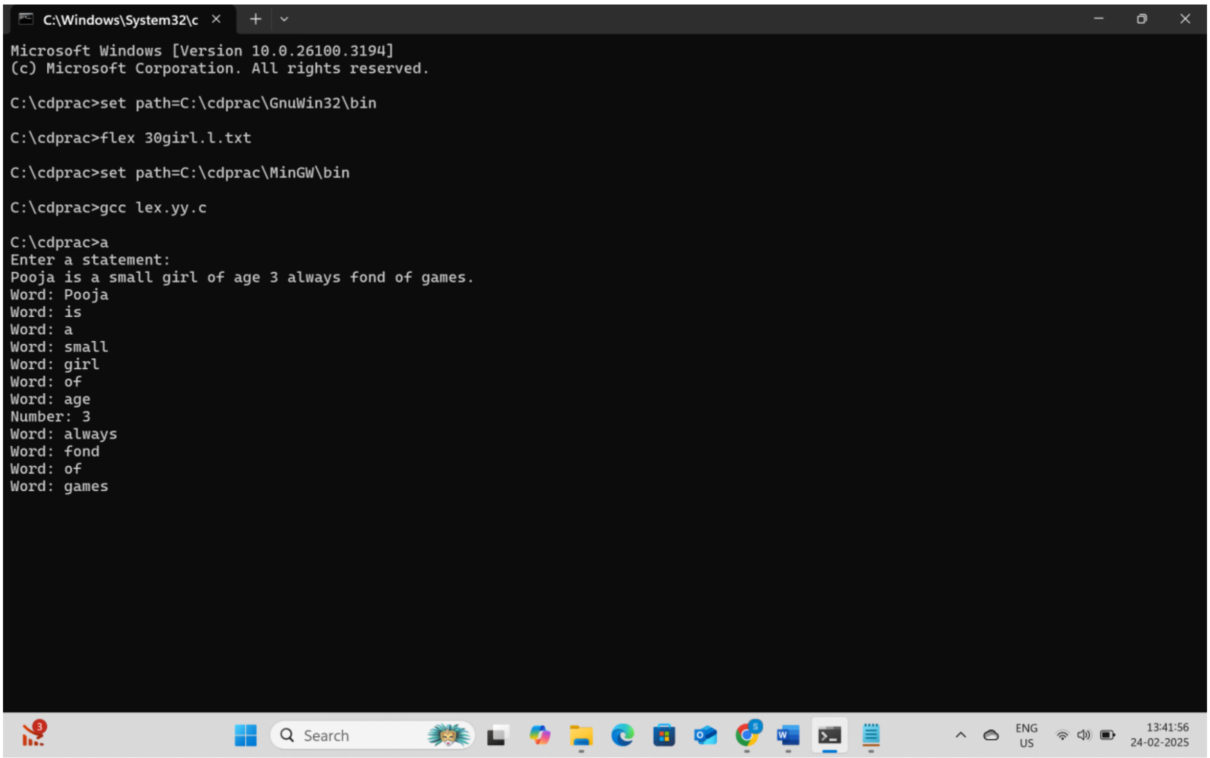
EXPERIMENT 30-40

Exp-30

PROGRAM:

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

int words = 0, numbers = 0;
%}

%%

[0-9]+      { printf("Number: %s\n", yytext); numbers++; }
[a-zA-Z]+   { printf("Word: %s\n", yytext); words++; }
[ \t\n]+    ;
.           ;

%%

int main() {
   printf("Enter a statement: \n");
   yylex();
   printf("\nTotal words: %d\n", words);
   printf("Total numbers: %d\n", numbers);
   return 0;
}

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 30girl.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a statement:
Pooja is a small girl of age 3 always fond of games.
Word: Pooja
Word: is
Word: a
Word: small
Word: girl
Word: of
Word: age
Number: 3
Word: always
Word: fond
Word: of
Word: games
```

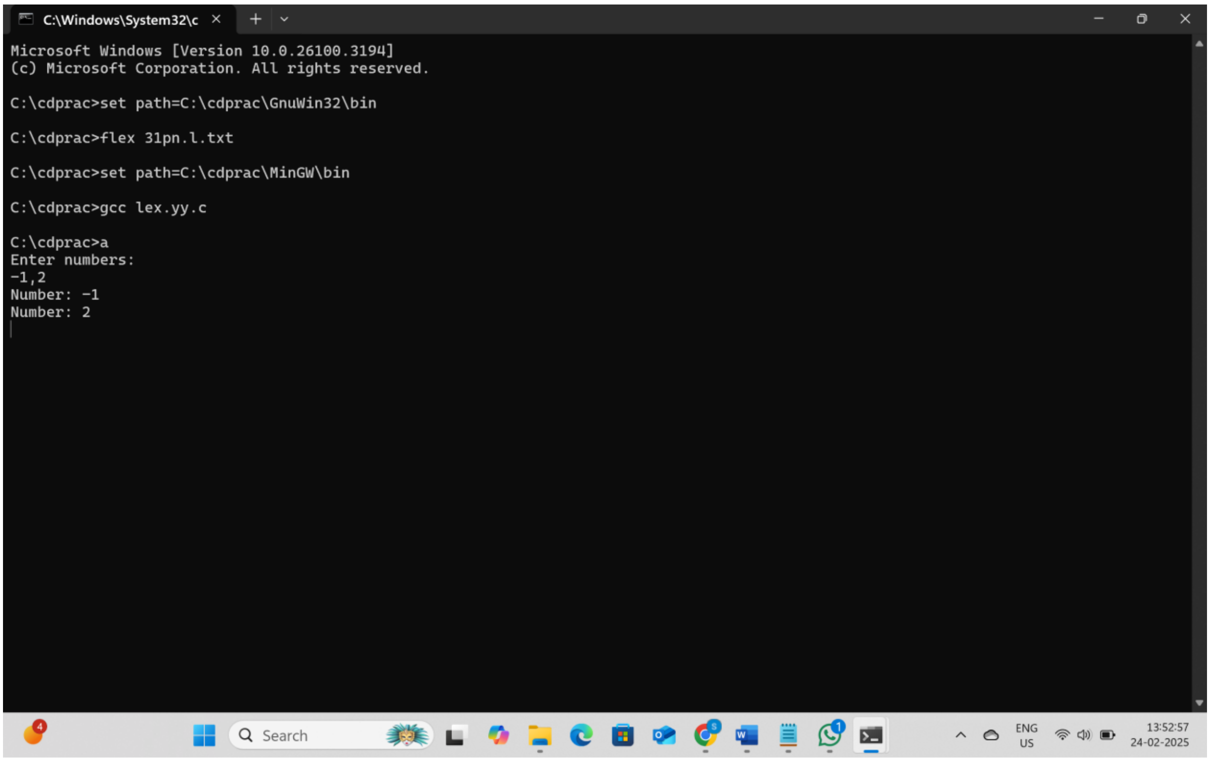
Exp-31

PROGRAM:

```
%{
#include <stdio.h>
int positive_count = 0, negative_count = 0;
%}

%%

-?[0-9]+  {
    if (yytext[0] == '-')
        negative_count++;
    else
        positive_count++;
    printf("Number: %s\n", yytext);
}

[ \t\n]+  ;
```

```
.       ;

%%

int main() {
    printf("Enter numbers: \n");
    yylex();
    printf("\nTotal Positive Numbers: %d\n", positive_count);
    printf("Total Negative Numbers: %d\n", negative_count);
    return 0;
}

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

OUTPUT:



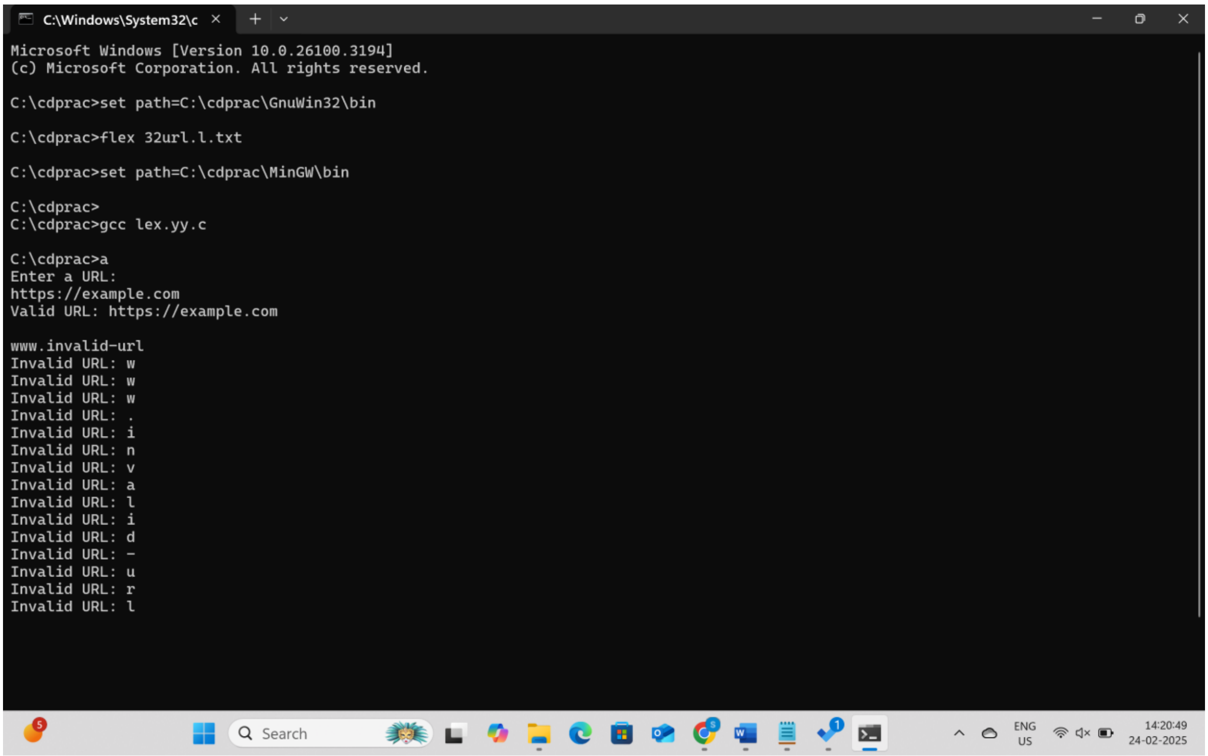
```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 31pn.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter numbers:
-1,2
Number: -1
Number: 2
```

Exp-32

PROGRAM:

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

%%

^https?:\/\/[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}(\/[a-zA-Z0-9._?=/-]*)?$   { printf("Valid URL: %s\n", yytext); }
.                                                       { printf("Invalid URL: %s\n", yytext); }

%%

int main() {
    printf("Enter a URL:\n");
    yylex();
    return 0;
}

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 32url.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a URL:
https://example.com
Valid URL: https://example.com

www.invalid-url
Invalid URL: w
Invalid URL: w
Invalid URL: w
Invalid URL: .
Invalid URL: i
Invalid URL: n
Invalid URL: v
Invalid URL: a
Invalid URL: l
Invalid URL: i
Invalid URL: d
Invalid URL: -
Invalid URL: u
Invalid URL: r
Invalid URL: l
```

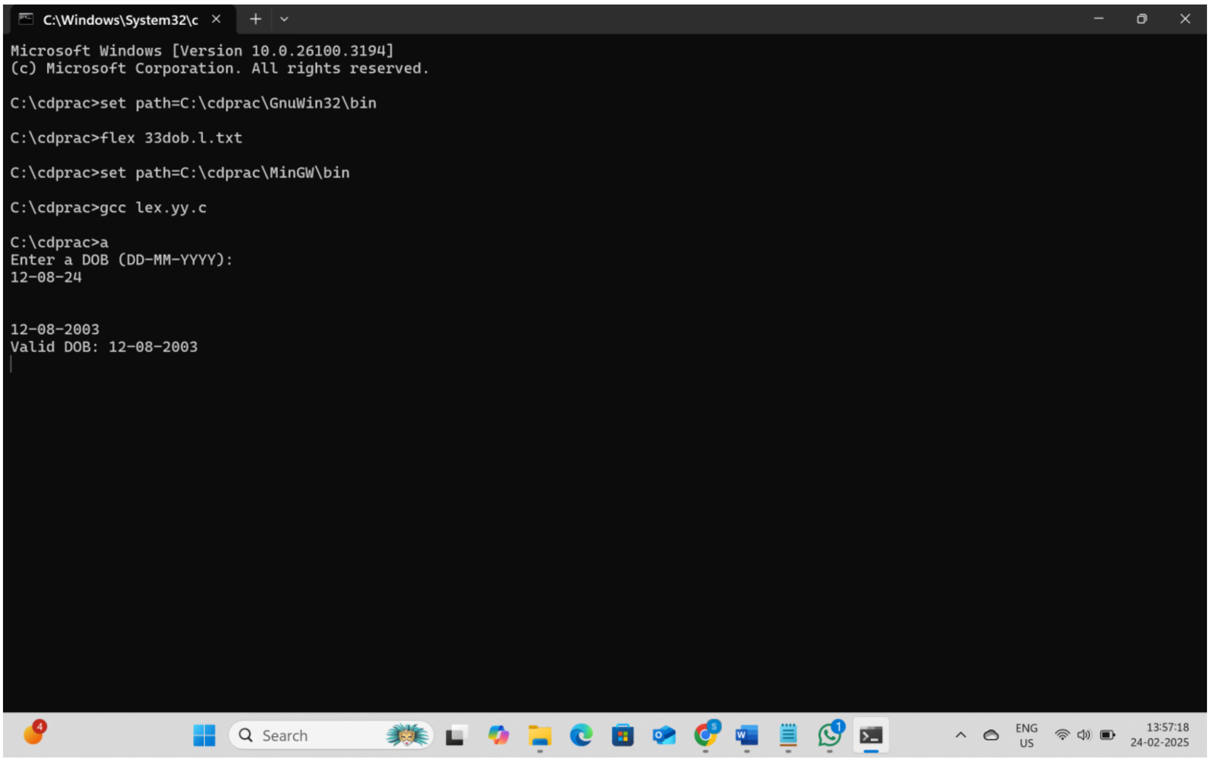
Exp-33

PROGRAM:

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


%%


(0[1-9]|[12][0-9]|3[01])[-/.](0[1-9]|1[0-2])[-/.](19|20)[0-9]{2}  { printf("Valid DOB: %s\n", yytext);
}
.|\n                                                    ;


%%


int main() {
    printf("Enter a DOB (DD-MM-YYYY): \n");
    yylex();
    return 0;
}
```

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 33dob.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a DOB (DD-MM-YYYY):
12-08-24


12-08-2003
Valid DOB: 12-08-2003
```

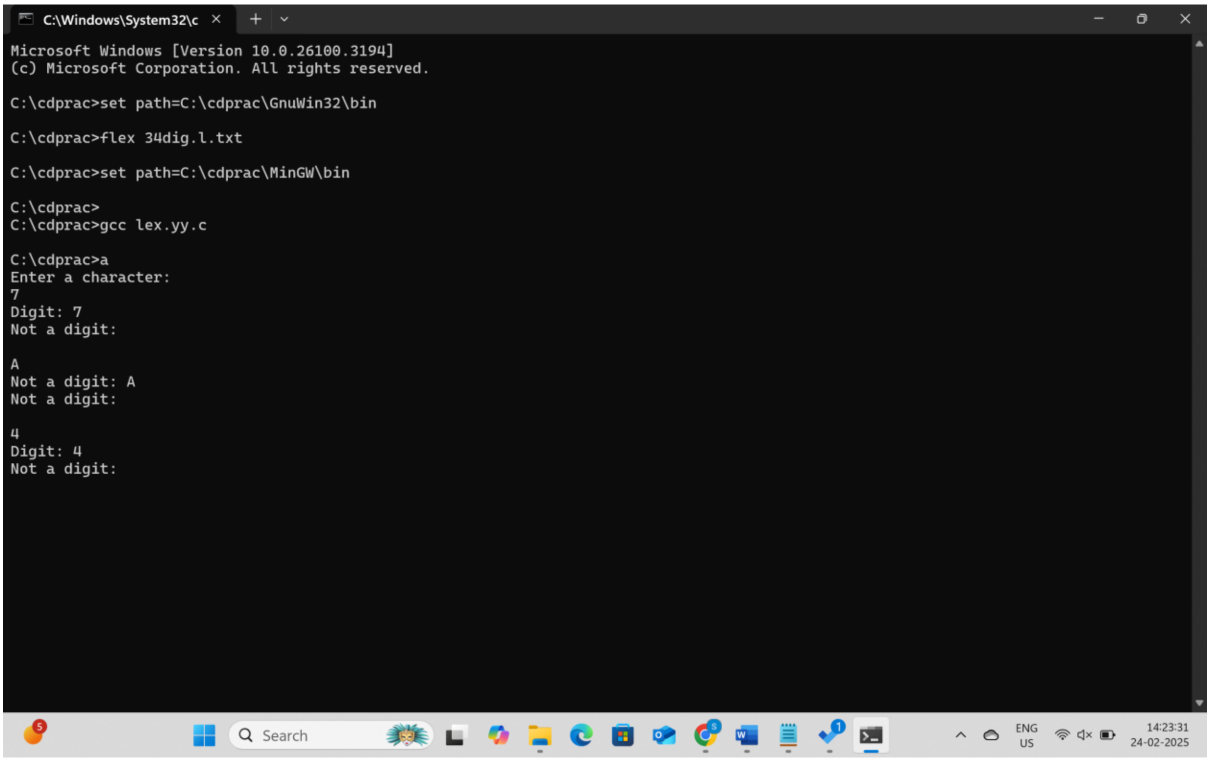
Exp-34

PROGRAM:

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

%%

[0-9]   { printf("Digit: %s\n", yytext); }
[^0-9]  { printf("Not a digit: %s\n", yytext); }

%%
```

```
int main() {
    printf("Enter a character:\n");
    yylex();
    return 0;
}

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 34dig.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a character:
7
Digit: 7
Not a digit:

A
Not a digit: A
Not a digit:

4
Digit: 4
Not a digit:
```

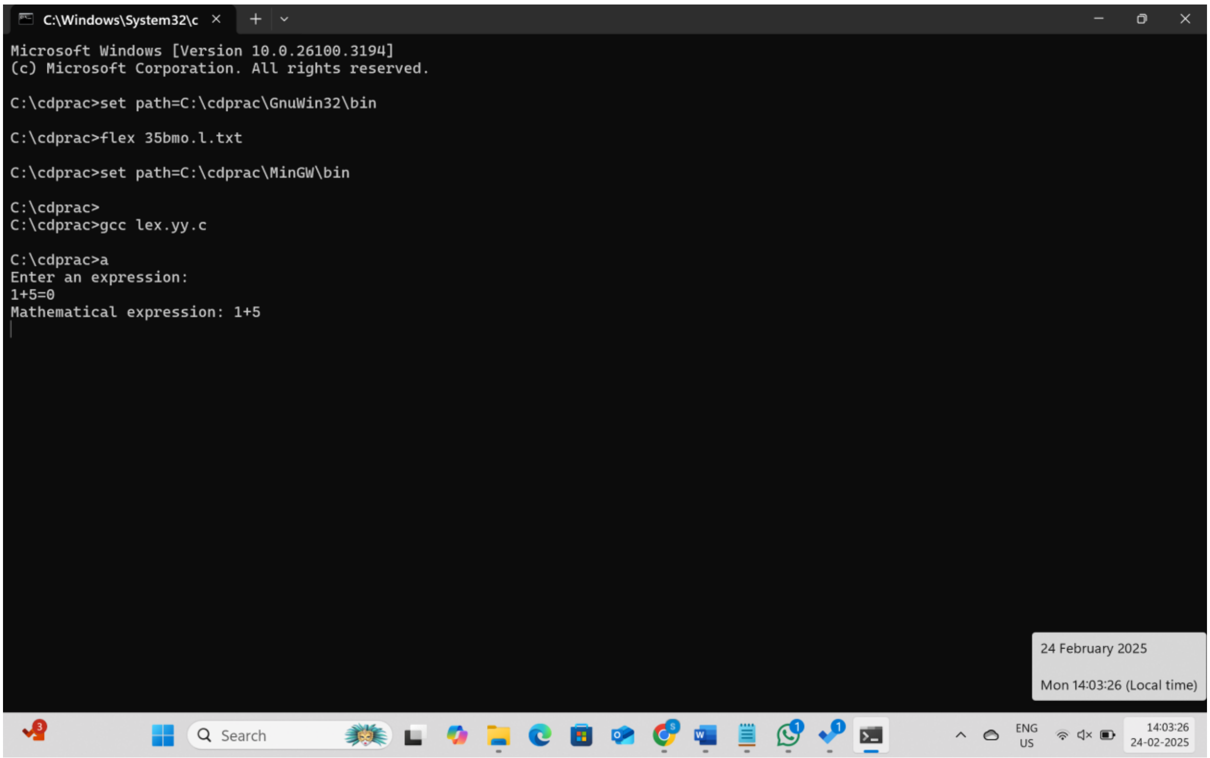
Exp-35

PROGRAM:

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


%%


[0-9]+[+\-*/][0-9]+  { printf("Mathematical expression: %s\n", yytext); }
```

```
.|\n                ;

%%

int main() {
    printf("Enter an expression: \n");
    yylex();
    return 0;
}

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

OUTPUT:



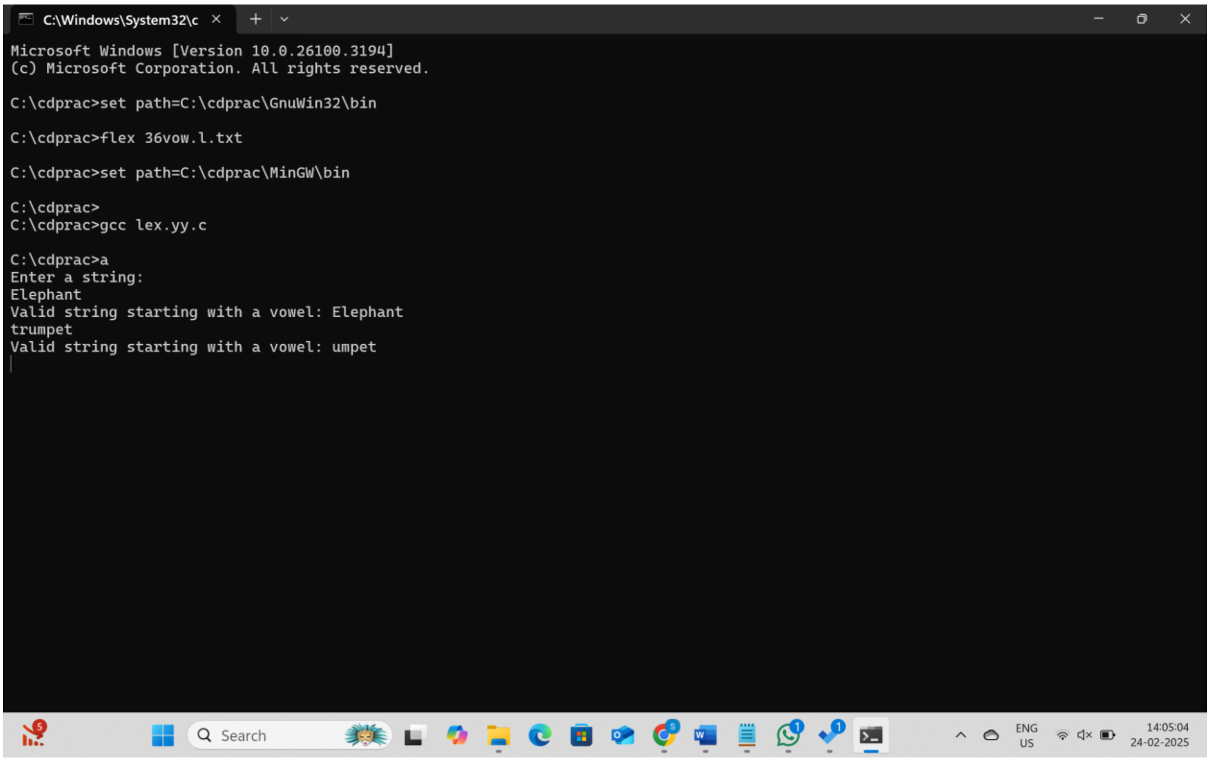
```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 35bmo.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter an expression:
1+5=0
Mathematical expression: 1+5
```

Exp-36

PROGRAM:

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

%%

[aeiouAEIOU][a-zA-Z]*  { printf("Valid string starting with a vowel: %s\n", yytext); }
.|\n                   ;

%%

int main() {
    printf("Enter a string: \n");
    yylex();
    return 0;
}

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

OUTPUT:



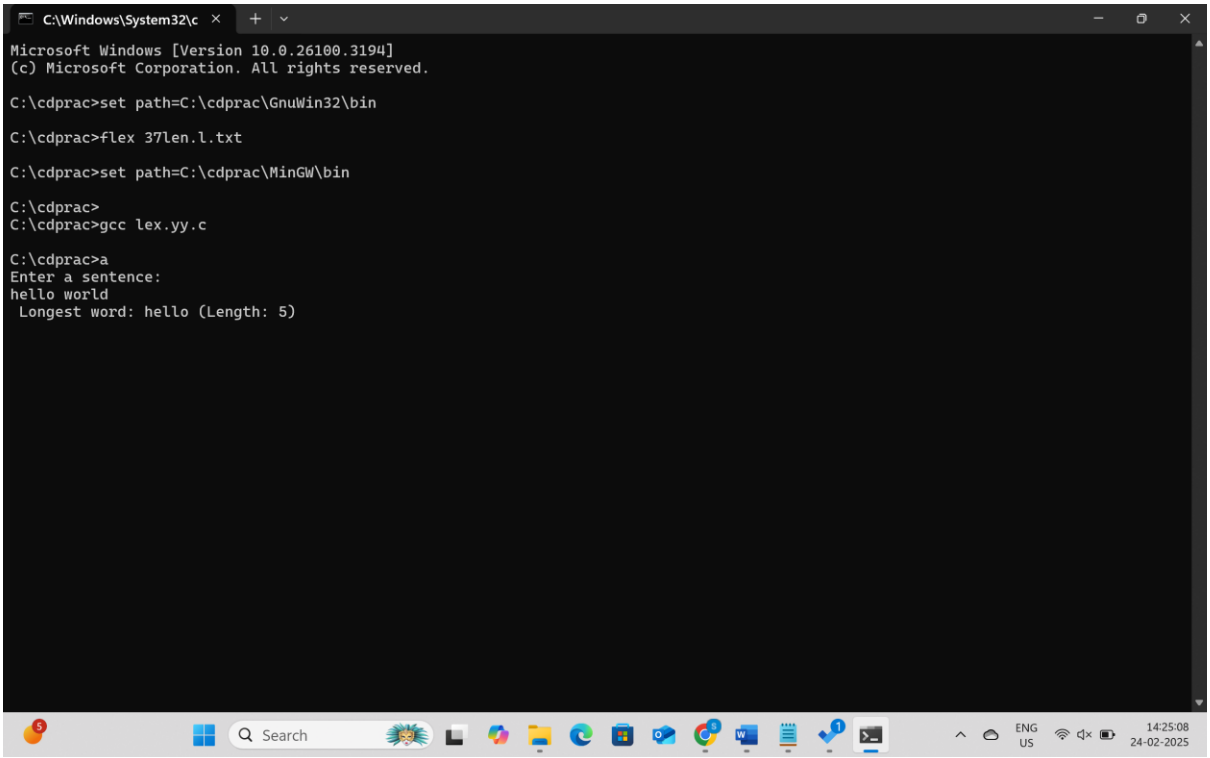
```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 36vow.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a string:
Elephant
Valid string starting with a vowel: Elephant
trumpet
Valid string starting with a vowel: umpet
```

Exp-37

PROGRAM:

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

int max_length = 0;
char longest_word[100];

%}

%%

[a-zA-Z]+ {
    int len = strlen(yytext);
    if (len > max_length) {
        max_length = len;
        strcpy(longest_word, yytext);
    }
}

\n {
    printf("Longest word: %s (Length: %d)\n", longest_word, max_length);
}

%%

int main() {
    printf("Enter a sentence:\n");
    yylex();
    return 0;
}

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 37len.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a sentence:
hello world
 Longest word: hello (Length: 5)
```

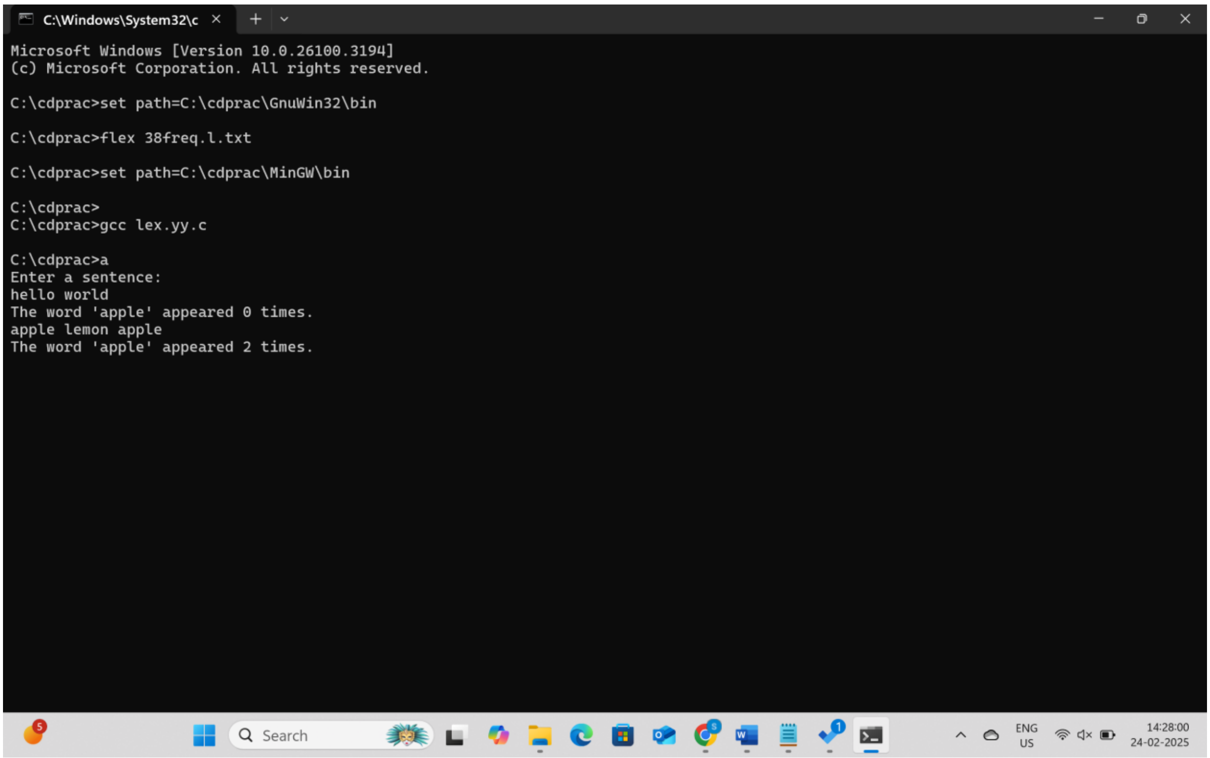
Exp-38

PROGRAM:

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

int count = 0;
char target[] = "apple"; // Change to any word you want to count

%}

%%
```

```
apple   { count++; }

\n      {
    printf("The word '%s' appeared %d times.\n", target, count);
}

.|\t    ;

%%

int main() {
    printf("Enter a sentence:\n");
    yylex();
    return 0;
}

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 38freq.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a sentence:
hello world
The word 'apple' appeared 0 times.
apple lemon apple
The word 'apple' appeared 2 times.
```

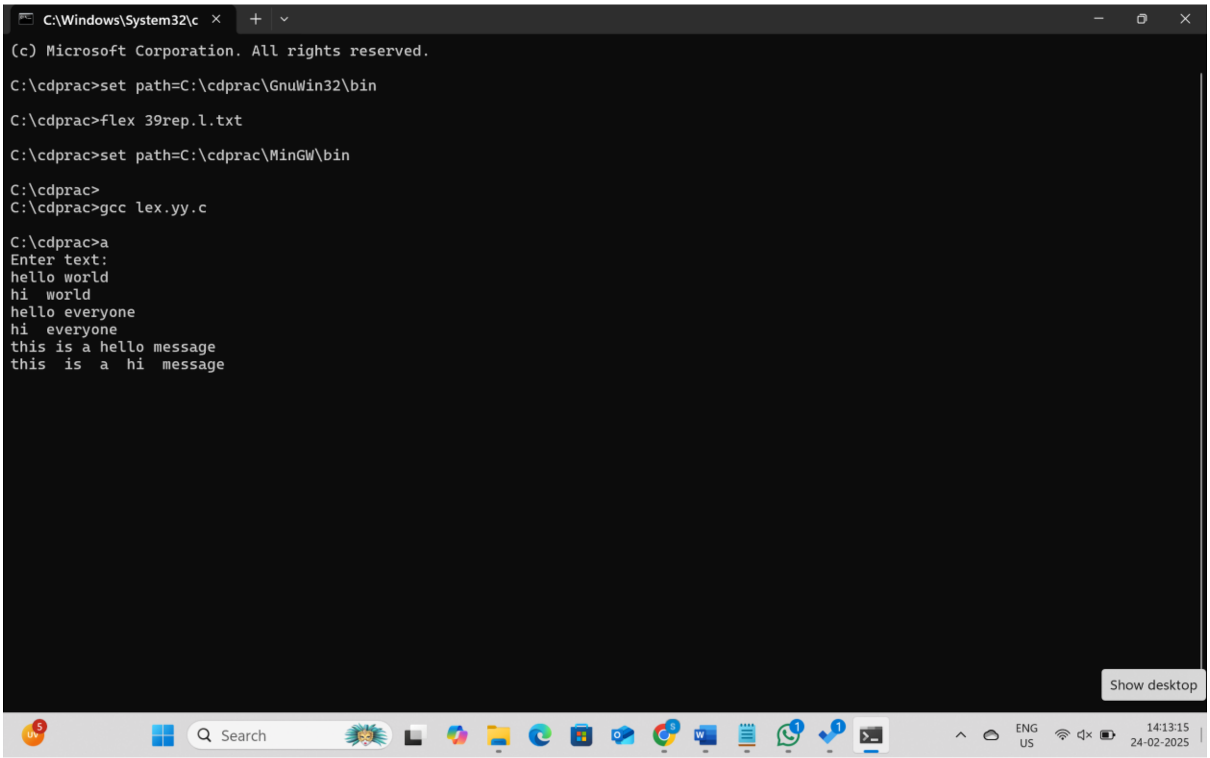
Exp-39

PROGRAM:

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

#define OLD_WORD "hello"
#define NEW_WORD "hi"

%}

%%

hello  { printf("%s ", NEW_WORD); }
[a-zA-Z]+  { printf("%s ", yytext); }
[ \t\n]    { printf("%s", yytext); }

%%
```

```
int main() {
    printf("Enter text: \n");
    yylex();
    return 0;
}

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

OUTPUT:



```
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 39rep.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter text:
hello world
hi  world
hello everyone
hi  everyone
this is a hello message
this  is  a  hi  message
```

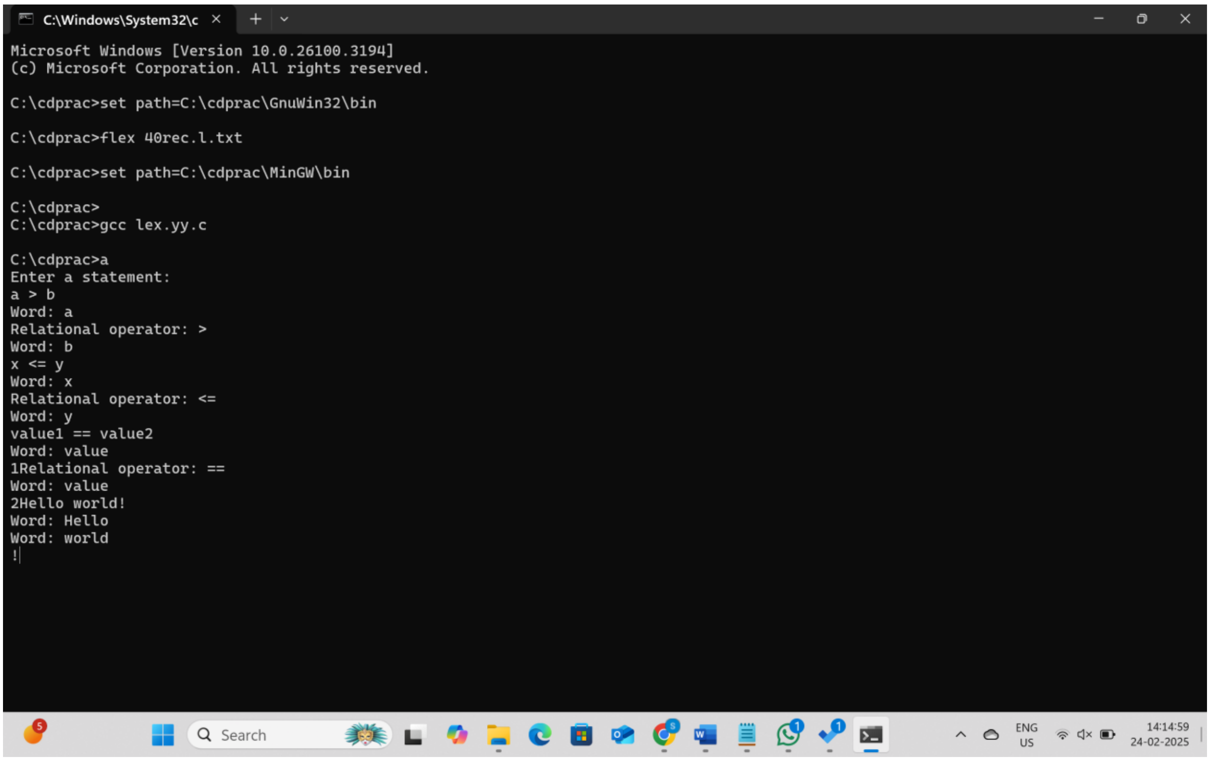
Exp-40

PROGRAM:

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

%%

(<=|>=|==|!=|<|>)  { printf("Relational operator: %s\n", yytext); }
```

```
[a-zA-Z]+        { printf("Word: %s\n", yytext); }
[ \t\n]          ;

%%

int main() {
    printf("Enter a statement: \n");
    yylex();
    return 0;
}

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

OUTPUT:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\cdprac>set path=C:\cdprac\GnuWin32\bin

C:\cdprac>flex 40rec.l.txt

C:\cdprac>set path=C:\cdprac\MinGW\bin

C:\cdprac>
C:\cdprac>gcc lex.yy.c

C:\cdprac>a
Enter a statement:
a > b
Word: a
Relational operator: >
Word: b
x <= y
Word: x
Relational operator: <=
Word: y
value1 == value2
Word: value
1Relational operator: ==
Word: value
2Hello world!
Word: Hello
Word: world
!
```