

## CSA1428 - Compiler Design

## LAB ACTIVITY-5

**1. Write a LEX specification file to take input C program from a .c file and count the number of characters, number of lines & number of words.**

**Input Source Program: (sample5.c)**

```
#include <stdio.h>

int main()
{
    int number1, number2, sum;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);
    sum = number1 + number2;
    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
```

**Code (Lex):**

```
%{
    int nchar, nword, nline;
}%
%%

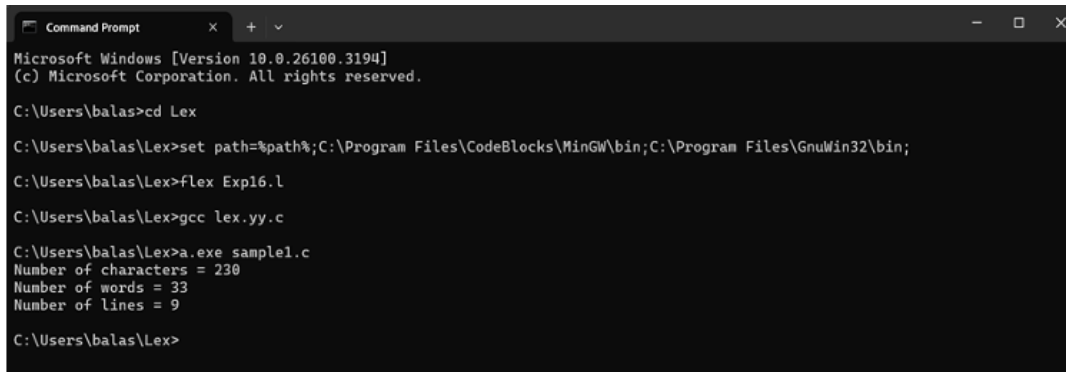
\n { nline++; nchar++; }
[^ \t\n]+ { nword++; nchar += yyleng; }
. { nchar++; }
%%

int yywrap(void) {
    return 1;
}

int main(int argc, char *argv[]) {
    yyin = fopen(argv[1], "r");
    yylex();
    printf("Number of characters = %d\n", nchar);
}
```

```
printf("Number of words = %d\n", nword);
printf("Number of lines = %d\n", nline);
fclose(yyin);
}
```

### Output:



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

C:\Users\balas>cd Lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp16.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe sample1.c
Number of characters = 230
Number of words = 33
Number of lines = 9
C:\Users\balas\Lex>
```

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

### Input Source File: (sample6.c)

```
#include<stdio.h>

int main()
{
int a,b,c; /*variable declaration*/
printf("enter two numbers");
scanf("%d %d",&a,&b);
c=a+b;//adding two numbers
printf("sum is %d",c);
return 0;
}
```

### Code (Lex):

```
%{
int com=0;
}%
%s COMMENT
%%
"/*" {BEGIN COMMENT;}
<COMMENT>"*/" {BEGIN 0; com++;}
```

```

<COMMENT>\n {com++;}

<COMMENT>. {;}

\\.* {; com++;}

.\n {fprintf(yyout,"%s",yytext);}

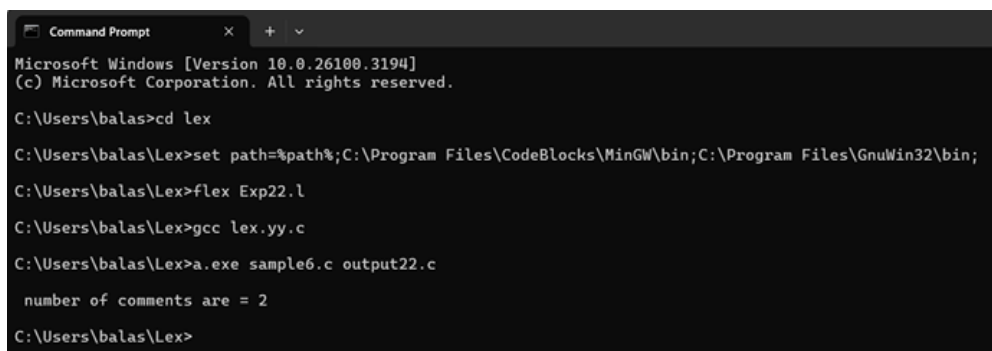
%%

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:



```

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

C:\Users\balas>cd lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp22.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe sample6.c output22.c

number of comments are = 2
C:\Users\balas\Lex>

```

### Output File (output22.c):

```
output22.c
1  #include<stdio.h>
2  int main()
3  {
4      int a,b,c;
5      printf("Enter two numbers ");
6      scanf("%d %d",&a,&b);
7      c=a+b;
8      printf("sum is %d",c);
9      return 0;
10 }
```

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

### Code (Lex):

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

int main()
{
printf("Enter String :\n");
yylex();
}

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

### Output:

```
Command Prompt - a.exe
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

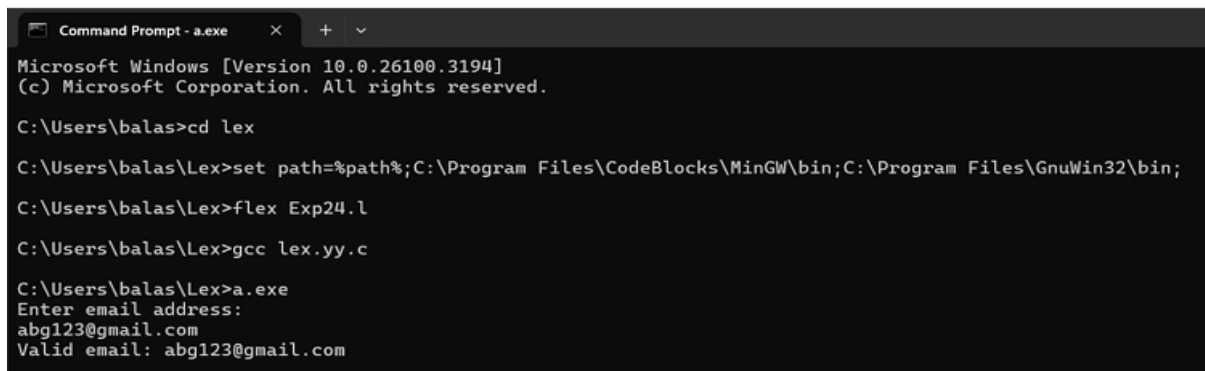
C:\Users\balas>cd lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp23.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe
Enter String :
CAPITAL of INDIA is DELHI
CAPITAL is a capital word
INDIA is a capital word
DELHI
is a capital word
```

#### 4. Write a LEX Program to check the email address is valid or not.

##### Code (Lex):

```
%{  
#include <stdio.h>  
#include <string.h>  
%}  
%%  
[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,} { printf("Valid email: %s\n", yytext); }  
.  
%%  
int main() {  
printf("Enter email address: \n");  
yylex();  
return 0;  
}  
int yywrap() {  
return 1;  
}
```

##### Output:



```
Command Prompt - a.exe
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\balas>cd lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp24.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe
Enter email address:
abg123@gmail.com
Valid email: abg123@gmail.com
```

#### 5. Write a LEX Program to convert the substring abc to ABC from the given input string.

##### Code (Lex):

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

```

abc { printf("ABC"); }

. { printf("%s", yytext); }

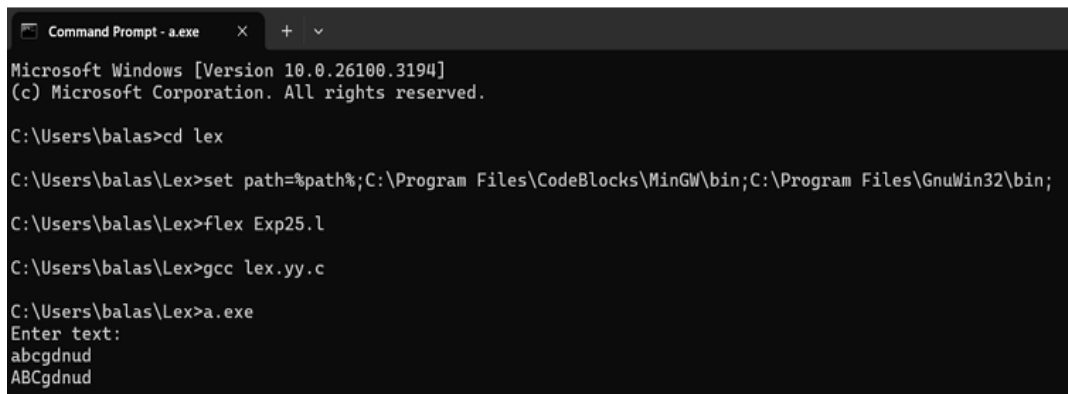
%%

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

int yywrap() {
return 1;
}

```

### Output:



```

Command Prompt - a.exe
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\balas>cd lex

C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;

C:\Users\balas\Lex>flex Exp25.l

C:\Users\balas\Lex>gcc lex.yy.c

C:\Users\balas\Lex>a.exe
Enter text:
abcgdnud
ABCgdnud

```

**6. The Company ABC runs with employees with several departments. The Organization manager had all the mobile numbers of employees. Assume that you are the manager and need to verify the valid mobile numbers because there may be some invalid numbers present. Implement a LEX program to check whether the mobile number is valid or not.**

### Code (Lex):

```

%%

[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:

```

Command Prompt - a.exe
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\balas>cd lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp26.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe
Enter Mobile Number : 8787454567
Mobile Number Valid

```

**7. Implement Lexical Analyzer using LEX or FLEX (Fast Lexical Analyzer). The program should separate the tokens in the given C program and display with appropriate caption.**

### Input Source Program: (sample7.c)

```

#include<stdio.h>

void main()

{

int a,b,c = 30;

printf("hello");

}

```

### Code (Lex):

```

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 { }
%%

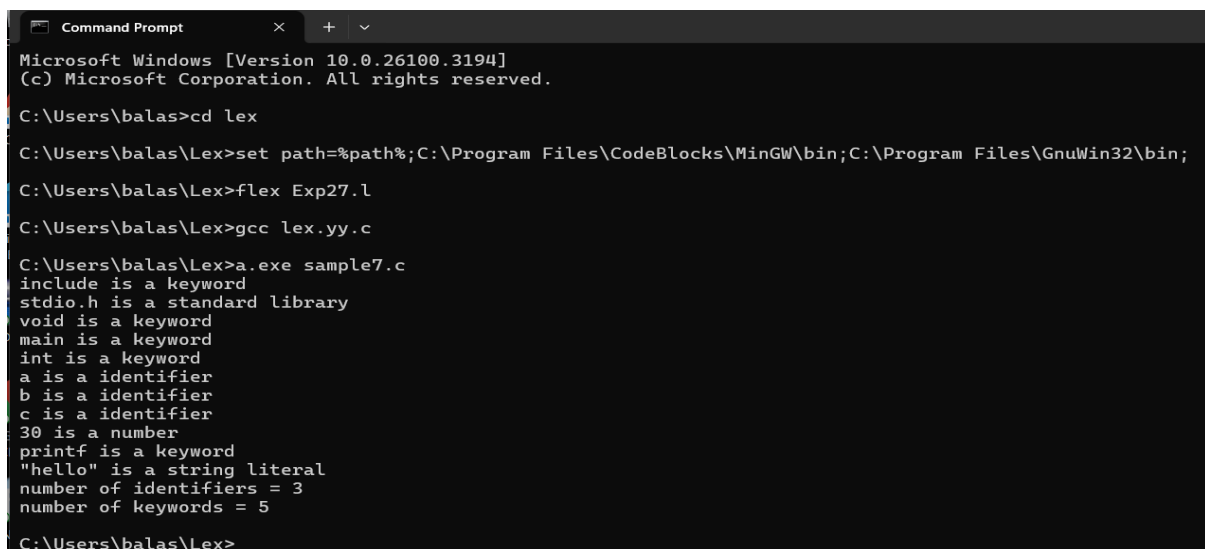
int yywrap(void) {
return 1;
}

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 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\balas>cd lex

C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;

C:\Users\balas\Lex>flex Exp27.l

C:\Users\balas\Lex>gcc lex.yy.c

C:\Users\balas\Lex>a.exe sample7.c
include is a keyword
stdio.h is a standard library
void is a keyword
main is a keyword
int is a keyword
a is a identifier
b is a identifier
c is a identifier
30 is a number
printf is a keyword
"hello" is a string literal
number of identifiers = 3
number of keywords = 5

C:\Users\balas\Lex>

```

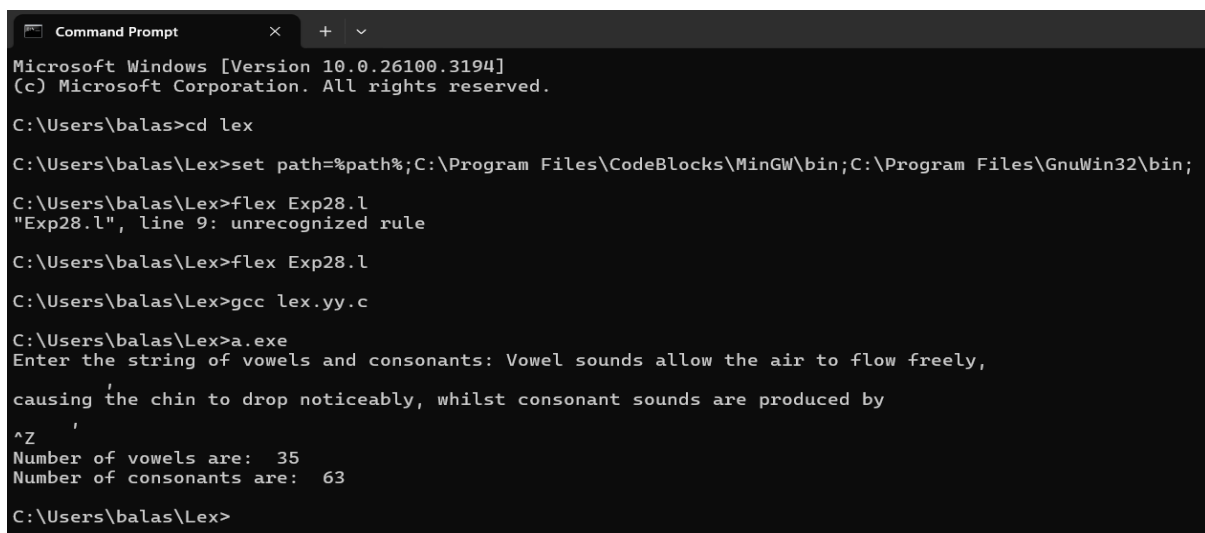
**8. In a class, an English teacher was teaching the vowels and consonants to the students. She says “Vowel sounds allow the air to flow freely, causing the chin to drop noticeably, whilst consonant sounds are produced by restricting the air flow”. As a class activity the students are asked to identify the vowels and consonants in the given word/sentence and count the number of elements in each. Write an algorithm to help the student to count the number of vowels and consonants in the given sentence.**



### Code (Lex):

```
%{  
    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 10.0.26100.3194]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\balas>cd lex  
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;  
C:\Users\balas\Lex>flex Exp28.l  
"Exp28.l", line 9: unrecognized rule  
C:\Users\balas\Lex>flex Exp28.l  
C:\Users\balas\Lex>gcc lex.yy.c  
C:\Users\balas\Lex>a.exe  
Enter the string of vowels and consonants: Vowel sounds allow the air to flow freely,  
causing the chin to drop noticeably, whilst consonant sounds are produced by  
^Z  
Number of vowels are: 35  
Number of consonants are: 63  
C:\Users\balas\Lex>
```

**9. Keywords are predefined, reserved words used in programming that have special meanings to the compiler. Keywords are part of the syntax and they cannot be used as an identifier. In general there are 32 keywords. The prime function of Lexical Analyser is token Generation. Among the 6 types of tokens, differentiating Keyword and Identifier is a challenging issue. Thus write a LEX program to separate keywords and identifiers.**

**Input Source File(sample8.c):**

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,b,c = 30;
```

```
printf("hello");
```

```
}
```

**Code (Lex):**

```
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:**

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

C:\Users\balas>cd lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp29.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe sample9.c
include is a keyword
stdio.h is a standard library
void is a keyword
main is a keyword
int is a keyword
a is a identifier
b is a identifier
c is a identifier
30 is a number
printf is a keyword
"hello" is a string literal
number of identifiers = 3
number of keywords = 5
```

**10. Write a LEX program to recognise numbers and words in a statement. Pooja is a small girl of age 3 always fond of games. Due to the pandemic, she was not allowed to play outside. So her mother designs a gaming event by showing a flash card. Pooja has to separate the numbers in one list and words in another list shown in the flash card.**

**Code (Lex):**

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <ctype.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] ; /* Ignore whitespace */
. ; /* Ignore other characters */
%%
int main() {
    printf("Enter a statement:\n");
```

```
yylex();

printf("\nTotal Words: %d\n", words);

printf("Total Numbers: %d\n", numbers);

return 0;

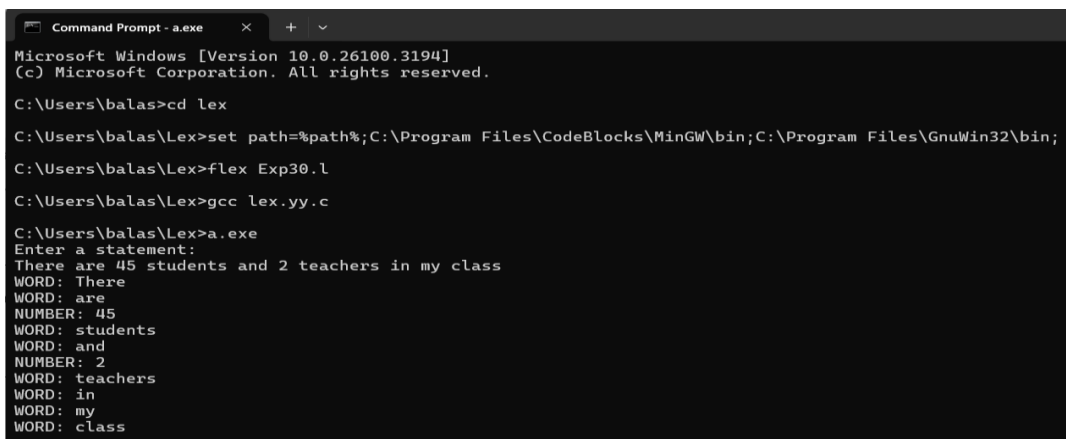
}

int yywrap() {

    return 1;

}
```

### Output:



```
Command Prompt - a.exe
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\balas>cd lex
C:\Users\balas\Lex>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;
C:\Users\balas\Lex>flex Exp30.l
C:\Users\balas\Lex>gcc lex.yy.c
C:\Users\balas\Lex>a.exe
Enter a statement:
There are 45 students and 2 teachers in my class
WORD: There
WORD: are
NUMBER: 45
WORD: students
WORD: and
NUMBER: 2
WORD: teachers
WORD: in
WORD: my
WORD: class
```