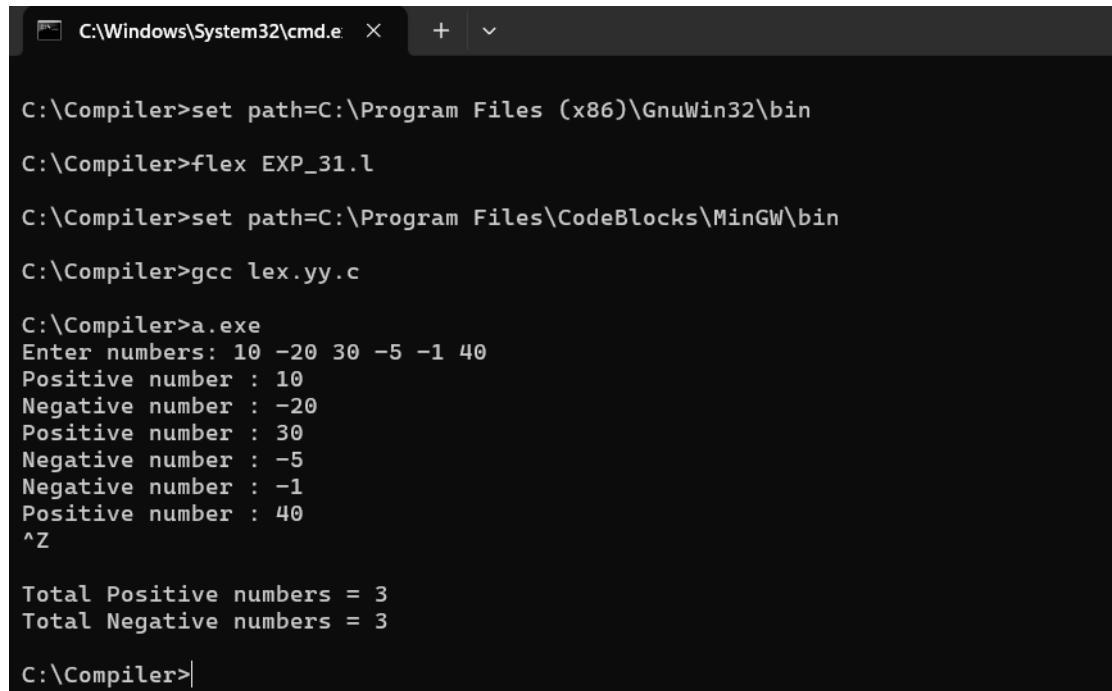


CODE

```
%{  
#include <stdio.h>  
int pos=0, neg=0;  
%}  
  
%%  
[-][0-9]+      { neg++; printf("Negative number : %s\n", yytext); }  
[0-9]+         { pos++; printf("Positive number : %s\n", yytext); }  
[ \t\n]+        ; /* ignore spaces */  
.             ; /* ignore anything else */  
%%  
  
int yywrap() { return 1; }  
  
int main() {  
    printf("Enter numbers: ");  
    yylex();  
    printf("\nTotal Positive numbers = %d", pos);  
    printf("\nTotal Negative numbers = %d\n", neg);  
    return 0;  
}
```

SCREENSHOT



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.e'. The user has set the path to include GnuWin32 and MinGW bin directories. They run 'flex EXP_31.l' to generate a lexer, then 'gcc lex.yy.c' to compile it into 'a.exe'. When they enter '10 -20 30 -5 -1 40' at the prompt, the program counts them as follows:

```
C:\Compiler>set path=C:\Program Files (x86)\GnuWin32\bin  
C:\Compiler>flex EXP_31.l  
C:\Compiler>set path=C:\Program Files\CodeBlocks\MinGW\bin  
C:\Compiler>gcc lex.yy.c  
C:\Compiler>a.exe  
Enter numbers: 10 -20 30 -5 -1 40  
Positive number : 10  
Negative number : -20  
Positive number : 30  
Negative number : -5  
Negative number : -1  
Positive number : 40  
^Z  
  
Total Positive numbers = 3  
Total Negative numbers = 3  
C:\Compiler>
```