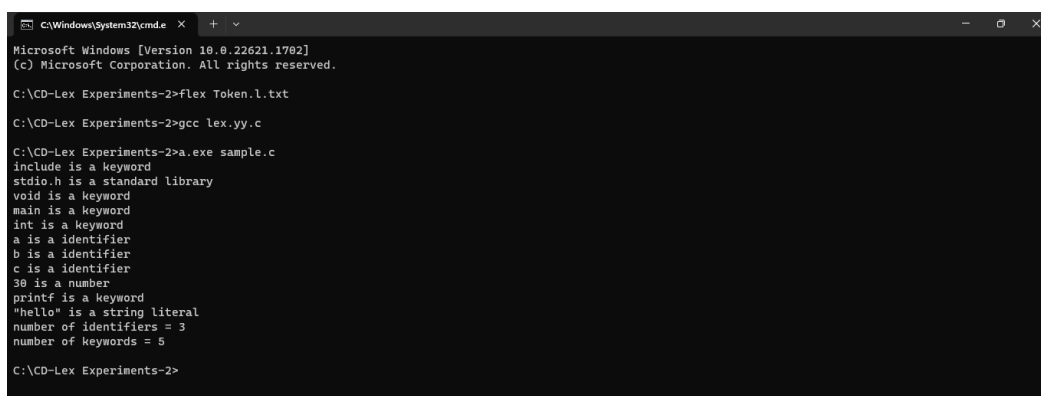


## 31&34:TOKENS LEX PROGRAM:

```
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:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\CD-Lex Experiments-2>flex Token.l.txt
C:\CD-Lex Experiments-2>gcc lex.yy.c
C:\CD-Lex Experiments-2>a.exe sample.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:\CD-Lex Experiments-2>
```

