Experiment -27

Develop a lexical Analyzer to count comment lines and remove them in a given c file using lexical program.

Sample C Program:

```
#include <stdio.h>
// this is a single comment
int main() {
   /* multi
      line
      comment */
   printf("Hello");
}
```

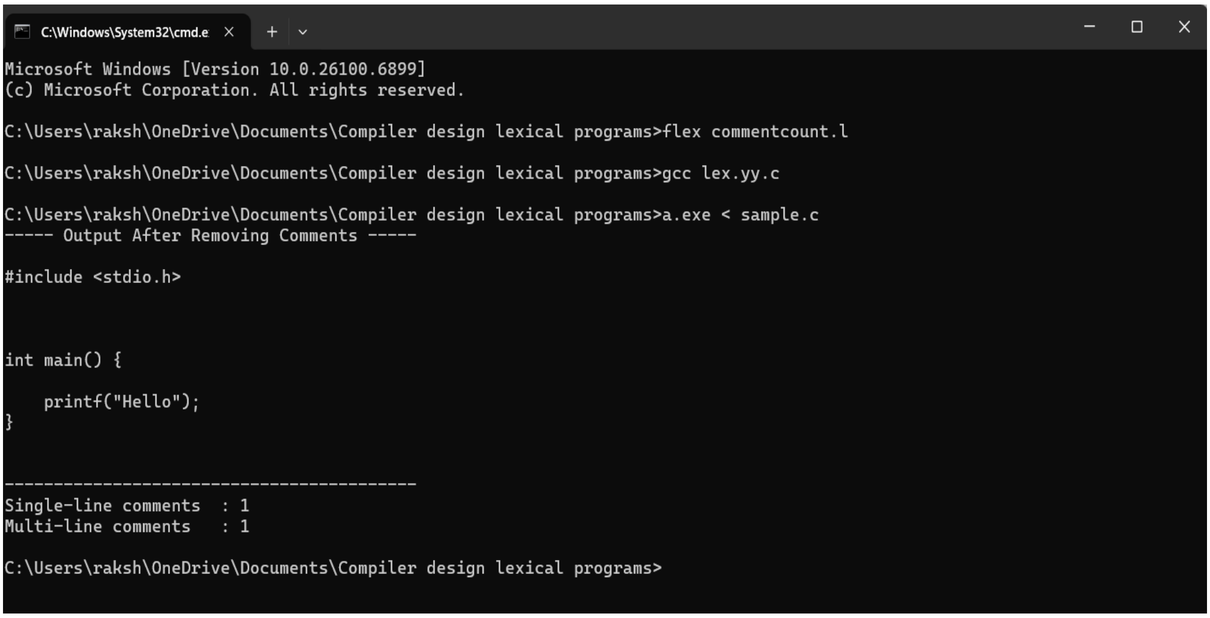
Program:

```
%{
#include <stdio.h>
int single = 0, multi = 0;
%}
%%
"//".*              { single++; /* skip single-line comment */ }
"/*"([^*]|\*+[^*/])*"\*/" {
                     multi++; /* skip multi-line comment */
                    }
.|\n                { putchar(yytext[0]); }   /* print non-comment text */
%%


int main() {
   printf("----- Output After Removing Comments -----\n\n");
   yylex();
   printf("\n\n------------------------------------------\n");
   printf("Single-line comments  : %d\n", single);
```

```
    printf("Multi-line comments   : %d\n", multi);
    return 0;
}
int yywrap() { return 1; }
```

Output:



```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>flex commentcount.l

C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>gcc lex.yy.c

C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>a.exe < sample.c
----- Output After Removing Comments -----

#include <stdio.h>



int main() {

    printf("Hello");
}


------------------------------------------
Single-line comments  : 1
Multi-line comments   : 1

C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>
```