Extend the lexical Analyzer to Check comments, defined as follows in C:

a) A comment begins with // and includes all characters until the end of that line.

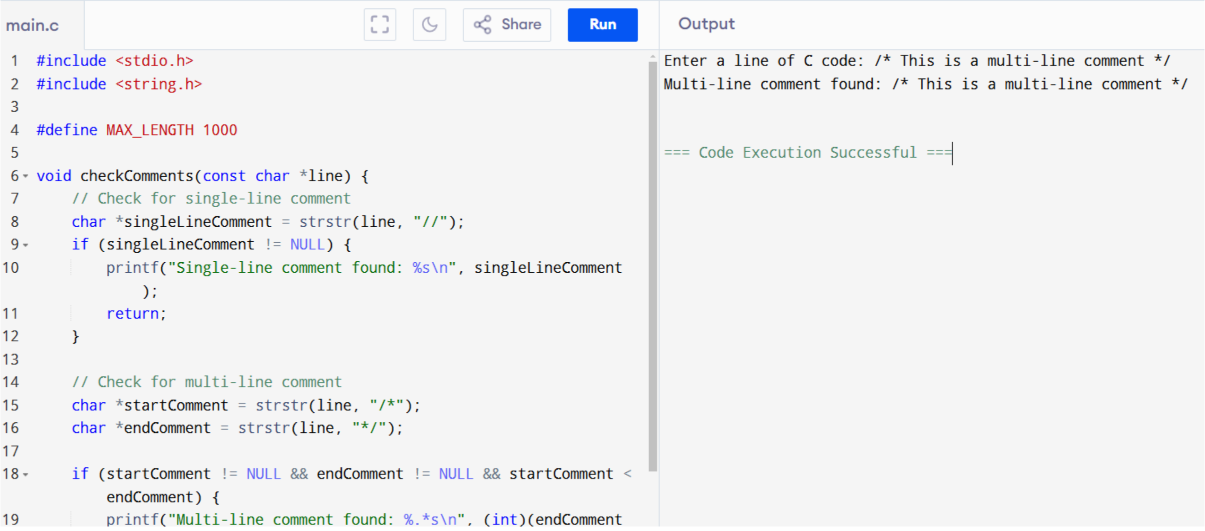
b) A comment begins with /* and includes all characters through the next occurrence of the character sequence */Develop a lexical Analyzer to identify whether a given line is a comment or not.

PGM:

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

#define MAX_LENGTH 1000

void checkComments(const char *line) {
   // Check for single-line comment
   char *singleLineComment = strstr(line, "//");
   if (singleLineComment != NULL) {
      printf("Single-line comment found: %s\n", singleLineComment);
      return;
   }

   // Check for multi-line comment
   char *startComment = strstr(line, "/*");
   char *endComment = strstr(line, "*/");

   if (startComment != NULL && endComment != NULL && startComment <
endComment) {
      printf("Multi-line comment found: %.*s\n", (int)(endComment - startComment + 2),
startComment);
      return;
   } else if (startComment != NULL) {
      printf("Multi-line comment starts but does not end in this line: %s\n", startComment);
```

```
        return;
    }

    // If no comments are found
    printf("No comments found in the line.\n");
}

int main() {
    char line[MAX_LENGTH];

    printf("Enter a line of C code: ");
    fgets(line, MAX_LENGTH, stdin);

    checkComments(line);

    return 0;
}
```

OUTPUT:



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

#define MAX_LENGTH 1000

void checkComments(const char *line) {
    // Check for single-line comment
    char *singleLineComment = strstr(line, "//");
    if (singleLineComment != NULL) {
        printf("Single-line comment found: %s\n", singleLineComment
            );
        return;
    }

    // Check for multi-line comment
    char *startComment = strstr(line, "/*");
    char *endComment = strstr(line, "*/");

    if (startComment != NULL && endComment != NULL && startComment <
        endComment) {
        printf("Multi-line comment found: %.*s\n", (int)(endComment
```

```
Enter a line of C code: /* This is a multi-line comment */
Multi-line comment found: /* This is a multi-line comment */


=== Code Execution Successful ===
```