**Department of CSE**

**Compiler Lab (CSE 352)**

**Lab Report 01**

Identifying Whether a Given Line is a Comment in C

**Submitted By :**

**Fahima Abida Chowdhury**

ID: 0432220005101135

Semester: Spring-2025

Batch: 52(6B1)

**Submitted To :**

[**Md. Ismail**](https://uits.edu.bd/md-ismail/)

Lecturer

Department of CSE

University of Information Technology and Sciences

**Experiment No.: 01**  
**Experiment Name:** Identifying Whether a Given Line is a Comment in C

**Problem Statement**

The objective of this lab is to write a C program that identifies whether a given line is a comment or not. Comments are non-executable parts of a C program that are used to improve code readability and documentation. There are two types of comments in C:

* **Single-line comments**, which begin with //
* **Multi-line comments**, which start with /\* and end with \*/

The program takes a line of input and checks whether it matches either of these comment formats.

**Implementation**

#include <stdio.h>

#include <string.h>

int main() {

char s[1000];

fgets(s, sizeof(s), stdin);

size\_t len = strlen(s);

if (len > 0 && s[len - 1] == '\n') {

s[len - 1] = '\0';

len--;

}

if (s[0] == '/' && s[1] == '/') {

printf("Yes, this is a single-line comment.\n");

}

else if (s[0] == '/' && s[1] == '\*' && s[len - 2] == '\*' && s[len - 1] == '/') {

printf("Yes, this is a multi-line comment.\n");

}

else {

printf("No comment exists here.\n");

}

return 0;

}

**Input**

This program reads a single line of text from the user (via standard input). The user is expected to enter a potential comment line.

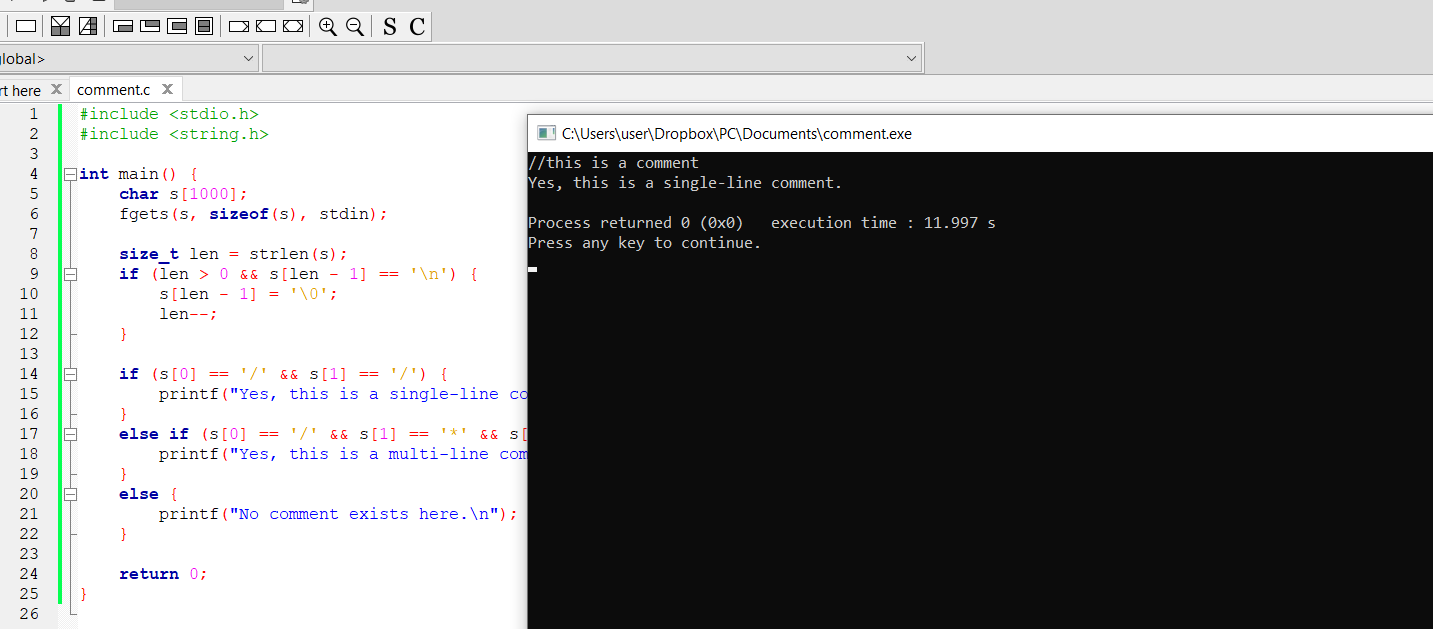
**Example Input 1:**

/\* This is a comment \*/

**Output**

The program prints one of the following messages based on the input:

* "Yes, this is a single-line comment."
* "Yes, this is a multi-line comment."
* "No comment exists here."



**Conclusion**

This lab experiment demonstrates how to detect C-style comments using a simple string-matching technique in C. The program successfully distinguishes between single-line and multi-line comments, and accurately identifies non-comment lines.

Understanding comment detection is useful not only for compiler design but also in creating tools like syntax highlighters, linters, or preprocessors. This exercise also reinforces the usage of character arrays, string manipulation, and conditional logic in C programming.