**Lab session 2nd question**

**2..** Extend the lexical Analyzer to Check comments, dened 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.

**Aim:**

To develop a lexical analyzer in C that checks if a given line is a comment in C, considering:

1. **Single-line comments**: Start with // and continue until the end of the line.
2. **Multi-line comments**: Start with /\* and end with \*/.

**Procedure:**

1. Read the input string (line of code).
2. Check if the string starts with // → **Single-line comment**.
3. Check if the string starts with /\* and ends with \*/ → **Multi-line comment**.
4. If neither condition is met, it's not a comment.

**Program:**

#include <stdio.h>

#include <string.h>

void check\_comment(char line[]) {

int length = strlen(line);

// Remove leading and trailing spaces

while (length > 0 && (line[length - 1] == ' ' || line[length - 1] == '\n'))

line[--length] = '\0';

while (\*line == ' ') line++; // Remove leading spaces

// Check for single-line comment

if (strncmp(line, "//", 2) == 0) {

printf("Single-line comment detected\n");

}

// Check for multi-line comment

else if (strncmp(line, "/\*", 2) == 0 && length >= 4 && strcmp(line + length - 2, "\*/") == 0) {

printf("Multi-line comment detected\n");

}

else {

printf("Not a comment\n");

}

}

int main() {

char line[200];

// Input from user

printf("Enter a line of C code: ");

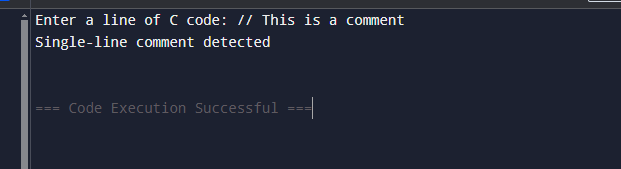
fgets(line, sizeof(line), stdin);

check\_comment(line);

return 0;

}

**Output**

****