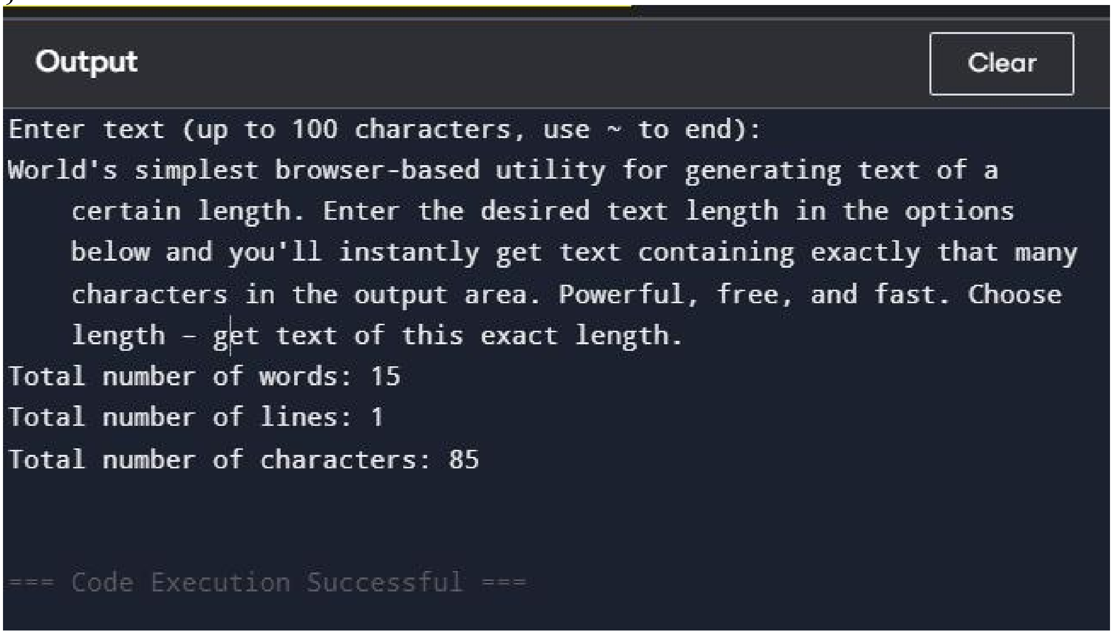
Exp. No. 5

Design a lexical Analyzer to find the number of whitespaces and newline characters using C.

Program:

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

int main() {
    char str[100];
    int words = 0, lines = 0, characters = 0;
    int inWord = 0; // Flag to track word boundaries

    printf("Enter text (up to 100 characters, use ~ to end):\n");
    fgets(str, sizeof(str), stdin); // Read input safely

    // Replace '~' with null terminator if present
    char *tildePos = strchr(str, '~');
    if (tildePos != NULL) {
        *tildePos = '\0';
    }

    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == ' ' || str[i] == '\t') {
            inWord = 0; // Word ended
        } else if (str[i] == '\n') {
            lines++;
            inWord = 0; // Word ended
        } else {
            characters++; // Count all non-space characters
            if (inWord == 0) {
                words++; // Start of a new word
                inWord = 1;
            }
        }
    }

    // Adjust for an empty input case
    if (characters > 0 && lines == 0) {
        lines = 1;
    }

    printf("Total number of words: %d\n", words);
    printf("Total number of lines: %d\n", lines);
    printf("Total number of characters: %d\n", characters);

    return 0;
}
```



```
Enter text (up to 100 characters, use ~ to end):
World's simplest browser-based utility for generating text of a
    certain length. Enter the desired text length in the options
    below and you'll instantly get text containing exactly that many
    characters in the output area. Powerful, free, and fast. Choose
    length - get text of this exact length.
Total number of words: 15
Total number of lines: 1
Total number of characters: 85


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