AIM : 7A. FIFOS: Full duplex communication between two independent processes. First process accepts sentences and writes on one pipe to be read by second process and second process counts number of characters, number of words and number of lines in accepted sentences, writes this output in a text file and writes the contents of the file on second pipe to be read by first process and displays on standard output.

```c
//LAB 7A – FIFOS.c

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

int main() {
    int pipe1[2], pipe2[2];
    pid_t pid;
    char inputSentence[100];

    if (pipe(pipe1) == -1 || pipe(pipe2) == -1) {
        perror("Pipe failed");
        exit(EXIT_FAILURE);
    }

    pid = fork();

    if (pid < 0) {
        perror("Fork failed");
        exit(EXIT_FAILURE);
    }

    if (pid > 0) {
        // Parent process (Sender)
        close(pipe1[0]); // Close reading end of pipe1
        close(pipe2[1]); // Close writing end of pipe2

        printf("Enter a sentence: ");
        fgets(inputSentence, sizeof(inputSentence), stdin);

        write(pipe1[1], inputSentence, strlen(inputSentence) + 1);
        close(pipe1[1]); // Close writing end of pipe1

        char output[100];
        read(pipe2[0], output, sizeof(output));
        printf("Received from process 2: %s\n", output);
        close(pipe2[0]); // Close reading end of pipe2
    } else {
        // Child process (Receiver)
        close(pipe1[1]); // Close writing end of pipe1
        close(pipe2[0]); // Close reading end of pipe2

        char receivedSentence[100];
        read(pipe1[0], receivedSentence, sizeof(receivedSentence));
        close(pipe1[0]); // Close reading end of pipe1

        // Count characters, words, and lines
        int charCount = 0, wordCount = 0, lineCount = 0;
        char* token = strtok(receivedSentence, " \n");
        while (token != NULL) {
            wordCount++;
            charCount += strlen(token);
            token = strtok(NULL, " \n");
        }

        for (int i = 0; receivedSentence[i] != '\0'; i++) {
            if (receivedSentence[i] == '\n') {
                lineCount++;
            }
        }

        // Write output to a text file
        FILE* outputFile = fopen("output.txt", "w");
        fprintf(outputFile, "Character Count: %d\n", charCount);
        fprintf(outputFile, "Word Count: %d\n", wordCount);
        fprintf(outputFile, "Line Count: %d\n", lineCount);
        fclose(outputFile);

        // Write contents of the file to pipe2
        outputFile = fopen("output.txt", "r");
        char output[100];
        fgets(output, sizeof(output), outputFile);
        write(pipe2[1], output, strlen(output) + 1);
        fclose(outputFile);
        close(pipe2[1]); // Close writing end of pipe2
    }

    return 0;
}
```