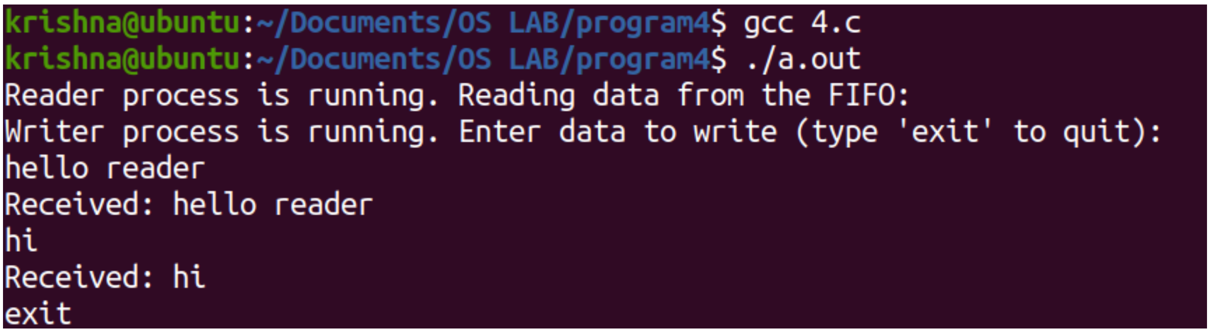
Program 4. Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>

#define FIFO_NAME "myfifo"

int main() {
    int fd;
    char buffer[BUFSIZ];

    // Create the FIFO (named pipe) if it doesn't already exist
    if (mkfifo(FIFO_NAME, 0666) == -1) {
        perror("mkfifo");
        exit(EXIT_FAILURE);
    }

    // Writer process
    if (fork() == 0) {
        printf("Writer process is running. Enter data to write (type 'exit' to quit):\n");

        fd = open(FIFO_NAME, O_WRONLY);
        if (fd == -1) {
            perror("open");
            exit(EXIT_FAILURE);
        }

        while (1) {
            fgets(buffer, BUFSIZ, stdin);

            if (strcmp(buffer, "exit\n") == 0) {
                break;
            }

            write(fd, buffer, strlen(buffer) + 1);
        }

        close(fd);
    }
    // Reader process
    else {
        printf("Reader process is running. Reading data from the FIFO:\n");

        fd = open(FIFO_NAME, O_RDONLY);
        if (fd == -1) {
            perror("open");
            exit(EXIT_FAILURE);
        }

        while (1) {
            if (read(fd, buffer, BUFSIZ) == 0) {
                break;
            }

            printf("Received: %s", buffer);
        }

        close(fd);
    }

    return 0;
}
```



```
krishna@ubuntu:~/Documents/OS LAB/program4$ gcc 4.c
krishna@ubuntu:~/Documents/OS LAB/program4$ ./a.out
Reader process is running. Reading data from the FIFO:
Writer process is running. Enter data to write (type 'exit' to quit):
hello reader
Received: hello reader
hi
Received: hi
exit
```