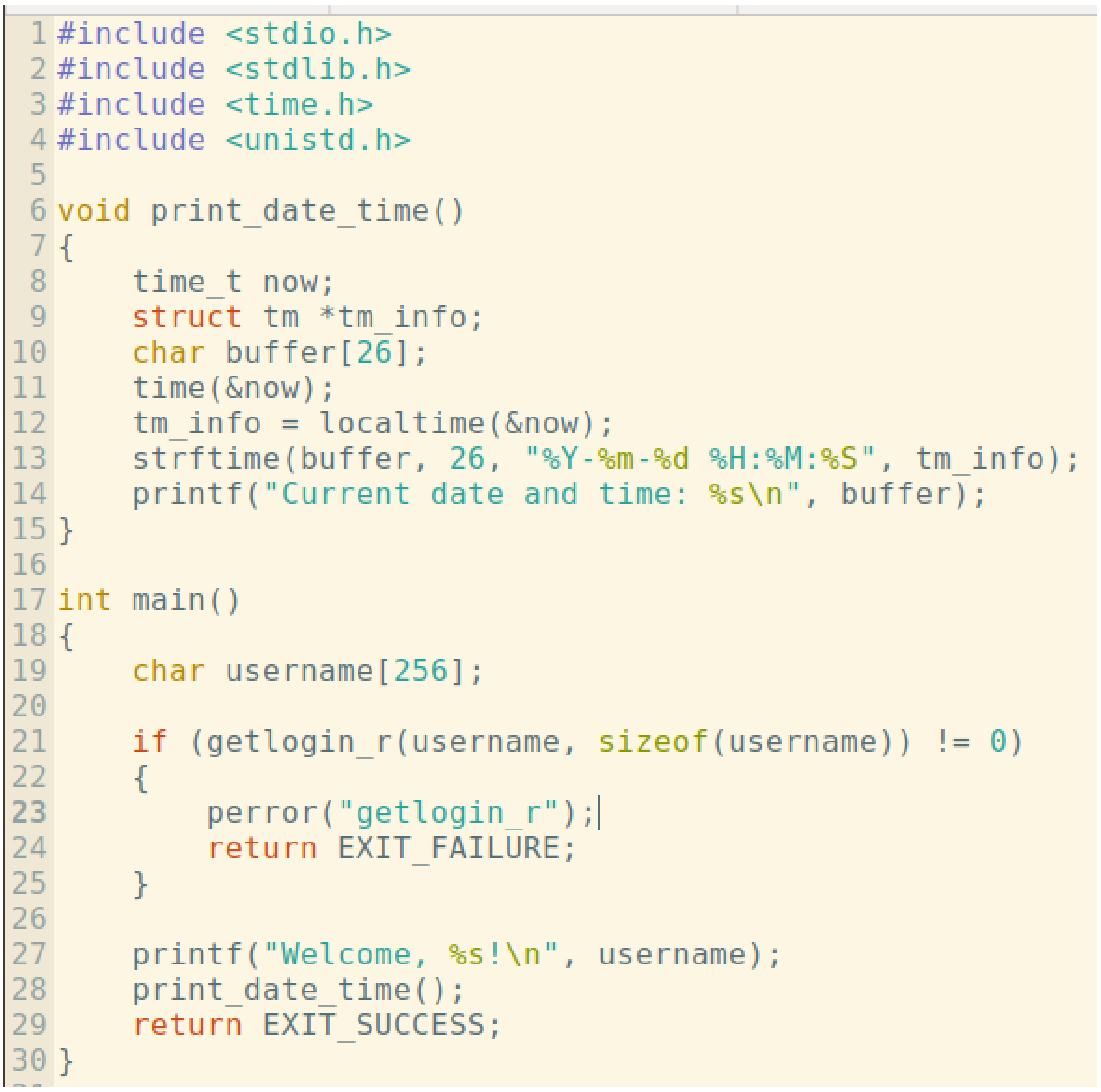
Operating System Lab 4 | Assignment 4
Arnav Samal
122CS0107

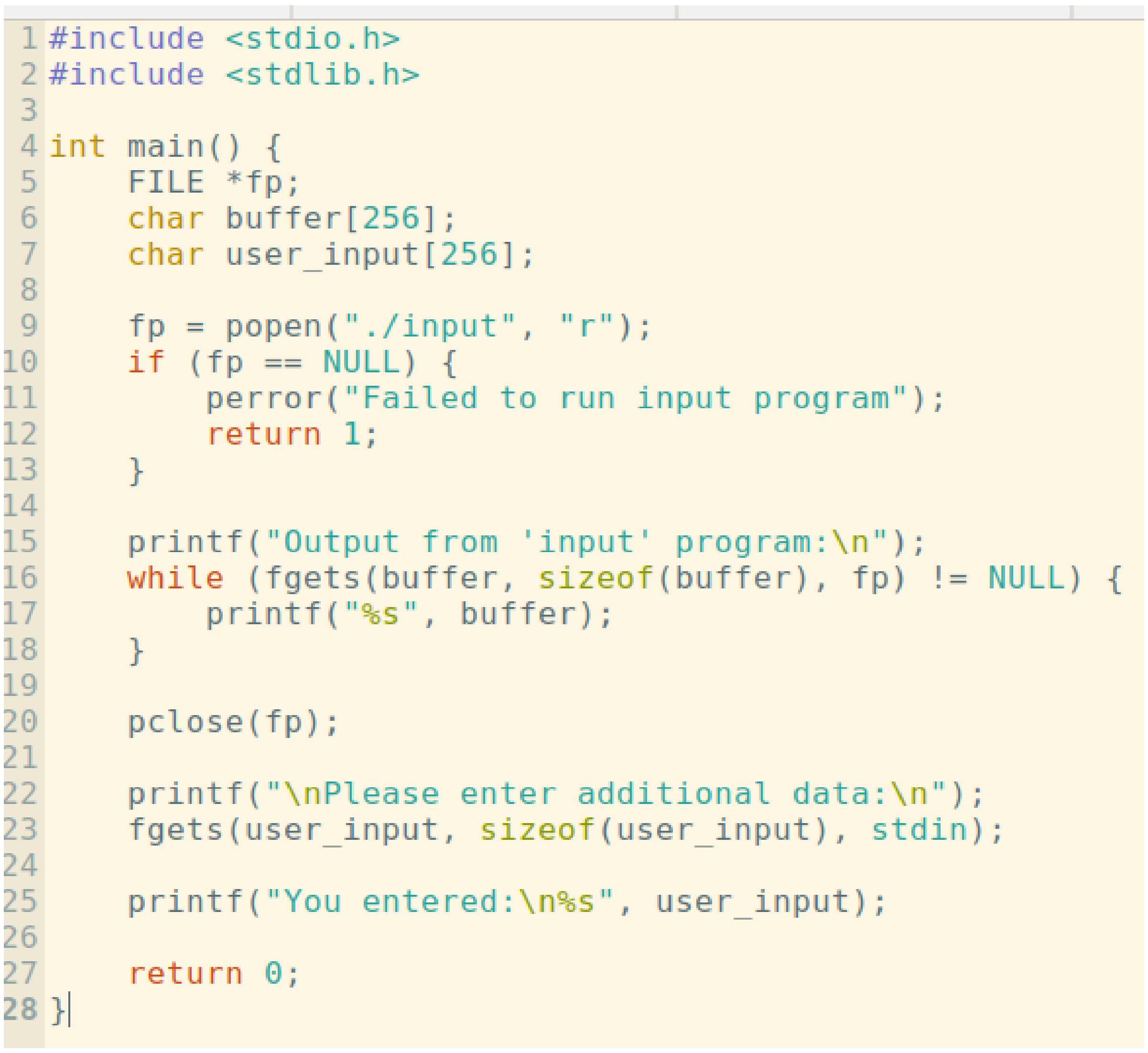
Q1. You have an existing program, “input.c,” which prints a welcome message for the currently logged-in user and prints the date and time. The compiled file of this program is “input”. Write another program that calls the compiled file “input” and prints the data entered by the user.

Program: (input.c)



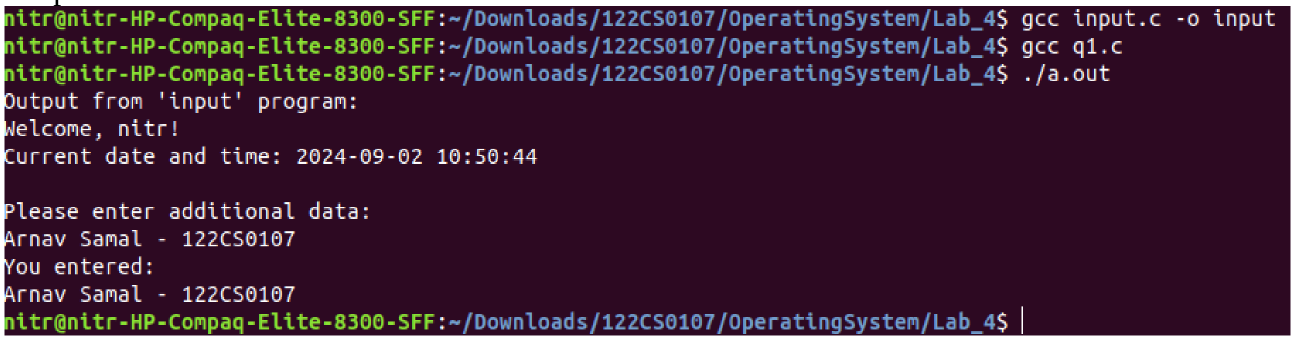
```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>

void print_date_time()
{
    time_t now;
    struct tm *tm_info;
    char buffer[26];
    time(&now);
    tm_info = localtime(&now);
    strftime(buffer, 26, "%Y-%m-%d %H:%M:%S", tm_info);
    printf("Current date and time: %s\n", buffer);
}

int main()
{
    char username[256];

    if (getlogin_r(username, sizeof(username)) != 0)
    {
        perror("getlogin_r");
        return EXIT_FAILURE;
    }

    printf("Welcome, %s!\n", username);
    print_date_time();
    return EXIT_SUCCESS;
}
```

Program: (q1.c)



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

int main() {
    FILE *fp;
    char buffer[256];
    char user_input[256];

    fp = popen("./input", "r");
    if (fp == NULL) {
        perror("Failed to run input program");
        return 1;
    }

    printf("Output from 'input' program:\n");
    while (fgets(buffer, sizeof(buffer), fp) != NULL) {
        printf("%s", buffer);
    }

    pclose(fp);

    printf("\nPlease enter additional data:\n");
    fgets(user_input, sizeof(user_input), stdin);

    printf("You entered:\n%s", user_input);

    return 0;
}
```

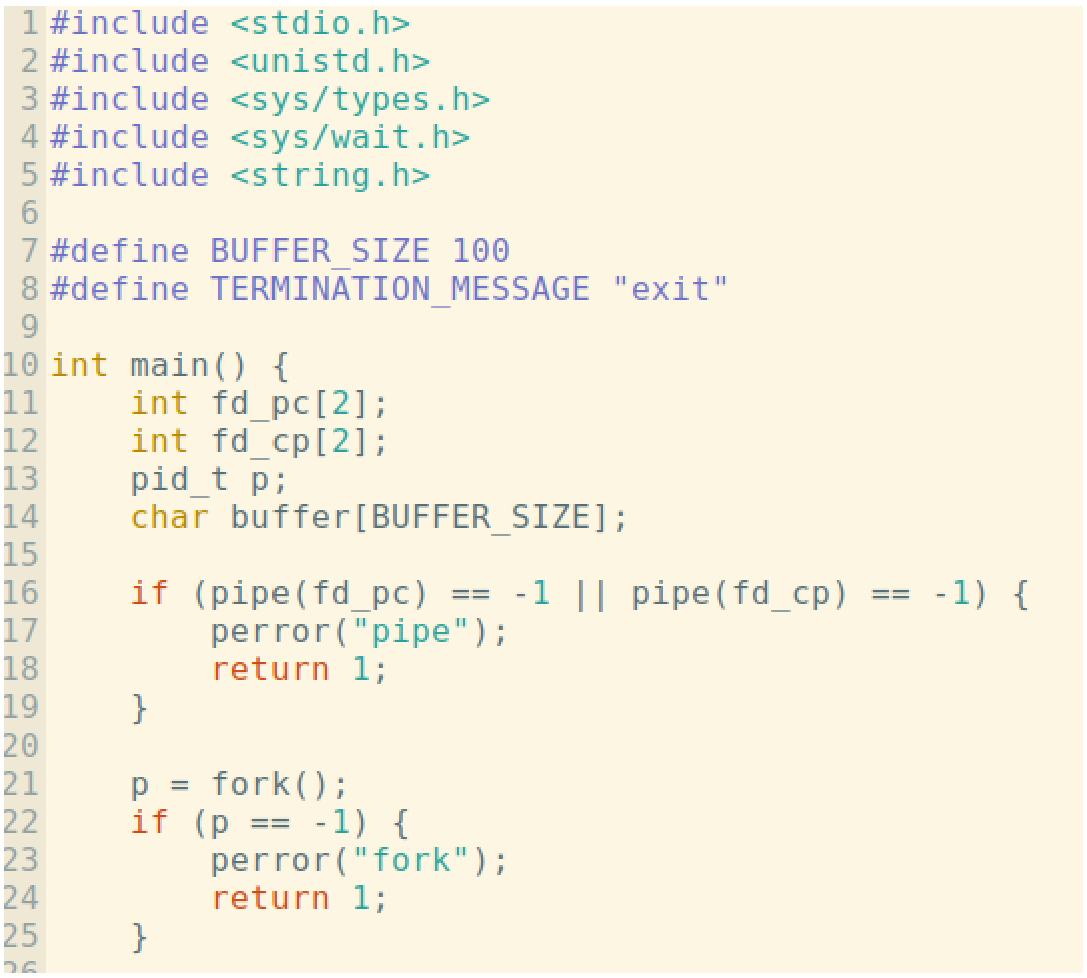
Output:



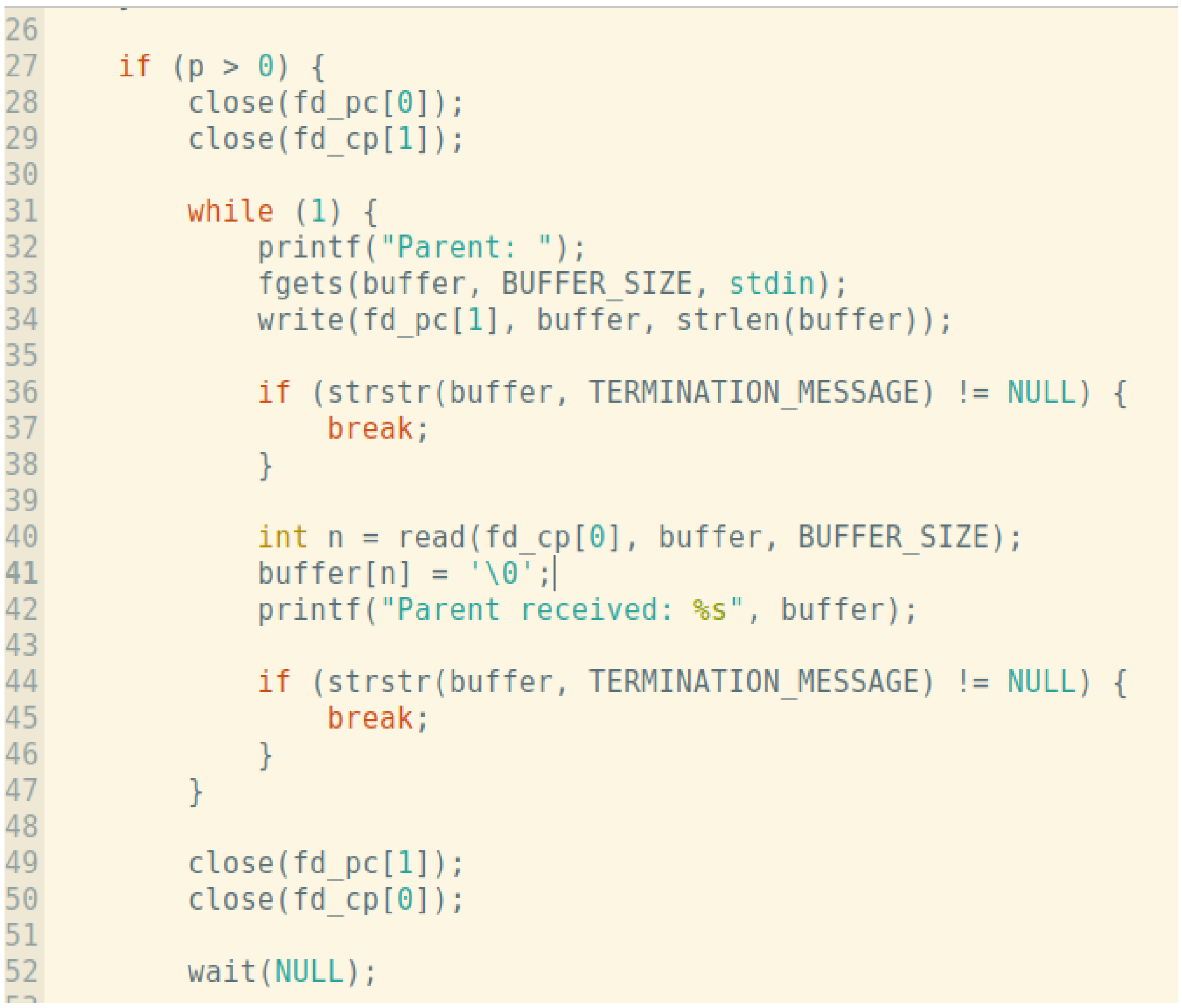
```
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ gcc input.c -o input
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ gcc q1.c
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ ./a.out
Output from 'input' program:
Welcome, nitr!
Current date and time: 2024-09-02 10:50:44

Please enter additional data:
Arnav Samal - 122CS0107
You entered:
Arnav Samal - 122CS0107
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ 
```

Q2. Create a basic chat application using pipes where the parent and child processes can send and receive messages in a loop until a specific termination message (e.g., "exit") is sent.

Program: (q2.c)

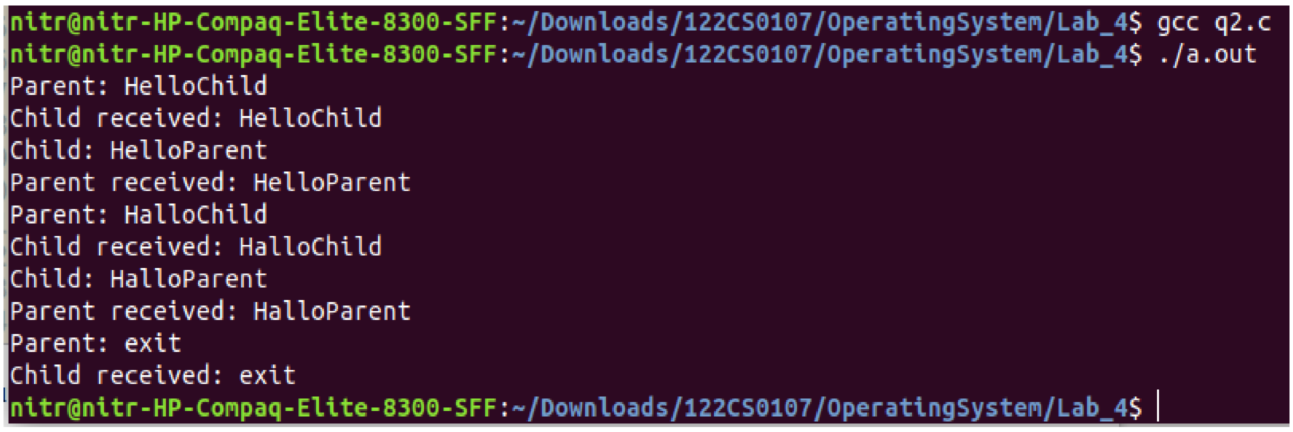


```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <string.h>

#define BUFFER_SIZE 100
#define TERMINATION_MESSAGE "exit"

int main() {
    int fd_pc[2];
    int fd_cp[2];
    pid_t p;
    char buffer[BUFFER_SIZE];

    if (pipe(fd_pc) == -1 || pipe(fd_cp) == -1) {
        perror("pipe");
        return 1;
    }

    p = fork();
    if (p == -1) {
        perror("fork");
        return 1;
    }
```



```c

    if (p > 0) {
        close(fd_pc[0]);
        close(fd_cp[1]);

        while (1) {
            printf("Parent: ");
            fgets(buffer, BUFFER_SIZE, stdin);
            write(fd_pc[1], buffer, strlen(buffer));

            if (strstr(buffer, TERMINATION_MESSAGE) != NULL) {
                break;
            }

            int n = read(fd_cp[0], buffer, BUFFER_SIZE);
            buffer[n] = '\0';
            printf("Parent received: %s", buffer);

            if (strstr(buffer, TERMINATION_MESSAGE) != NULL) {
                break;
            }
        }

        close(fd_pc[1]);
        close(fd_cp[0]);

        wait(NULL);

    } else {
        close(fd_pc[1]);
        close(fd_cp[0]);

        while (1) {

            int n = read(fd_pc[0], buffer, BUFFER_SIZE);
            buffer[n] = '\0';
            printf("Child received: %s", buffer);

            if (strstr(buffer, TERMINATION_MESSAGE) != NULL) {
                break;
            }

            printf("Child: ");
            fgets(buffer, BUFFER_SIZE, stdin);
            write(fd_cp[1], buffer, strlen(buffer));

            if (strstr(buffer, TERMINATION_MESSAGE) != NULL) {
                break;
            }
        }

        close(fd_pc[0]);
        close(fd_cp[1]);

        _exit(0);
    }

    return 0;
}
```

Output:



```
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ gcc q2.c
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ ./a.out
Parent: HelloChild
Child received: HelloChild
Child: HelloParent
Parent received: HelloParent
Parent: HalloChild
Child received: HalloChild
Child: HalloParent
Parent received: HalloParent
Parent: exit
Child received: exit
nitr@nitr-HP-Compaq-Elite-8300-SFF:~/Downloads/122CS0107/OperatingSystem/Lab_4$ 
```