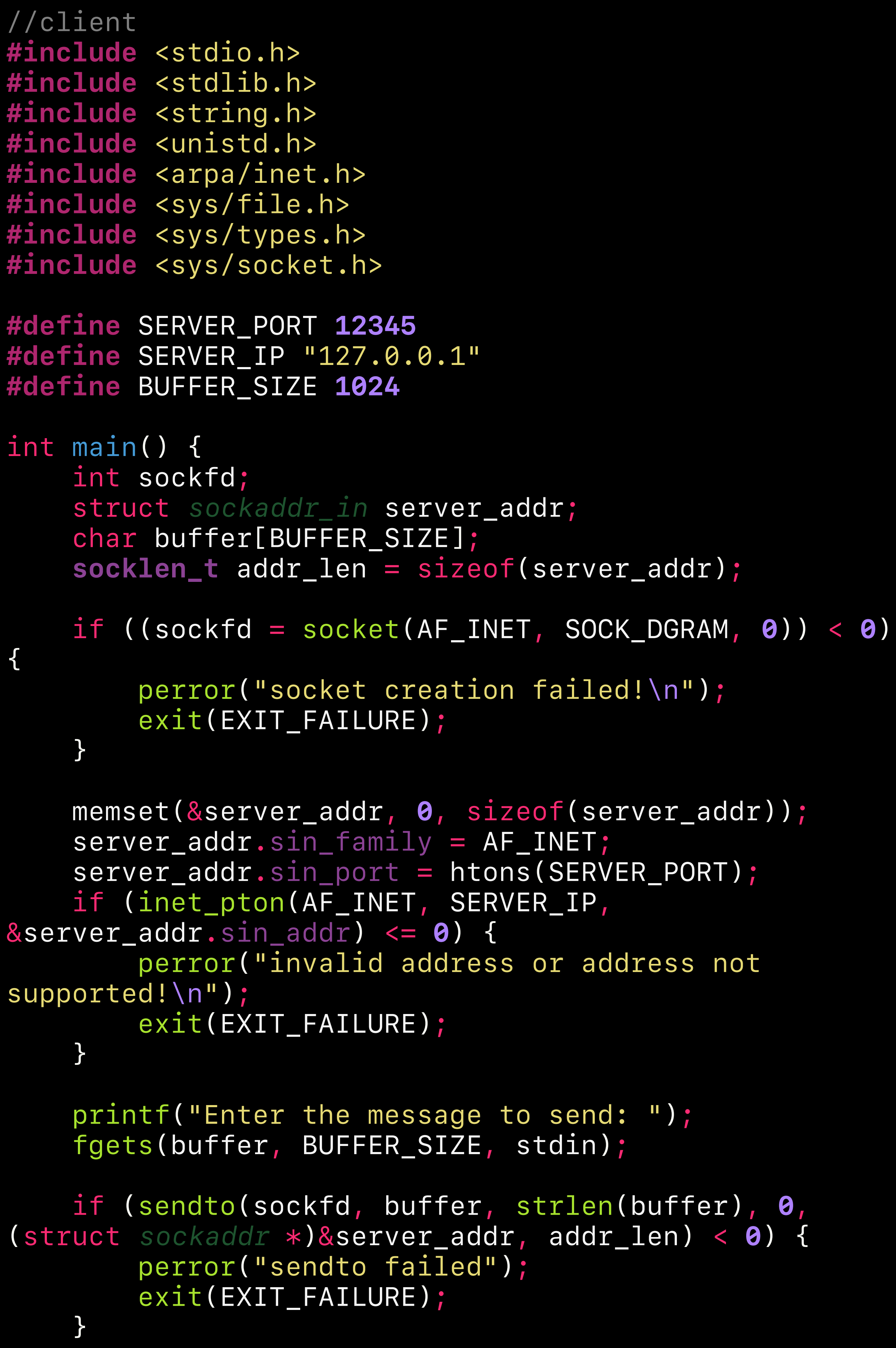
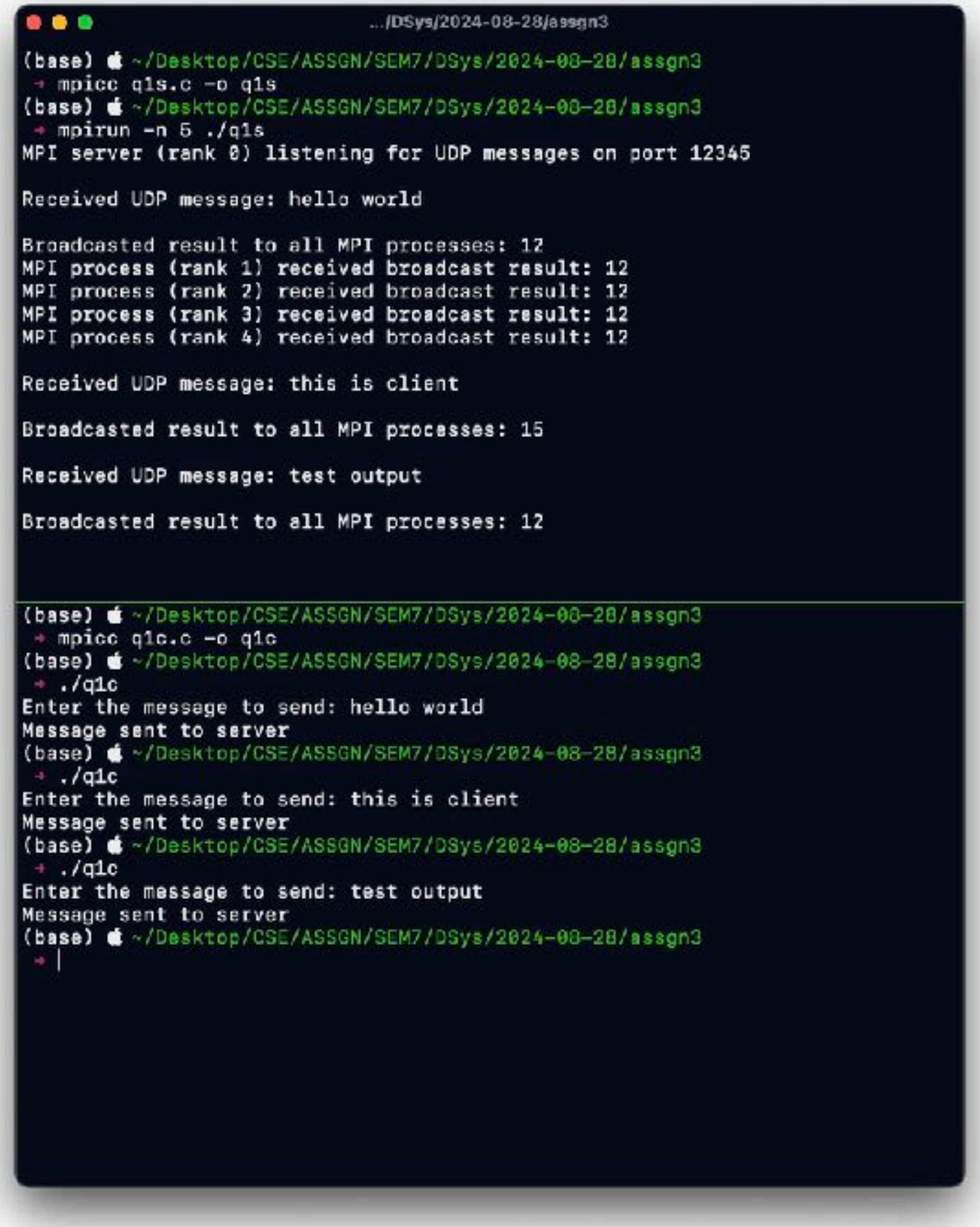
ASSIGNMENT 3
02/09/24

NAME : SHRESTH SONKAR
REGNO : 20214272
GROUP : CS7D
TOPIC : DISTRIBUTED SYSTEM
CODE : CS-17201



```
//client
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/file.h>
#include <sys/types.h>
#include <sys/socket.h>

#define SERVER_PORT 12345
#define SERVER_IP "127.0.0.1"
#define BUFFER_SIZE 1024

int main() {
    int sockfd;
    struct sockaddr_in server_addr;
    char buffer[BUFFER_SIZE];
    socklen_t addr_len = sizeof(server_addr);

    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0)
{
        perror("socket creation failed!\n");
        exit(EXIT_FAILURE);
    }

    memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(SERVER_PORT);
    if (inet_pton(AF_INET, SERVER_IP,
&server_addr.sin_addr) <= 0) {
        perror("invalid address or address not
supported!\n");
        exit(EXIT_FAILURE);
    }

    printf("Enter the message to send: ");
    fgets(buffer, BUFFER_SIZE, stdin);

    if (sendto(sockfd, buffer, strlen(buffer), 0,
(struct sockaddr *)&server_addr, addr_len) < 0) {
        perror("sendto failed");
        exit(EXIT_FAILURE);
    }
    printf("Message sent to server\n");
    close(sockfd);
    return 0;
}

//server
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <mpi.h>

#define UDP_PORT 12345
#define BUFFER_SIZE 1024

int main(int argc, char *argv[]) {
    int rank, size;
    MPI_Status status;
    char buffer[BUFFER_SIZE];
    int udp_socket;
    struct sockaddr_in server_addr, client_addr;
    socklen_t addr_len = sizeof(client_addr);

    // Initialize MPI
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank == 0) {
        printf("MPI server (rank %d) listening for UDP
messages on port %d\n", rank, UDP_PORT);
        if ((udp_socket = socket(AF_INET, SOCK_DGRAM,
0)) < 0) {
            perror("UDP socket creation failed!\n");
            MPI_Finalize();
            exit(EXIT_FAILURE);
        }

        memset(&server_addr, 0, sizeof(server_addr));
        server_addr.sin_family = AF_INET;
        server_addr.sin_addr.s_addr = INADDR_ANY;
        server_addr.sin_port = htons(UDP_PORT);
        if (bind(udp_socket, (const struct sockaddr
*)&server_addr, sizeof(server_addr)) < 0) {
            perror("UDP socket bind failed!\n");
            close(udp_socket);
            MPI_Finalize();
            exit(EXIT_FAILURE);
        }

        while (1) {
            int n = recvfrom(udp_socket, buffer,
BUFFER_SIZE, 0, (struct sockaddr *)&client_addr,
&addr_len);
            if (n < 0) {
                perror("UDP recvfrom failed!\n");
                break;
            }

            buffer[n] = '\0';
            printf("\nReceived UDP message: %s\n",
buffer);

            int result = strlen(buffer);
            MPI_Bcast(&result, 1, MPI_INT, 0,
MPI_COMM_WORLD);
            printf("Broadcasted result to all MPI
processes: %d\n", result);
        }
        close(udp_socket);
    } else {
        int received_result;
        MPI_Bcast(&received_result, 1, MPI_INT, 0,
MPI_COMM_WORLD);
        printf("MPI process (rank %d) received
broadcast result: %d\n", rank, received_result);
    }

    MPI_Finalize();
    return 0;
}
```



```
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ mpicc q1s.c -o q1s
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ mpirun -n 5 ./q1s
MPI server (rank 0) listening for UDP messages on port 12345

Received UDP message: hello world

Broadcasted result to all MPI processes: 12
MPI process (rank 1) received broadcast result: 12
MPI process (rank 2) received broadcast result: 12
MPI process (rank 3) received broadcast result: 12
MPI process (rank 4) received broadcast result: 12

Received UDP message: this is client

Broadcasted result to all MPI processes: 15

Received UDP message: test output

Broadcasted result to all MPI processes: 12
```

```
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ mpicc q1c.c -o q1c
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ ./q1c
Enter the message to send: hello world
Message sent to server
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ ./q1c
Enter the message to send: this is client
Message sent to server
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ ./q1c
Enter the message to send: test output
Message sent to server
(base)  ~/Desktop/CSE/ASSGN/SEM7/DSys/2024-08-28/assgn3
 ➜ |
```