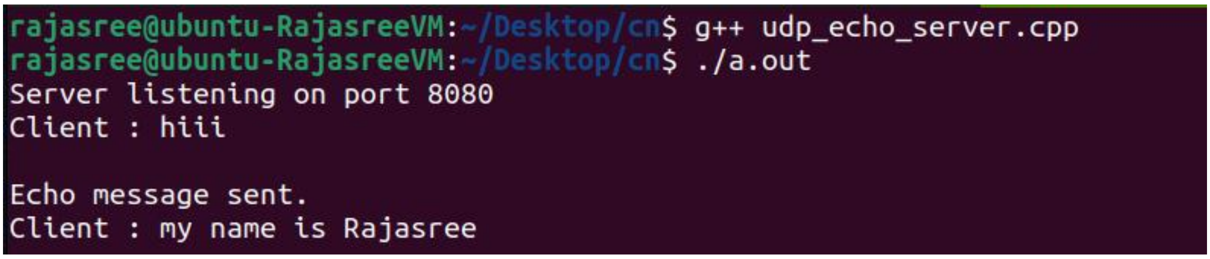
/*udp echo client server*/

//udp echo server

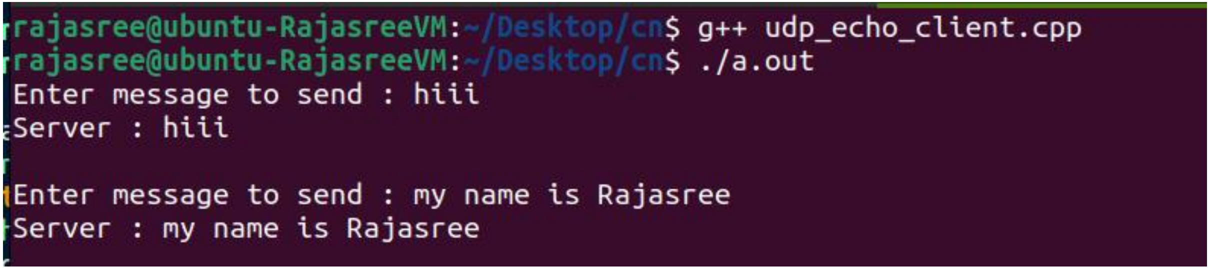
```
#include <iostream>
#include <string>
#include <cstring>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUFFER_SIZE 1024

int main() {
    int sockfd;
    struct sockaddr_in servaddr, cliaddr;
    char buffer[BUFFER_SIZE];
    socklen_t len;

    // Creating socket file descriptor
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        std::cerr << "socket creation failed" << std::endl;
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));
    memset(&cliaddr, 0, sizeof(cliaddr));

    // Filling server information
    servaddr.sin_family = AF_INET; // IPv4
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // Bind the socket with the server address
    if (bind(sockfd, (const struct sockaddr *)&servaddr, sizeof(servaddr)) < 0) {
        std::cerr << "bind failed" << std::endl;
        exit(EXIT_FAILURE);
    }

    std::cout << "Server listening on port " << PORT << std::endl;

    while (1) {
        len = sizeof(cliaddr);
        int n = recvfrom(sockfd, (char *)buffer, BUFFER_SIZE, MSG_WAITALL, (struct sockaddr *)&cliaddr, &len);
        buffer[n] = '\0';
        std::cout << "Client : " << buffer << std::endl;
        sendto(sockfd, (const char *)buffer, strlen(buffer), MSG_CONFIRM, (const struct sockaddr *)&cliaddr, len);
        std::cout << "Echo message sent." << std::endl;
    }

    return 0;
}
```



```
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ g++ udp_echo_server.cpp
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ ./a.out
Server listening on port 8080
Client : hiii

Echo message sent.
Client : my name is Rajasree
```

//udp echo client

```
#include <iostream>
#include <string>
#include <cstring>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUFFER_SIZE 1024

int main() {
    int sockfd;
    char buffer[BUFFER_SIZE];
    struct sockaddr_in servaddr;

    // Creating socket file descriptor
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        std::cerr << "socket creation failed" << std::endl;
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));

    // Filling server information
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = INADDR_ANY;

    int n, len;
    while (1) {

        std::cout << "Enter message to send : ";

        fgets(buffer, BUFFER_SIZE, stdin);


        sendto(sockfd, (const char *)buffer, strlen(buffer), MSG_CONFIRM, (const struct sockaddr
*)&servaddr, sizeof(servaddr));


        n = recvfrom(sockfd, (char *)buffer, BUFFER_SIZE, MSG_WAITALL, (struct sockaddr *)&servaddr,
(socklen_t *)&len);

        buffer[n] = '\0';

        std::cout << "Server : " << buffer << std::endl;

    }


    close(sockfd);

    return 0;

}
```



```
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ g++ udp_echo_client.cpp
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ ./a.out
Enter message to send : hiii
Server : hiii

Enter message to send : my name is Rajasree
Server : my name is Rajasree
```