Source Code: Client

```
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#define MAX 80
#define PORT 5007
#define SA struct sockaddr

// Chat Function
void chat(int sockfd)
{
    // Create Variables
    char buff[MAX];
    int n;
    // Infinite Loop for chat
    while(1) {
        // Stores null values in the array
        bzero(buff, sizeof(buff));
        printf("\n     >Enter message: ");
        n = 0;
        // Read input from command prompt
        while ((buff[n++] = getchar()) != '\n')
            ;
        // Send message to server socket
        write(sockfd, buff, sizeof(buff));
        bzero(buff, sizeof(buff));
        // read message from server
        read(sockfd, buff, sizeof(buff));
        printf("       >From Server: %s", buff);
        if ((strncmp(buff, "exit", 4)) == 0) {
            printf("   >>Client Exit...\n");
            break;
        }
    }
}

// Driver Function
int main()
{
    // Variables
    int sockfd = 0;
    int connfd = 0;
    struct sockaddr_in servaddr, cli;

    // Create socket and varify
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        printf("       >>[FAILED] Socket creation failed...\n");
        exit(0);
    }
    else
        printf("       >>[SUCCESS] Socket created..\n");
```

```
    // bzero stores null values in the array
    bzero(&servaddr, sizeof(servaddr));

    // ASSIGN IP address, PORT #
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr("130.191.166.3"); // Server IP
    servaddr.sin_port = htons(PORT);

    // Connect client socket to server socket
    if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr)) != 0) {
        printf("       >>[FAILED] Cannot connect to server...\n");
        exit(0);
    }
    else
        printf("       >>[SUCCESS] Established connection to the
server...\n");

    // call chat function
    chat(sockfd);

    // close socket connection
    close(sockfd);
}
```

Source Code: Server

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#define MAX 80
#define PORT 5007
#define SA struct sockaddr

// Function designed for chat between client and server.
void chat(int sockfd)
{
    // Variables
    char buff[MAX];
    int n = 0;
    // Chat loop
    while (1) {
        // bzero stores null values in array
        bzero(buff, MAX);

        // Read message and copy to buffer
        read(sockfd, buff, sizeof(buff));
        // print buffer (contains message from client)
        printf("       >From client: %s\t     >To client : ", buff);
        bzero(buff, MAX);
        n = 0;
        // Copy server message in buffer
        while ((buff[n++] = getchar()) != '\n')
            ;

        // Reverse Message
        int i = 0;
        int j = 0;
        int temp = 0;
        int l = strlen(buff);
        // Reverse string by each char in array
        for (i = 0, j = l - 1; i < j; i++, j--) {
        temp = buff[i];
        buff[i] = buff[j];
        buff[j] = temp;
    }

        // Send message to client
        write(sockfd, buff, sizeof(buff));

        // End chat
        if (strncmp("tixe", buff, 4) == 0) {
            printf("   >>Server Exit...\n");
```

```
            break;
        }
    }
}

// Driver Function
int main()
{
    // Variables
    int sockfd = 0;
    int connfd = 0;
    int len = 0;
    struct sockaddr_in servaddr, cli;

    // Create Socket and verify
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        printf("       >>[FAILED] Socket creation failed...\n");
        exit(0);
    }
    else
        printf("       >>[SUCCESS] Socket has been created..\n");
    // bzero stores null values in array
    bzero(&servaddr, sizeof(servaddr));

    // Assign IP Address and PORT #
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(PORT);

    // Binding newly created socket to given IP and verification
    if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
        printf("       >>[FAILED] Socket cannot bind...\n");
        exit(0);
    }
    else
        printf("       >>[SUCCESS] Socket has binded..\n");

    // Server ready to listen for client
    if ((listen(sockfd, 5)) != 0) {
        printf("       >>[FAILED] Listen failed...\n");
        exit(0);
    }
    else
        printf("       >>[LOADING] Server is listening..\n");
    len = sizeof(cli);

    // Accept data package from client and establish socket connecton
    connfd = accept(sockfd, (SA*)&cli, &len);
    if (connfd < 0) {
        printf("       >>[FAILED] Server connection failed...\n");
        exit(0);
    }
    else
        printf("       >>[SUCCESS] Server connection accepted from
client...\n");
```

```
    // Chat function
    chat(connfd);

    // After exit chat -> close socket
    close(sockfd);
}
```