**Q1**

**Server.c**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include <unistd.h>

#include<string.h>

int main()

{

int sockfd\_150,fd1\_150, length\_150,i\_150;

char buf\_150[100]; /\* We will use this buffer for communication \*/

struct sockaddr\_in sa\_addr\_150,cl\_addr\_150;

sockfd\_150=socket(AF\_INET,SOCK\_STREAM,0);

 sa\_addr.sin\_family=AF\_INET;

 sa\_addr.sin\_addr.s\_addr=INADDR\_ANY;

 sa\_addr.sin\_port=htons(8000);

 memset(sa\_addr.sin\_zero, '\0', sizeof sa\_addr.sin\_zero);

 i=bind(sockfd\_150,(struct sockaddr \*)&sa\_addr\_150,sizeof(sa\_addr\_150));

 printf("test %d%d\n",sockfd\_150,i\_150);

 listen(sockfd\_150,5);

 length=sizeof(cl\_addr\_150);

 fd1\_150=accept(sockfd\_150, (struct sockaddr \*) &cl\_addr\_150,&length\_150);

 char received\_messages[100];

 char sent\_messages[100];

 while(1) {

     int i = recv(fd1, received\_messages, 100, 0);

     received\_messages[i] = '\0';

     printf("%s\n", received\_messages);

     if(strcmp(received\_messages,"exit")==0) {

         printf("%d", strcmp(received\_messages,"exit"));

         strcpy(sent\_messages,"Bye!");

         send(fd1,sent\_messages,100,0);

         break;

     } else {

         strcpy(sent\_messages, received\_messages);

         send(fd1, sent\_messages,100, 0);

     }

 }

close(fd1);

return 0;

}

CLIENT.C

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include <unistd.h>

#include<string.h>

int main()  {

    int i\_150,sockfd\_150;

    char buf[100];

    struct sockaddr\_in sa\_addr;

    sockfd\_150=socket(AF\_INET,SOCK\_STREAM,0);

    sa\_addr.sin\_family=AF\_INET;

    sa\_addr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

    sa\_addr.sin\_port=htons(8000);

    memset(sa\_addr.sin\_zero, '\0', sizeof sa\_addr.sin\_zero);

    i\_150=connect(sockfd\_150,(struct sockaddr \*)&sa\_addr,sizeof(sa\_addr));

    char messages[100];

    char received[100];

    while(1) {

        printf("\n");

        printf("Enter your message :: \n");

        scanf("%s",messages);

        send(sockfd\_150, messages, 100, 0);

        recv(sockfd\_150, received, 100, 0);

        printf("Message Received: %s", received);

        if(strcmp(messages,"exit")==0) {

         printf("%d", strcmp(messages,"exit"));

         strcpy(received,"Bye!");

         send(sockfd\_150,received,100,0);

         break;

     }

    }

    close(sockfd\_150);

    return 0;

}

**Q2**

**Server.c**

#include <netinet/in.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

#define PORT 8090

int main()

{

    int server\_fd\_150, new\_socket\_150, valread\_150;

    struct sockaddr\_in address;

    char str[100];

    int addrlen = sizeof(address);

    char buffer[1024] = { 0 };

    char\* hello = "Hello from server";

    if ((server\_fd\_150 = socket(AF\_INET,

                        SOCK\_STREAM, 0)) == 0) {

        perror("socket failed");

        exit(EXIT\_FAILURE);

    }

    address.sin\_family = AF\_INET;

    address.sin\_addr.s\_addr = INADDR\_ANY;

    address.sin\_port = htons(PORT);

    if (bind(server\_fd\_150, (struct sockaddr\*)&address,sizeof(address)) < 0){

        perror("bind failed");

        exit(EXIT\_FAILURE);

    }

    if (listen(server\_fd\_150, 3) < 0){

        perror("listen");

        exit(EXIT\_FAILURE);

    }

    if ((new\_socket\_150 = accept(server\_fd\_150,(struct sockaddr\*)&address,(socklen\_t\*)&addrlen)) < 0){

        perror("accept");

        exit(EXIT\_FAILURE);

    }

    valread\_150 = read(new\_socket\_150, str,sizeof(str));

    int i, j, temp;

    int l = strlen(str);

    printf("\nString sent by client:%s\n", str);

    for (i = 0, j = l - 1; i < j; i++, j--){

        temp = str[i];

        str[i] = str[j];

        str[j] = temp;

    }

    send(new\_socket\_150, str, sizeof(str), 0);

    printf("\nModified string sent to client\n");

    return 0;

}

**Client.c**

// C client code to send string to reverse

#include <arpa/inet.h>

#include <netinet/in.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

#define PORT 8090

// Driver code

int main()

{

    struct sockaddr\_in address\_150;

    int sock\_150 = 0, valread\_150;

    struct sockaddr\_in serv\_addr;

    char str[100];

    printf("\nInput the string:");

    scanf("%[^\n]s", str);

    char buffer[1024] = { 0 };

    if ((sock\_150 = socket(AF\_INET,

                    SOCK\_STREAM, 0))

        < 0) {

        printf("\n Socket creation error \n");

        return -1;

    }

    memset(&serv\_addr, '0', sizeof(serv\_addr));

    serv\_addr.sin\_family = AF\_INET;

    serv\_addr.sin\_port = htons(PORT);

    if (inet\_pton(AF\_INET, "127.0.0.1",&serv\_addr.sin\_addr)<= 0) {

        printf("\nAddress not supported \n");

        return -1;

    }

    if (connect(sock\_150, (struct sockaddr\*)&serv\_addr,sizeof(serv\_addr))< 0) {

        printf("\nConnection Failed \n");

        return -1;

    }

    int l = strlen(str);

    send(sock\_150, str, sizeof(str), 0);

    valread\_150= read(sock\_150, str, l);

    printf("%s\n", str);

    return 0;

}