SOURCE CODE –

server.c

```
#include<unistd.h>
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>

int main(int argc,char **argv)
{
int len;
int sockfd,newfd,n;
struct sockaddr_in servaddr,cliaddr;
char buff[1024];
char str[1000];
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0)
perror("cannot create socket");
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=INADDR_ANY;
servaddr.sin_port=htons(atoi(argv[1]));
if(bind(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
perror("Bind error");

listen(sockfd,1);
len=sizeof(cliaddr);
printf("Waiting for connection\n");
newfd=accept(sockfd,(struct sockaddr*)&cliaddr,&len);
printf("Connected");
while(1)
{
        //Receiving the message
        n=read(newfd,buff,sizeof(buff));
        write(newfd,buff,sizeof(buff));
        printf("\nReceived Message is \t%s\n",buff);
}
close(newfd);
close(sockfd);
return 0;
}
```

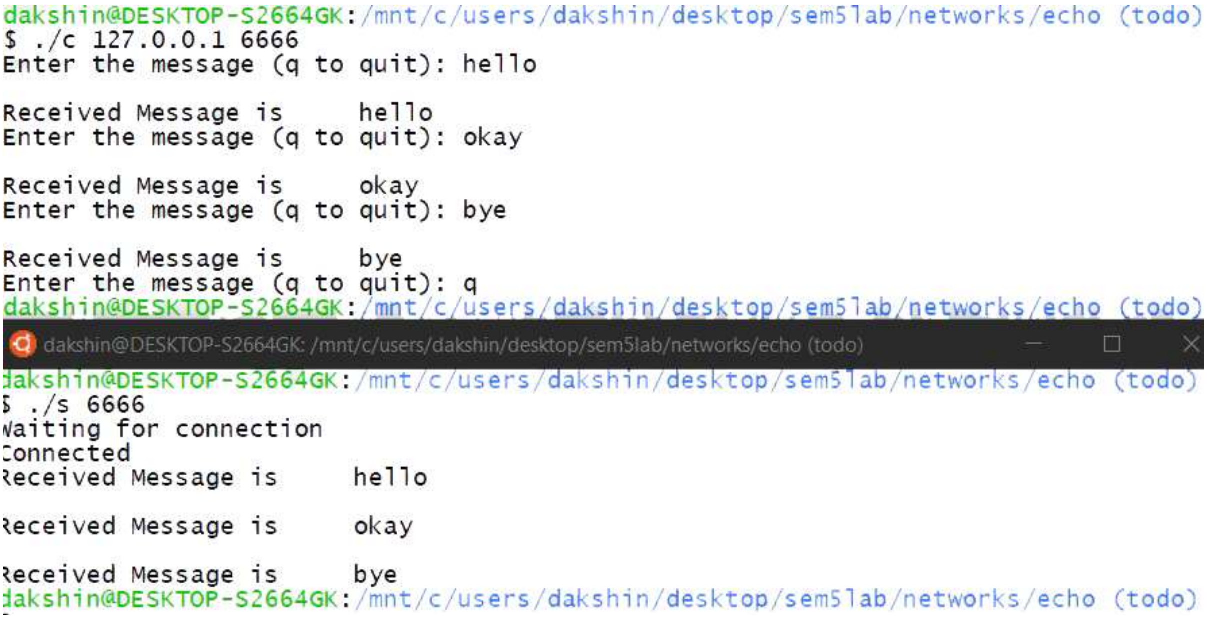
client.c

```
#include<stdlib.h>
#include<unistd.h>
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include <arpa/inet.h>
#include<errno.h>

int main(int argc,char **argv)
{
int len;
int sockfd,n;
int r;
struct sockaddr_in servaddr,cliaddr;
char str[1000];
char buff[1024];
char buff1[1024];
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0)
perror("cannot create socket");
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr(argv[1]);
servaddr.sin_port=htons(atoi(argv[2]));
r=connect(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));
if(r==-1)
  printf("Errorno = %d\n",errno);

while(1)
{
        //Sending Message
        printf("Enter the message (q to quit): ");
        scanf(" %[^\n]s",buff);
        if(!strcmp(buff,"q"))
                break;
        n=send(sockfd,buff,sizeof(buff),0);
        n=recv(sockfd,buff1,sizeof(buff),0);
        printf("\nReceived Message is \t%s\n",buff1);
}
close(sockfd);
return 0;
}
```

OUTPUT –



```
dakshin@DESKTOP-S2664GK:/mnt/c/users/dakshin/desktop/sem5lab/networks/echo (todo)
$ ./c 127.0.0.1 6666
Enter the message (q to quit): hello

Received Message is     hello
Enter the message (q to quit): okay

Received Message is     okay
Enter the message (q to quit): bye

Received Message is     bye
Enter the message (q to quit): q
dakshin@DESKTOP-S2664GK:/mnt/c/users/dakshin/desktop/sem5lab/networks/echo (todo)
```

dakshin@DESKTOP-S2664GK: /mnt/c/users/dakshin/desktop/sem5lab/networks/echo (todo)

```
dakshin@DESKTOP-S2664GK:/mnt/c/users/dakshin/desktop/sem5lab/networks/echo (todo)
$ ./s 6666
Waiting for connection
Connected
Received Message is     hello

Received Message is     okay

Received Message is     bye
dakshin@DESKTOP-S2664GK:/mnt/c/users/dakshin/desktop/sem5lab/networks/echo (todo)
```