***SERVER:***

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

//headers for socket and related functions

#include <sys/types.h>

#include <sys/socket.h>

//for including structures which will store information needed

#include <netinet/in.h>

#include <unistd.h>

//for gethostbyname

#include "netdb.h"

#include "arpa/inet.h"

#define MAX 1000

#define BACKLOG 5 // how many pending connections queue will hold

int main()

{

char serverMessage[MAX];

char clientMessage[MAX];

//create the server socket

int socketDescriptor = socket(AF\_INET, SOCK\_STREAM, 0);

struct sockaddr\_in serverAddress;

serverAddress.sin\_family = AF\_INET;

serverAddress.sin\_port = htons(9086);

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

//calling bind function to oir specified IP and port

bind(socketDescriptor, (struct sockaddr\*)&serverAddress, sizeof(serverAddress));

listen(socketDescriptor, BACKLOG);

//starting the accepting

int clientSocketDescriptor = accept(socketDescriptor, NULL, NULL);

while (1)

{

printf("\ntext message here .. :");

scanf("%s", serverMessage);

send(clientSocketDescriptor, serverMessage, sizeof(serverMessage) , 0);

//recieve the data from the server

recv(clientSocketDescriptor, &clientMessage, sizeof(clientMessage), 0) ;

//recieved data from the server successfully then printing the data obtained from the server

printf("\nCLIENT: %s", clientMessage);

}

//close the socket

close(socketDescriptor);

return 0;

}

***CLIENT:***

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

//headers for socket and related functions

#include <sys/types.h>

#include <sys/socket.h>

//for including structures which will store information needed

#include <netinet/in.h>

#include <unistd.h>

//for gethostbyname

#include "netdb.h"

#include "arpa/inet.h"

//defines

#define h\_addr h\_addr\_list[0] /\* for backward compatibility \*/

#define PORT 9086 // port number

#define MAX 1000 //maximum buffer size

//main function

int main(){

char serverResponse[MAX];

char clientResponse[MAX];

//creating a socket

int socketDescriptor = socket(AF\_INET, SOCK\_STREAM, 0);

//placeholder for the hostname and my ip address

char hostname[MAX], ipaddress[MAX];

struct hostent \*hostIP; //placeholder for the ip address

//if the gethostname returns a name then the program will get the ip address

if(gethostname(hostname,sizeof(hostname))==0){

hostIP = gethostbyname(hostname);//the netdb.h fucntion gethostbyname

}else{

printf("ERROR:FCC4539 IP Address Not ");

}

struct sockaddr\_in serverAddress;

serverAddress.sin\_family = AF\_INET;

serverAddress.sin\_port = htons(PORT);

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

connect(socketDescriptor, (struct sockaddr \*)&serverAddress, sizeof(serverAddress));

// getting the address port and remote host

printf("\nLocalhost: %s\n", inet\_ntoa(\*(struct in\_addr\*)hostIP->h\_addr));

printf("Local Port: %d\n", PORT);

printf("Remote Host: %s\n", inet\_ntoa(serverAddress.sin\_addr));

while (1)

{ //recieve the data from the server

recv(socketDescriptor, serverResponse, sizeof(serverResponse), 0);

//recieved data from the server successfully then printing the data obtained from the server

printf("\nSERVER : %s", serverResponse);

printf("\ntext message here... :");

scanf("%s", clientResponse);

send(socketDescriptor, clientResponse, sizeof(clientResponse), 0);

} //closing the socket

close(socketDescriptor);

return 0;

}

***OUTPUT:***