SERVER:

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<strings.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<stdlib.h>
#include<string.h>
int main(int argc, char *argv[])
{
    int sfd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in serveraddr, clientaddr;
    bzero(&serveraddr, sizeof(struct sockaddr_in));
    bzero(&clientaddr, sizeof(struct sockaddr_in));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = INADDR_ANY;
    serveraddr.sin_port=htons(7891);
    if(bind(sfd, (struct sockaddr*)&serveraddr, sizeof(serveraddr)) < 0)
    {
        perror("Bind error!\n");
        exit(0);
    }
    listen(sfd, 10);
    int ch = 1;
    int i = 0;
    int fd[10];
    char iplist[10][100];
    struct sockaddr_in clist[10];
    while(ch == 1)
    {
        socklen_t len = sizeof(struct sockaddr_in);
        int f1 = accept(sfd, (struct sockaddr *)&clientaddr, &len);
        fd[i] = f1;
        clist[i] = clientaddr;
        printf("New connection: ");
        struct sockaddr_in* pV4Addr = (struct sockaddr_in*)&clientaddr;
        struct in_addr ipAddr = pV4Addr->sin_addr;
        char str1[INET_ADDRSTRLEN];
        inet_ntop(AF_INET, &ipAddr, str1, INET_ADDRSTRLEN);
        strcpy(iplist[i], str1);
        i++;
        printf("%s\n", str1);
        sleep(1);
        printf("Enter 1 for more clients or 0 to quit: ");
        scanf("%d", &ch);
    }
    char *ipreq;
    char *macreply;
    macreply = (char*)malloc(100);
    ipreq = (char*)malloc(100);
    printf("Enter the IP to be requested: ");
    scanf("%s", ipreq);
    int total_connections = i;
    i = 0;
    while(i < total_connections) {
        write(fd[i], ipreq, 100);
        printf("Request sent to %s\n", iplist[i]);
```

```
        sleep(1);
        int nr = read(fd[i], macreply, 99);
        macreply[nr] = '\0';
        if(strcmp(macreply, "null") != 0) {
            printf("MAC Address of %s: %s\n", iplist[i], macreply);
            sleep(1);
            char buff[] = "10110110";
            write(fd[i], buff, sizeof(buff));
            break;
        }
        else {
            char buff[] = "IP mismatch!\n";
            write(fd[i], buff, sizeof(buff));
        }
        i++;
    }
    close(sfd);
    return 0;
}
```

CLIENT:

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<stdlib.h>
#include<fcntl.h>
#include<string.h>
#include<ctype.h>
int main(int argc, char *argv[])
{
    char *mac, *ip;
    mac = (char*)malloc(100);
    ip = (char*)malloc(100);
    for(int i = 0; i < 100; i++) ip[i] = '\0';
    FILE* fp = popen("ifconfig -a | grep -Po 'HWaddr \\K.*$'", "r");
    if(fgets(mac, 99, fp) != NULL) {
        printf("Client MAC Address is: %s", mac);
    }
    mac[17] = '\0';
    fp =  popen("ip route get 8.8.4.4 | head -1 | awk '{print $7}'", "r");
    if (fgets(ip, 99, fp) != NULL) {
        printf("Client IP Address is: %s",ip);
        ip[strlen(ip)-1] = '\0';
    }
    fclose(fp);
    int cfd = socket(AF_INET,SOCK_STREAM, 0);
    struct sockaddr_in serveraddr, clientaddr;
    bzero(&serveraddr, sizeof(struct sockaddr_in));
    bzero(&clientaddr, sizeof(struct sockaddr_in));
    bzero(&serveraddr,sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = inet_addr(argv[1]);
    serveraddr.sin_port = htons(7891);
    connect(cfd, (struct sockaddr*)&serveraddr, sizeof(serveraddr));
    char *buff;
    char *buff1 = "null";
    buff = (char*)malloc(100);
    int n1 = 0;
    while(n1 == 0)
        n1 = read(cfd, buff, 100);
    buff[n1] = '\0';
    printf("Received IP request is %s\n", buff);
    if(strcmp(buff, ip) == 0)
    {
        printf("IP matched, sending MAC Address to server!\n");
        write(cfd, mac, strlen(mac));
        read(cfd, buff, 100);
        printf("Received message: %s\n",buff);
    }
    else {
        printf("IP mismatch!\n");
        write(cfd, buff1, (sizeof(buff1)));
    }
    close(cfd);
    return 0;
}
```