```
#include <stdio.h>
#include <winsock2.h>

#define PORT_NUM 3800

struct cal_data
{
    int left_num;
    int right_num;
    char op;
    int result;
    short int error;
};

int main(int argc, char **argv)
{
    SOCKET sockfd;
    int addrlen;
    int cal_result;
    int left_num, right_num;
    struct sockaddr_in addr, cliaddr;
    struct cal_data rdata;

    WSADATA wsaData;

    if(WSAStartup(MAKEWORD(2,2), &wsaData) != NO_ERROR)
    {
        return 1;
    }

    if( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1 )
    {
        return 1;
    }

    memset((void *)&addr, 0x00, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = htonl(INADDR_ANY);
    addr.sin_port = htons(PORT_NUM);

    addrlen = sizeof(addr);
    if (bind(sockfd, (struct sockaddr *)&addr, addrlen) == -1)
    {
        return 1;
    }

    while(1)
```

```
    {
        addrlen = sizeof(cliaddr);
        recvfrom(sockfd, (char *)&rdata, sizeof(rdata), 0,(struct sockaddr
*)&cliaddr, &addrlen);
        #if DEBUG
        printf(lient "Info : %s (%d)₩n", inet_ntoa(cliaddr.sin_addr),
ntohs(cliaddr.sin_port));
        printf("%d %c %d₩n", ntohl(rdata.left_num), rdata.op,
ntohl(rdata.right_num));
        #endif

        left_num = ntohl(rdata.left_num);
        right_num = ntohl(rdata.right_num);
        switch(rdata.op)
        {
                case '+':
                        cal_result = left_num + right_num;
                        break;
                case '0':
                        cal_result = left_num - right_num;
                        break;
                case '*':
                        cal_result = left_num * right_num;
                        break;
                case '/':
                if(right_num == 0)
                {
                        rdata.error = htons(2);
                        break;
                }
                cal_result = left_num / right_num;
                break;
    }
        rdata.result = htonl(cal_result);
        sendto(sockfd, (char *)&rdata, sizeof(rdata), 0,
        (struct sockaddr *)&cliaddr, addrlen);
    }
    closesocket(sockfd);
    WSACleanup( );
    return 0;
}
```