사용한 IDE : Visual Studio 2010

(<http://blog.daum.net/wisepocket/74>) 를 통하여 WinPcap을 설치하고 Visual Studio 설정을 하였습니다.

(<http://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CC0QFjAB&url=http%3A%2F%2Fcopynull.tistory.com%2Fattachment%2Fcfile10.uf%402739C73B5274ACCD210E3B.cpp&ei=S7txVPGjDuTtmQWUuIGwAQ&usg=AFQjCNFmtOLE2yTE5aBDOxueZ8-v1sMsFw&bvm=bv.80185997,d.dGY&cad=rjt>) 구글을 통해서 기본적인 소스를 구했습니다.

**Assignment 3-1:  
- Capture.c 를 winpcap 라이브러리를 이용하는 버전으로 확장   
- Ethernet, ARP 정보 출력 부분 추가  
- 간단한 Filter 추가: protocol, ip address**

**(사용법)**

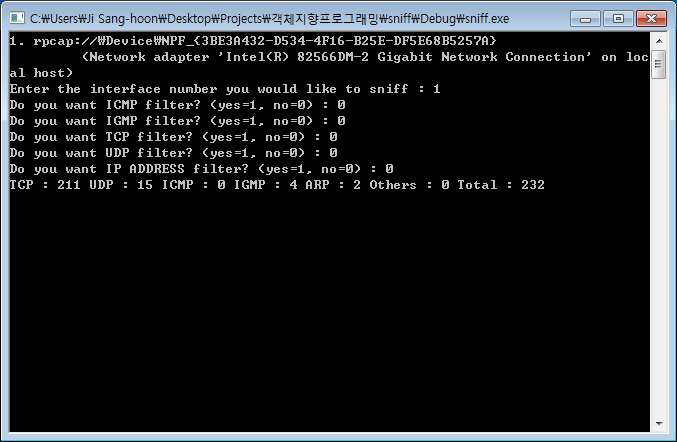
우선 인터페이스를 선택합니다.

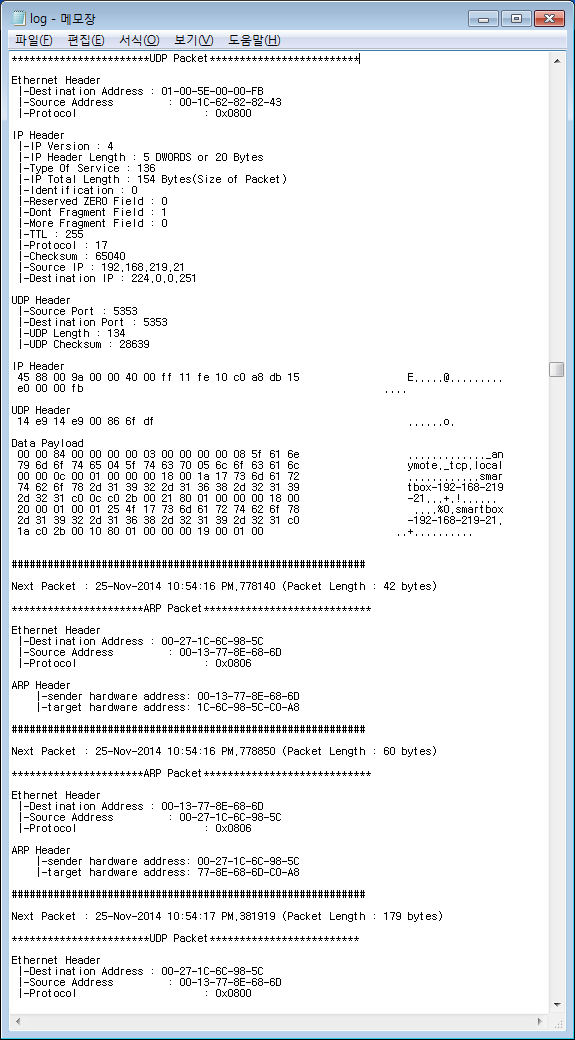
다음으로 사용자가 Filter를 선택할 수 있도록 하였습니다.

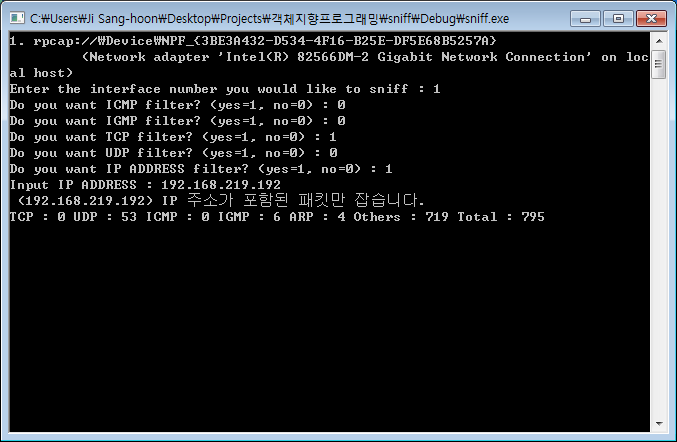
차례로 ICMP, IGMP, TCP, UDP 프로토콜의 필터 여부를 사용자에게서 1 또는 0으로 받아들입니다. (yes=1, no=0)

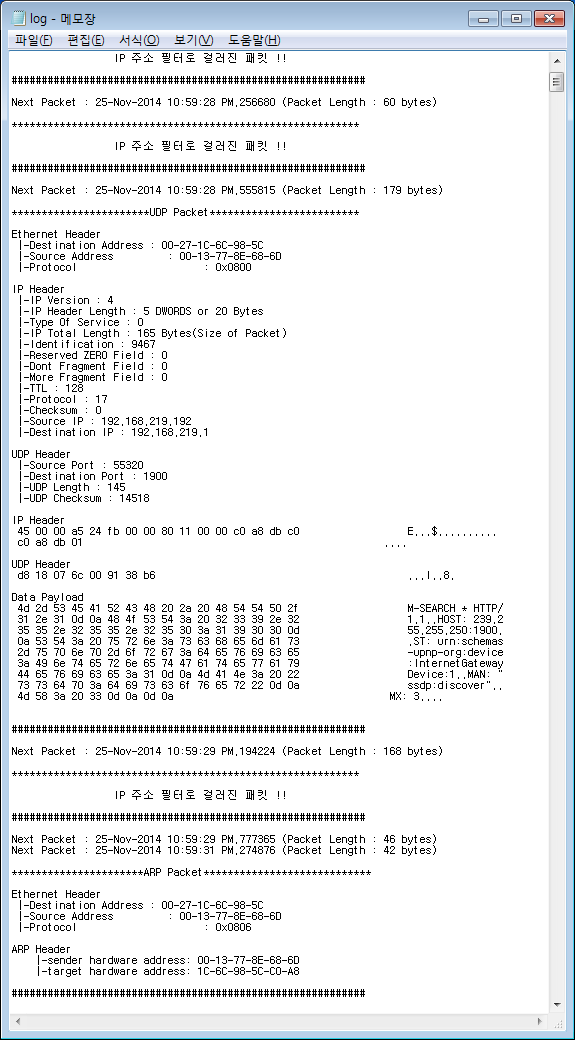
다음 사용자에게 IP 주소 필터를 사용할 지 여부를 물어보고, 사용한다면 IP 주소를 받아들입니다.

**(실행 화면)**









**Assignment 3-2:  
- PrjClient.cpp에 파일 전송기능 추가  
- 2개 이상 클라이언트의 동시 처리 기능 추가**

**(사용법)**

서버를 우선 실행합니다.

다시 프로그램을 실행시켜 클라이언트를 여럿이 접속시킬 수 있습니다.

서버가 살아있는 동안 모든 클라이언트와 채팅, 그림 그리기, 파일 전송이 가능합니다.

**(간단한 소스 분석)**

서버는 select() 함수를 이용한 멀티플렉싱을 합니다.

서버는 한 클라이언트가 보내온 데이터를 참여하는 모든 클라이언트의 소켓으로 그대로 보내주는 역할을 합니다.

서버 자신도 클라이언트와 채팅, 그림 그리기, 파일 전송을 해줄 수 있으며, 마찬가지로 보내고자 하는 데이터를 모든 클라이언트 소켓으로 보내줍니다.

파일 전송의 경우 recv()로 인한 블락킹 상태를 피하기 위해서 받을 데이터 크기 만큼 recv()를 하도록 하였습니다.

클라이언트 수에 제한을 두었습니다. 제한한 클라이언트 수를 넘어선다면, 서버가 클라이언트에게 접속을 끊기 위해 바로 closesocket ()를 통한 FIN을 보내도록 하였습니다.

**(실행 화면)**

