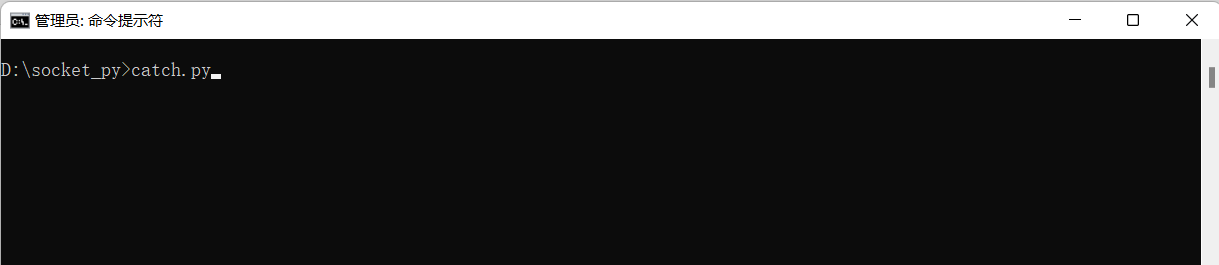
# 说明文档

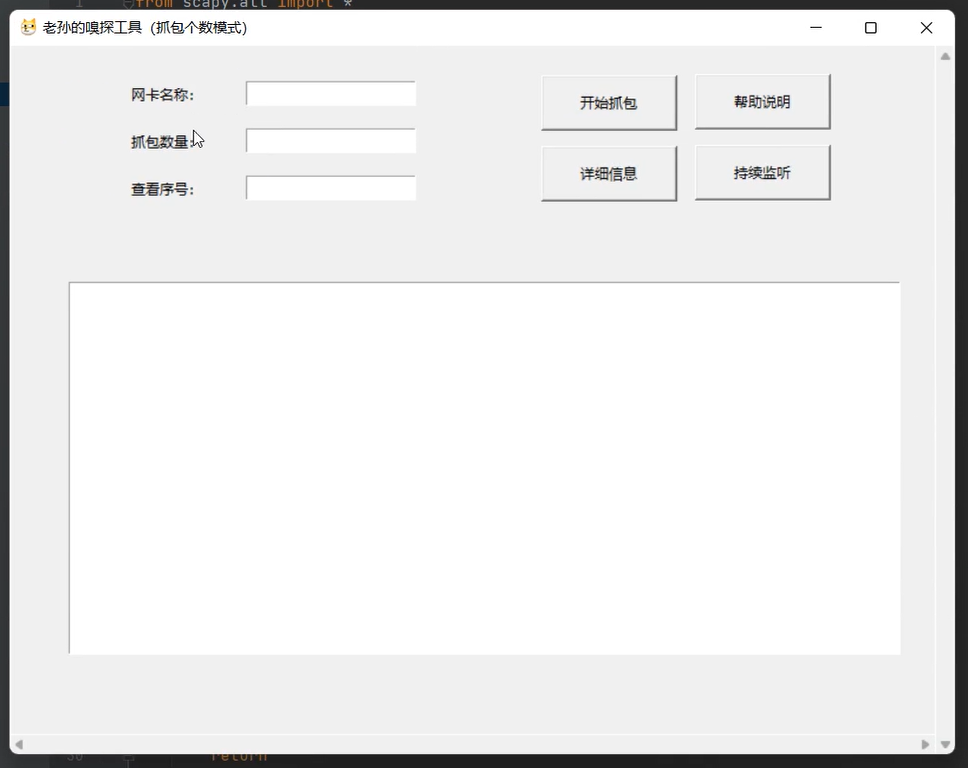
**选择题目**：1A 嗅探器的设计与实验

**编程语言**：python3

**思路简介**：本次实验我主要分为了两个模块：抓包个数模式、持续监听模式。累计抓包模式中，使用python第三方库scapy进行了开发，可以实现对所有网卡的抓包。持续监听模式，利用socket网络编程相关的库函数，对特定网卡进行了抓包，由于socket模块获取网卡信息需要使用管理员模式，我们可以用管理员命令行的形式运行源码的.py文件。本次实验采用tkinter模块进行了GUI的设计。



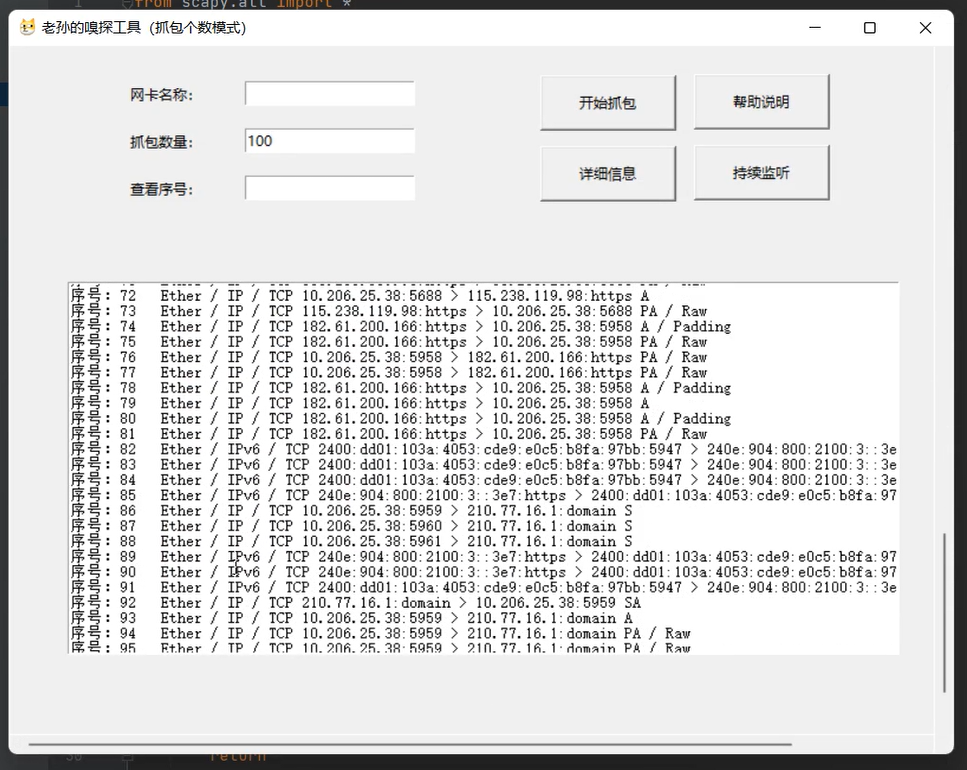
然后就会跳转至抓包个数模式的默认界面：



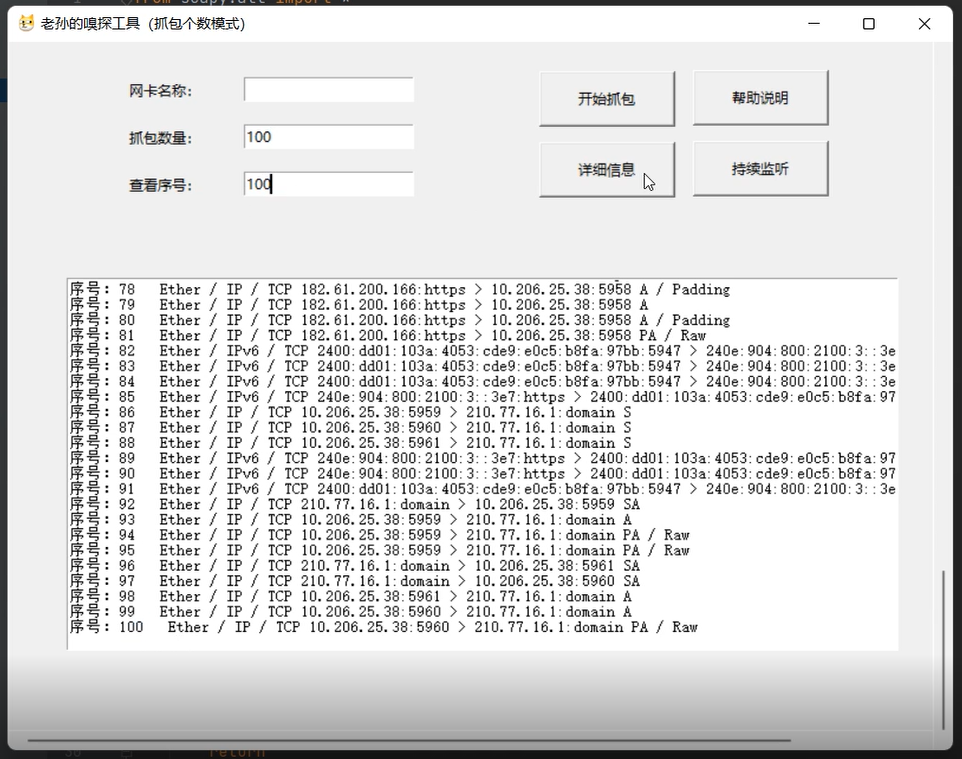
## 抓包个数模式

抓包个数模式下，输入网卡名称、抓包数量，点击开始抓包。软件会开始抓包，累计捕获输入的数量后，就会在下部的文本框进行显示。网卡名称默认为所有网卡，因此可以不填写。

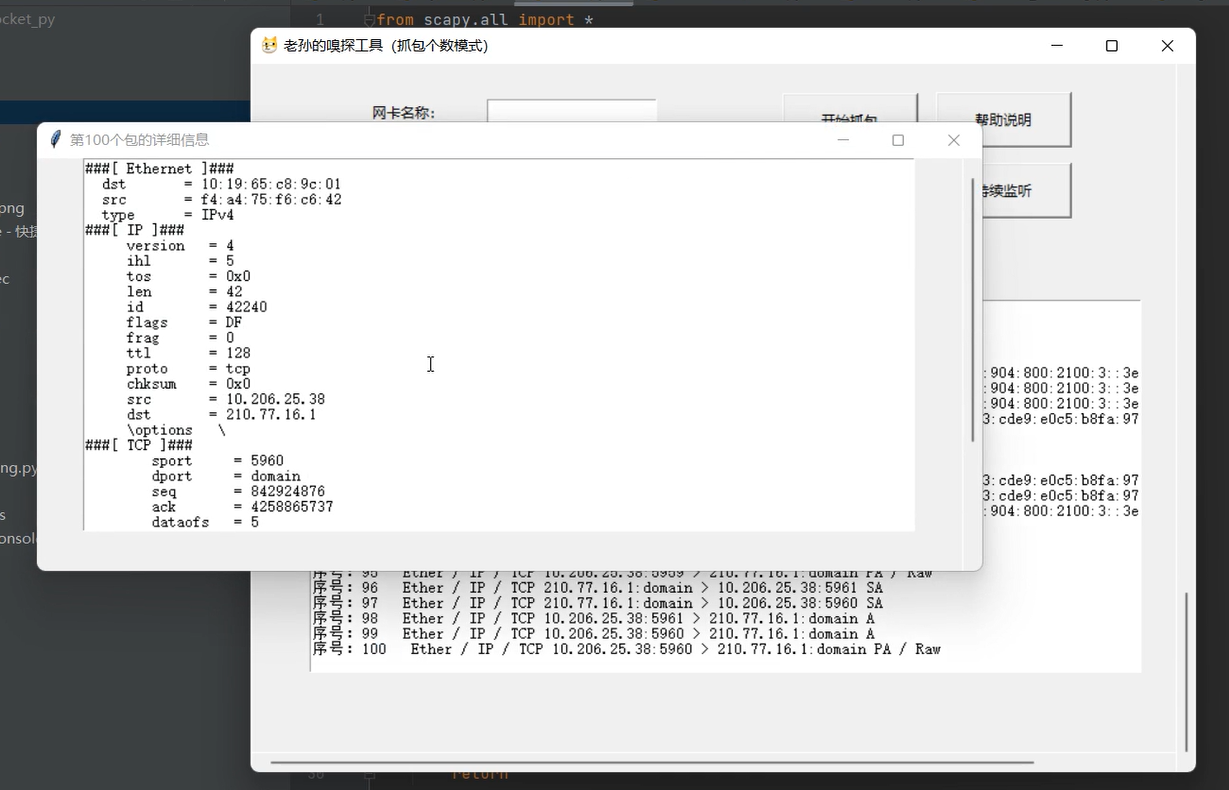
输入抓包个数后，点击开始抓包，结果如下图：



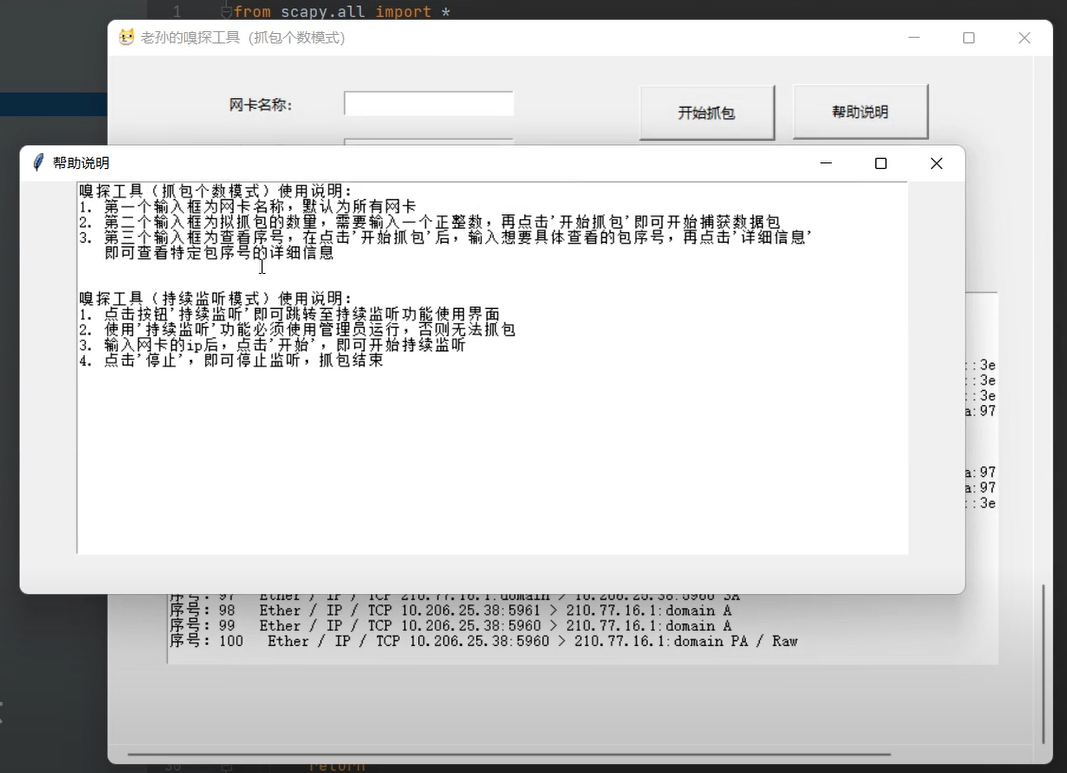
抓包结果在此已经显示成功。在查看序号模块输入想要详细查看的数据包序号，点击“详细信息”，即可弹窗显示该数据包的详细信息。



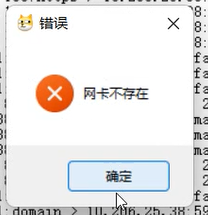
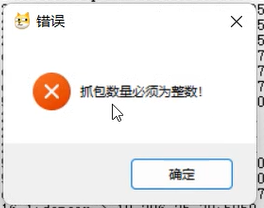
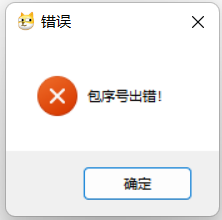
这里我们查看序号100的包信息：我们点击“详细信息”，弹窗显示如下：



可以看到，已经显示了具体的包的详细信息。在原窗口中，点击“帮助说明”可以查看软件使用说明：

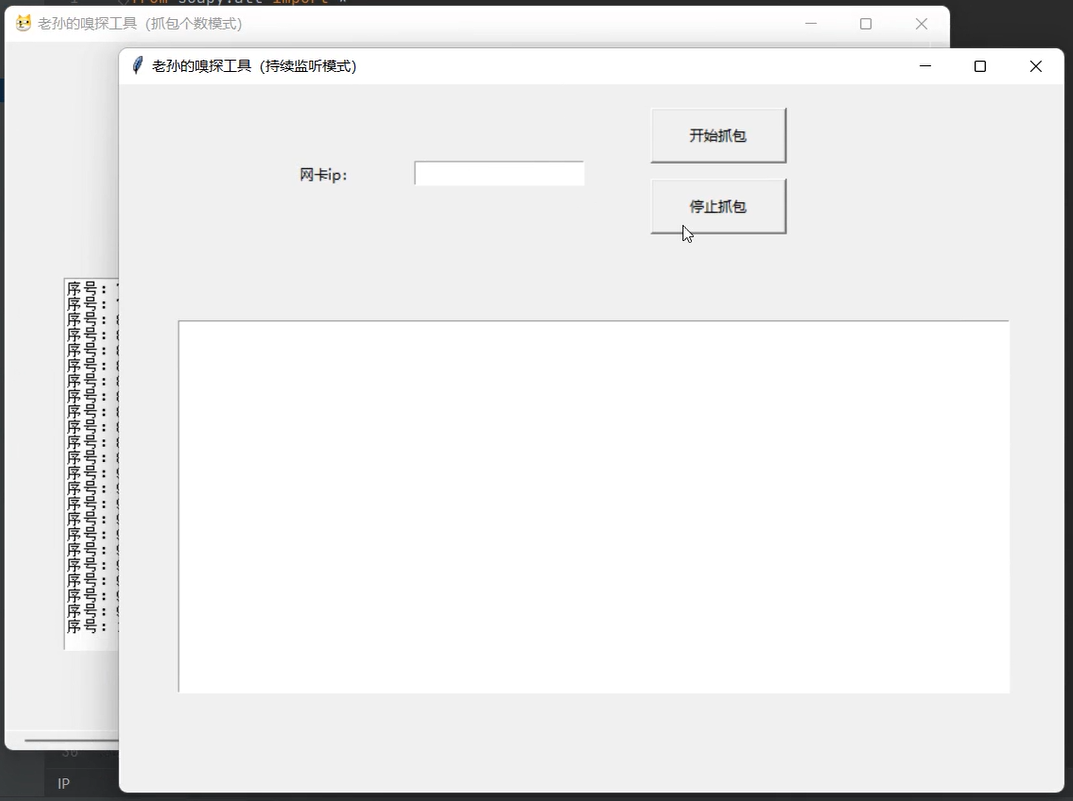


为了防止用户输入非法信息导致程序异常崩溃，我还通过python中的try: \_\_\_ except: \_\_\_模块进行了异常处理，如果网卡输入错误，会弹窗提示“网卡不存在”，如果填写抓包数量不为整数或详细查看的包序号错误，也会弹窗提醒，效果如下：

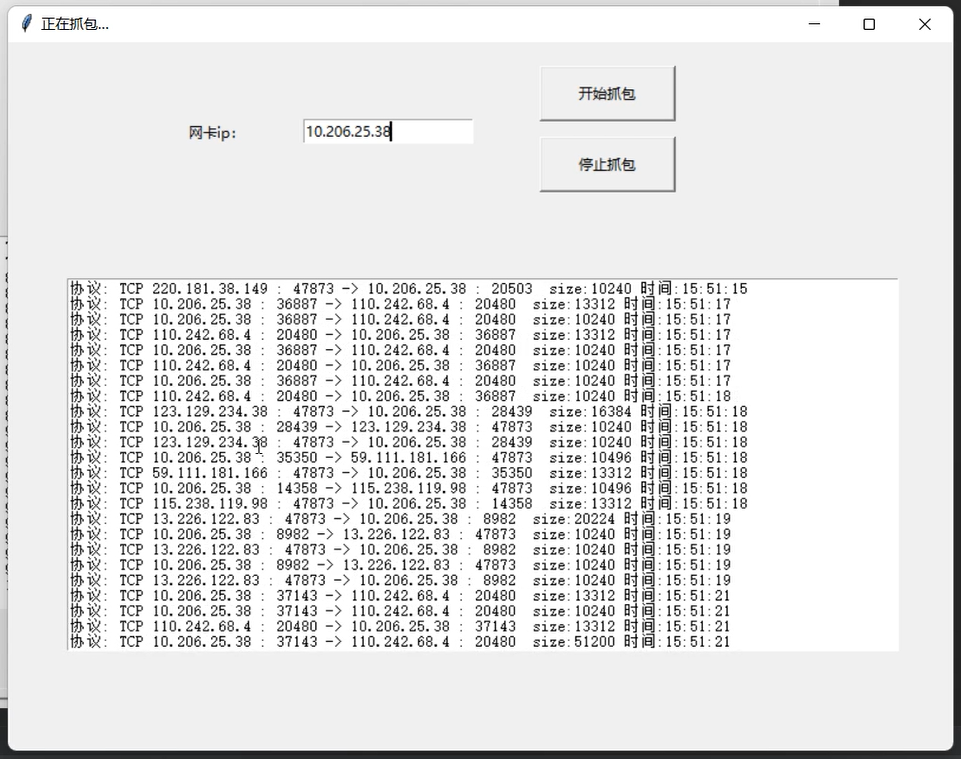
  

## 持续监听模式

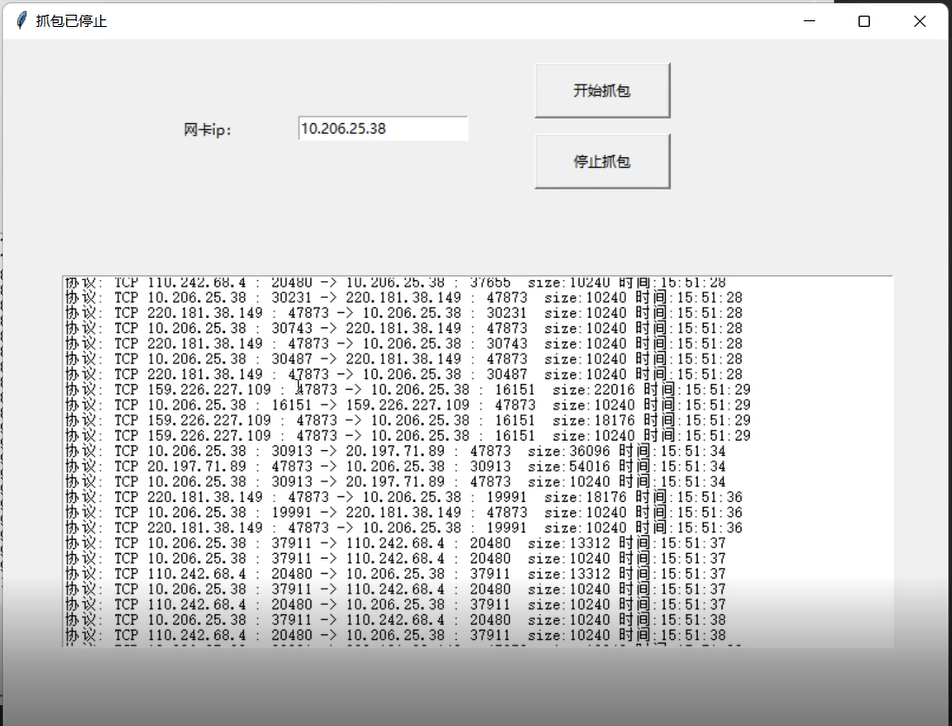
点击初始界面的“持续监听”，即可跳转至“持续监听模式”，如下图：



输入待抓包的网卡ip，点击开始抓包，即可进行持续监听。该模式使用socket网络编程模块，获取网卡信息必须要管理员权限。由于该模块是对IP数据报的首部进行解析，因此解析的协议只能是IP数据包首部协议字段可解析的协议，本次实现了TCP/UDP/ICMP的解析，如果是其他协议，则直接显示协议字段的值。抓包效果如下：



再点击“停止抓包”，抓包结束：



## 心路历程、收获总结与讨论

由于我编程水平比较垃圾，再加上python的开源库函数较为丰富，于是我首选python作为嗅探器的实现语言。经过我持续的奋斗，改bug，被各种问题缠身几日后，终于用python实现了一个嗅探器。初拿这个题目，我首先想到了今日翻看过的一本书《Python黑帽子：黑客与渗透测试编程之道》这本书，里面有关于socket编程和scapy模块的详细描述可供参考。

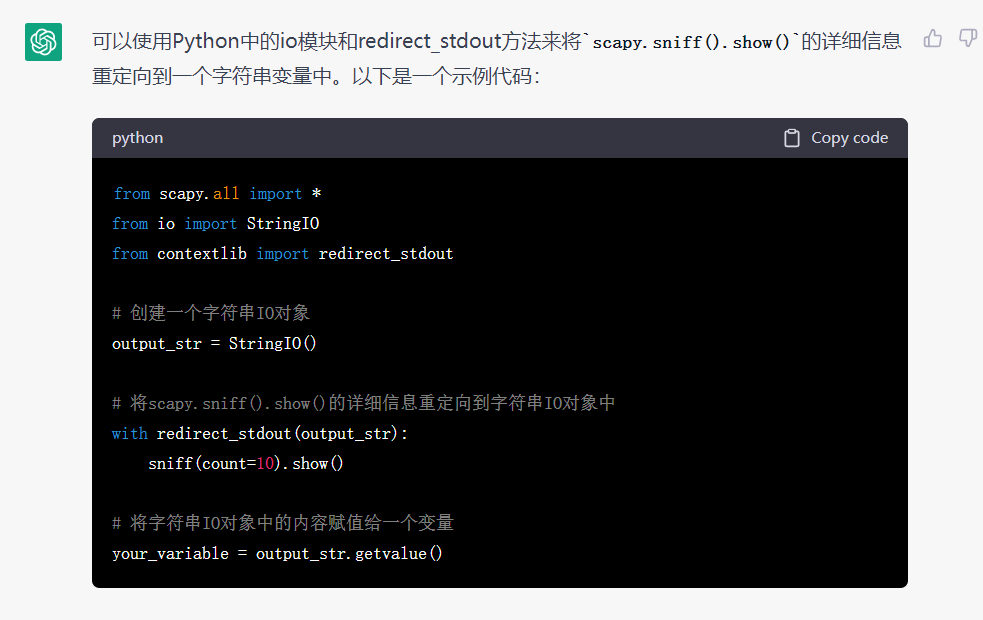
我首先对scapy库函数进行了了解。发现这个第三方库非常强大，基本上已经实现了嗅探的脚本。可以显示包的大致信息和详细信息。运用该库中的sniff函数即可实现嗅探。获得包列表：

packages = sniff(iface=var1, count=eval(var2))

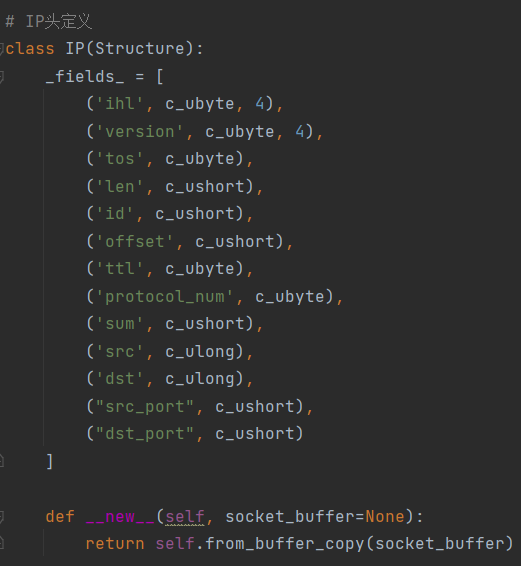
然后还可以用下列语句获得包的详细信息，将详细信息输出到控制台：

packages[i].show()

我感觉那我只需要写一个UI界面，传递一下参数好像就可以解决问题了，于是我就使用tkinter库进行了UI界面的编写。在输出详细信息时，遇到了一个问题：scapy里面的.show()方法好像只能将包的详细信息输出到控制台，无法赋值给变量。于是我又网上搜索解决方案，最终chatgpt给我了一个可以实现的方法：



然后第一个模块就基本完成，我转念一想，我写的这程序不是纯粹套壳嘛！就是调用了现成的接口，设计了个UI界面，并没有对包的帧首部/包头进行解析，感觉并没有达到这个实验的核心目的。于是我便想到了socket网络编程模块，同样参照了《Python黑帽子：黑客与渗透测试编程之道》这本书，对IP数据包的报文首部格式进行了定义：



然后再创建进程，套接字绑定地址，创建线程，并用如下语句获取数据包，并读取首部：

raw\_buffer = sniffer.recvfrom(65535)[0]

ip\_header = IP(raw\_buffer[0:24])

再对首部的各自段进行了解析，这才实现了对数据包的分析。比较遗憾的是，由于是对IP数据包的首部进行解析，可解析的协议仅仅是IP数据报文首部协议字段的值，所以无法解析将HTTP/DNS/FTP等这种应用层协议。不过最终也实现了完整的功能，还是比较有成就感的。这是在后续可以进一步改进的。

在设计UI界面的时候，我去图书馆找了一本关于tkinter库使用的工具书，对着书敲了一下午代码，大概了解了UI界面的编写思路。在完成了嗅探核心功能后，再修改代码，加入了交互的UI界面设计部分。最终完成了整个实验。通过实验，我的编程能力有了一定提升，同时对socket编程、创建套接字的流程也有了一定了解，同时也了解了强大的scapy模块，并对UI界面设计的python实现也有了一定认识，感觉还是有所收获的。