《软件安全》实验报告

姓名：张丛 学号：2113662 班级：信安一班

**实验名称：**

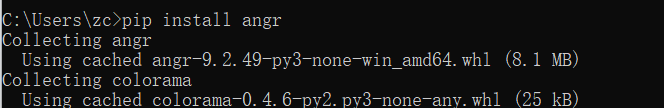
Angr应用示例

**实验要求：**

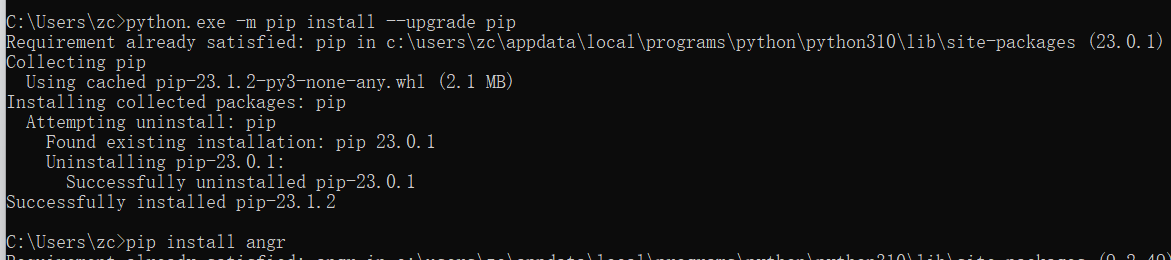
根据课本8.4.3章节，复现sym-write示例的两种angr求解方法，并就如何使用angr以及怎么解决一些实际问题做一些探讨。

**实验过程：**

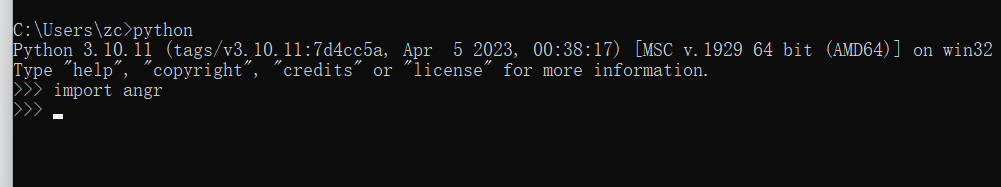
1. 安装python3,安装Angr



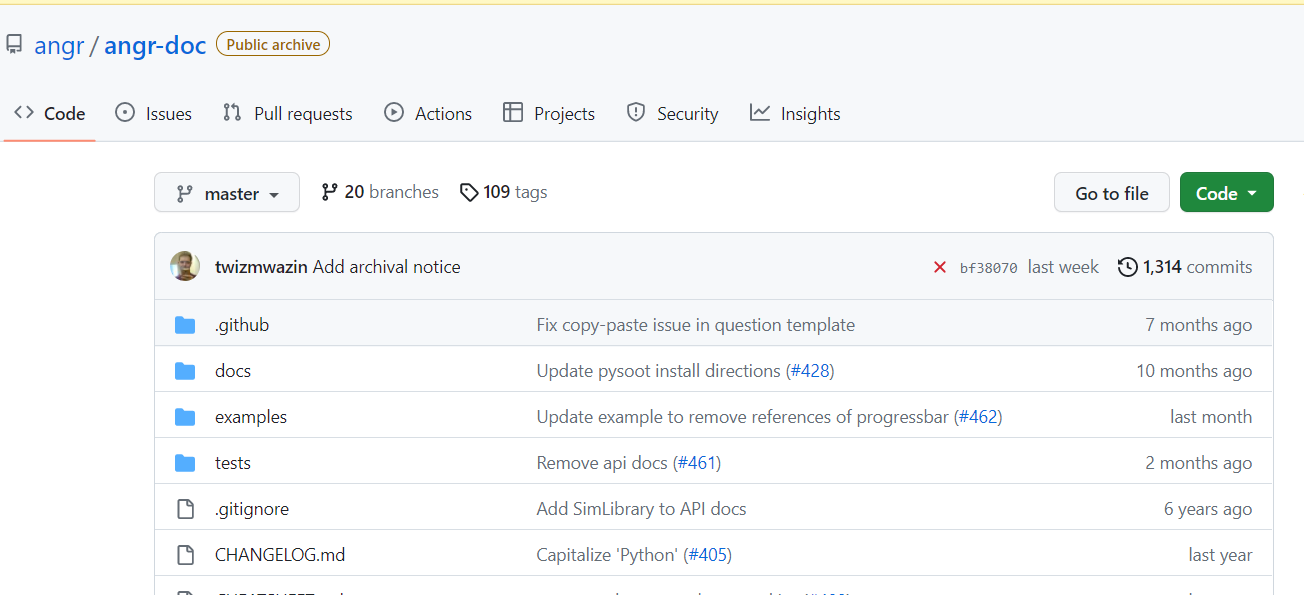
中途不能成功安装Angr，需要更新pip:



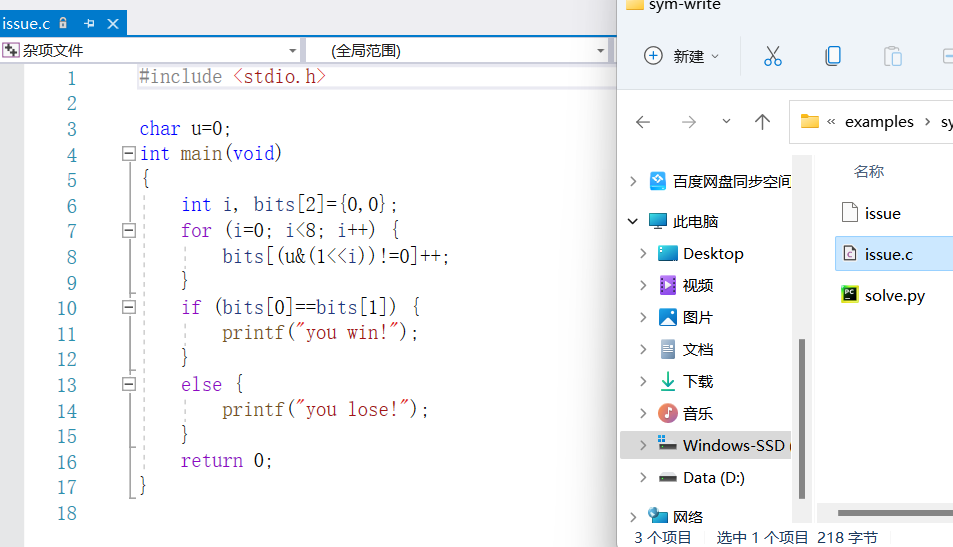
成功安装Angr:



1. 下载angr-doc-master文件



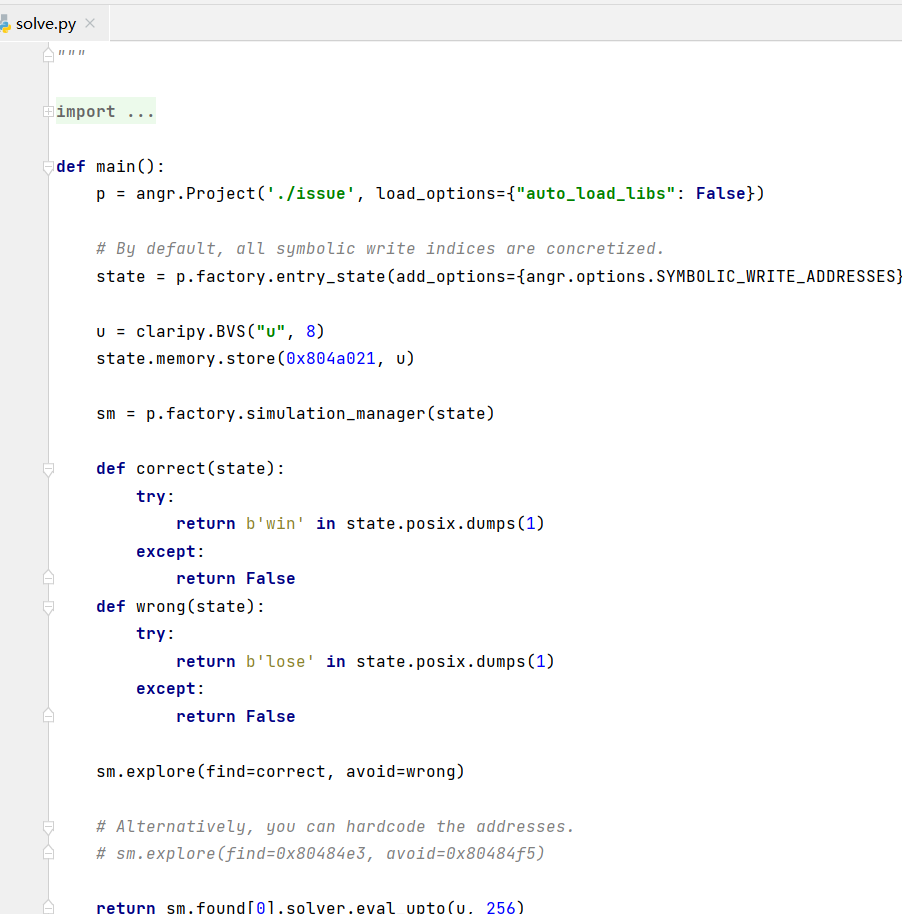
1. 查看函数



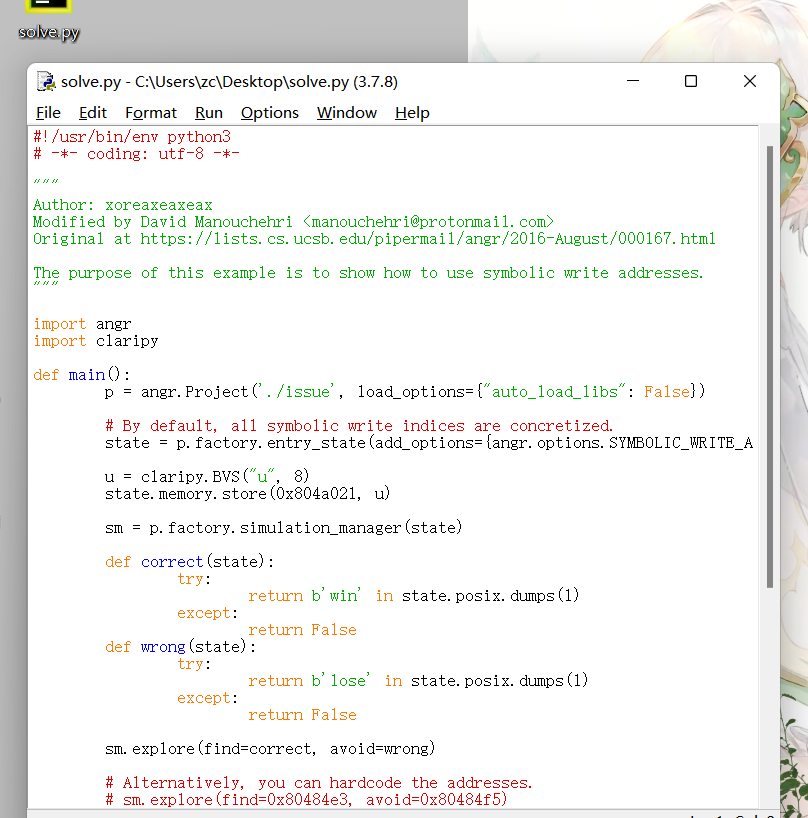
目标是，破解哪一个u的值可以输出”you win”

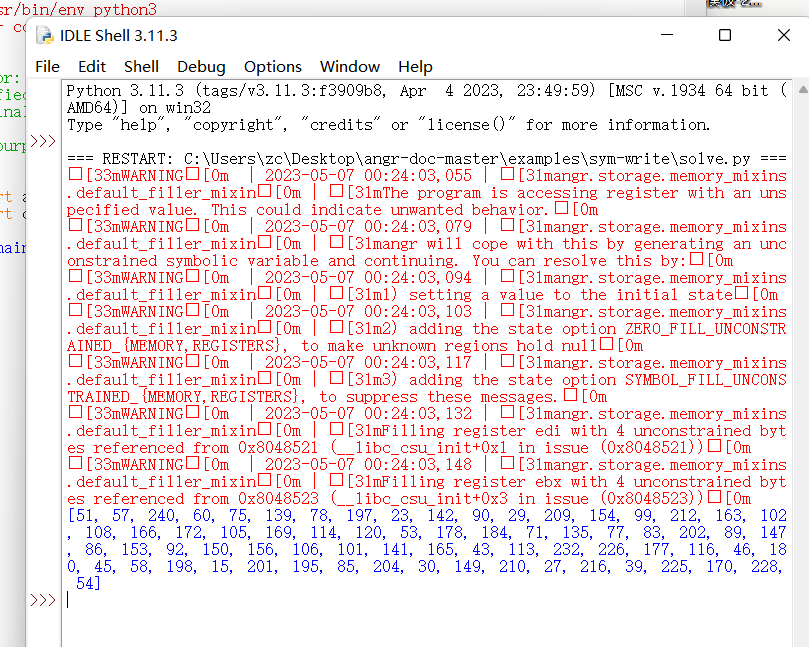
1. 利用Angr，符号查找

代码如下，注释不再赘述



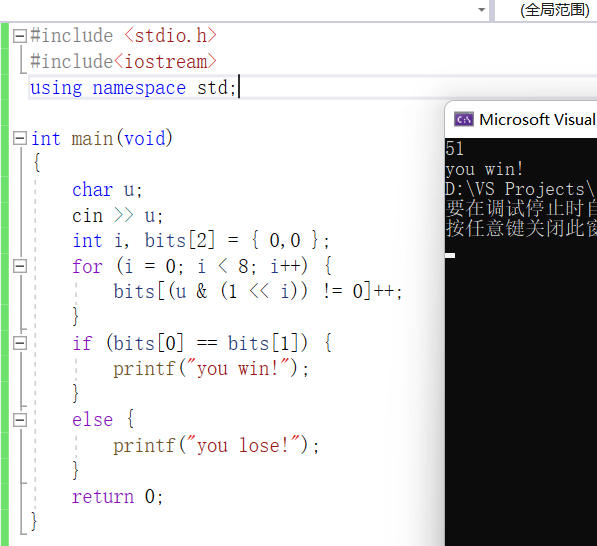
使用IDLE运行：

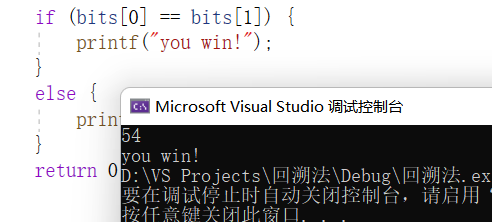




可以看到，当u取值：51,57……54时，将会输出“you win!”

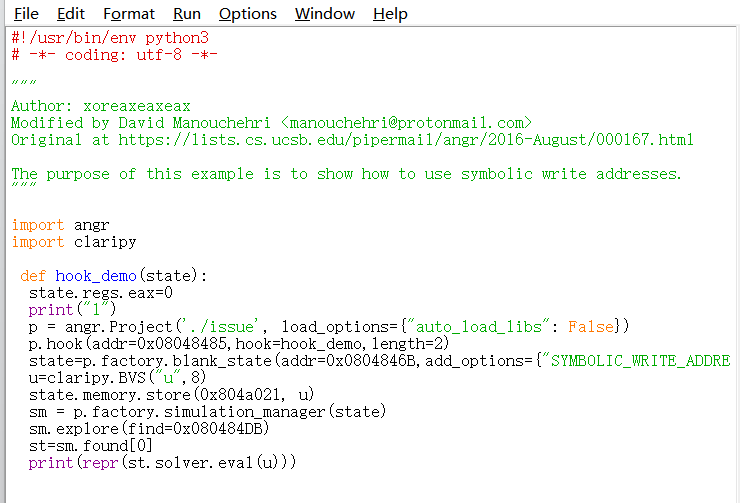
在VS上验证：





可以验证实验成功

1. 方法二



同样可以进行破解

**心得体会：**

**实验难点：**

配置环境，angr安装（因为python版本、pip版本）；

理解求解的代码，需要结合课上内容

**利用angr解决实际问题：**

1. 漏洞利用：Angr可以用于搜索二进制代码中存在的漏洞，然后通过构造输入以利用这些漏洞。

2. 安全性检查：Angr可以用于静态分析二进制代码以确定其中的安全问题，如缓冲区溢出、格式化字符串漏洞等。

3. 反汇编和反编译：Angr可以将二进制代码反汇编为汇编代码，或反编译为高级语言代码。

4. 二进制重定向：Angr可以用于分析二进制代码中包含的函数和数据结构，并对它们进行重定向，以便进行增量分析或其他分析技术。

5. 模糊测试：Angr可以用于生成有效的测试用例，以测试二进制代码的稳定性和安全性。

**利用angr步骤：**

1. 运行程序：我们在得到一个程序时，首先需要对此程序创建一个Angr工程：p = angr.Project(‘program’)

(1) 命令行参数

(2) 标准输入

2. Angr中程序的几种状态：状态在Angr中表示着程序符号执行后的几种结果，在Angr中，当获取到程序入口点的状态后，我们需要使用Angr的Simgr模拟器来进行符号执行

3. 获取输出