

**实验报告**



**题目： 拆解二进制炸弹**

**班 级： 2023211307**

**学 号： 2023211075**

**姓 名： 魏生辉**

**学 院： 计算机学院**

**202 年 11 月 10 日**

一、实验目的  
1.理解C语言程序的机器级表示。  
2.初步掌握GDB调试器的用法。  
3.阅读C编译器生成的x86-64机器代码，理解不同控制结构生成的基本指令模式，过程的实现。

1. 实验环境
2. Windows PowerShell（10.99.0.230）
3. Ubuntu 2204.2.33.0
4. Objdump命令反汇编
5. GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1

三、实验内容

登录bupt1服务器，在home目录下可以找到Evil博士专门为你量身定制的一个bomb，当运行时，它会要求你输入一个字符串，如果正确，则进入下一关，继续要求你输入下一个字符串；否则，炸弹就会爆炸，输出一行提示信息并向计分服务器提交扣分信息。因此，本实验要求你必须通过反汇编和逆向工程对bomb执行文件进行分析，找到正确的字符串来解除这个的炸弹。

本实验通过要求使用课程所学知识拆除一个“binary bombs”来增强对程序的机器级表示、汇编语言、调试器和逆向工程等方面原理与技能的掌握。 “binary bombs”是一个Linux可执行程序，包含了5个阶段（或关卡）。炸弹运行的每个阶段要求你输入一个特定字符串，你的输入符合程序预期的输入，该阶段的炸弹就被拆除引信；否则炸弹“爆炸”，打印输出 “BOOM!!!”。炸弹的每个阶段考察了机器级程序语言的一个不同方面，难度逐级递增。

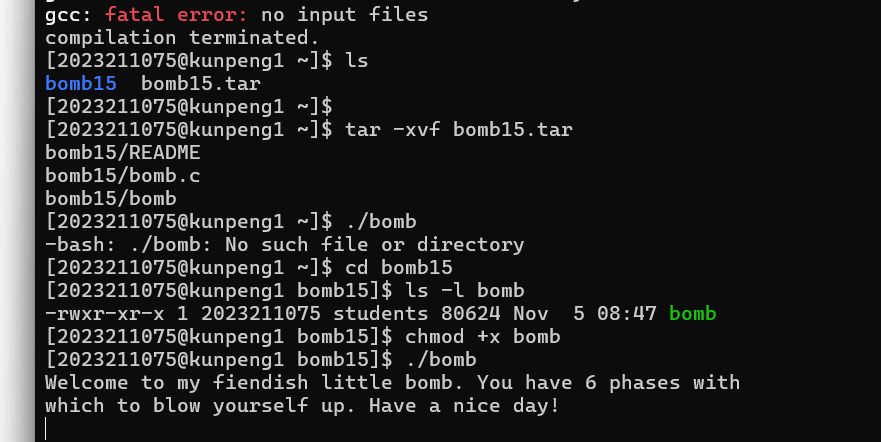
为完成二进制炸弹拆除任务，需要使用gdb调试器和objdump来反汇编bomb文件，可以单步跟踪调试每一阶段的机器代码，也可以阅读反汇编代码，从中理解每一汇编语言代码的行为或作用，进而设法推断拆除炸弹所需的目标字符串。实验2的具体内容见实验2说明。

四、实验步骤及实验分析

**（六个阶段+隐藏阶段）**

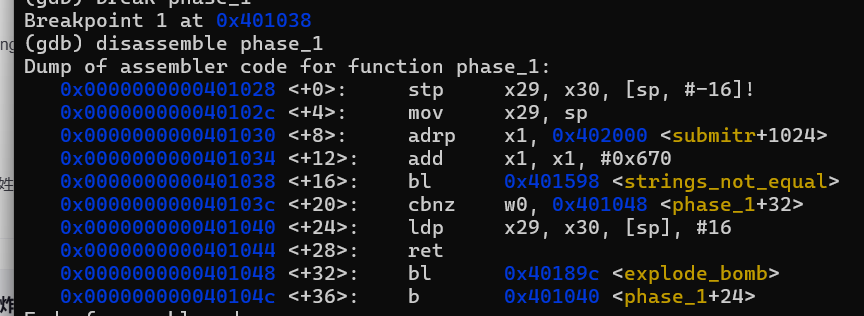
**准备工作**

**先获取炸弹**

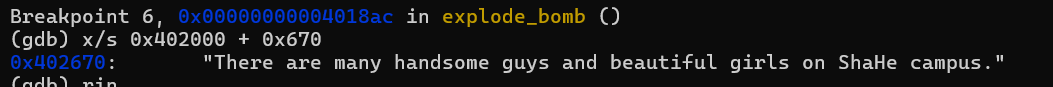
****

**阶段一**

**①先打个断点，并看一下汇编**

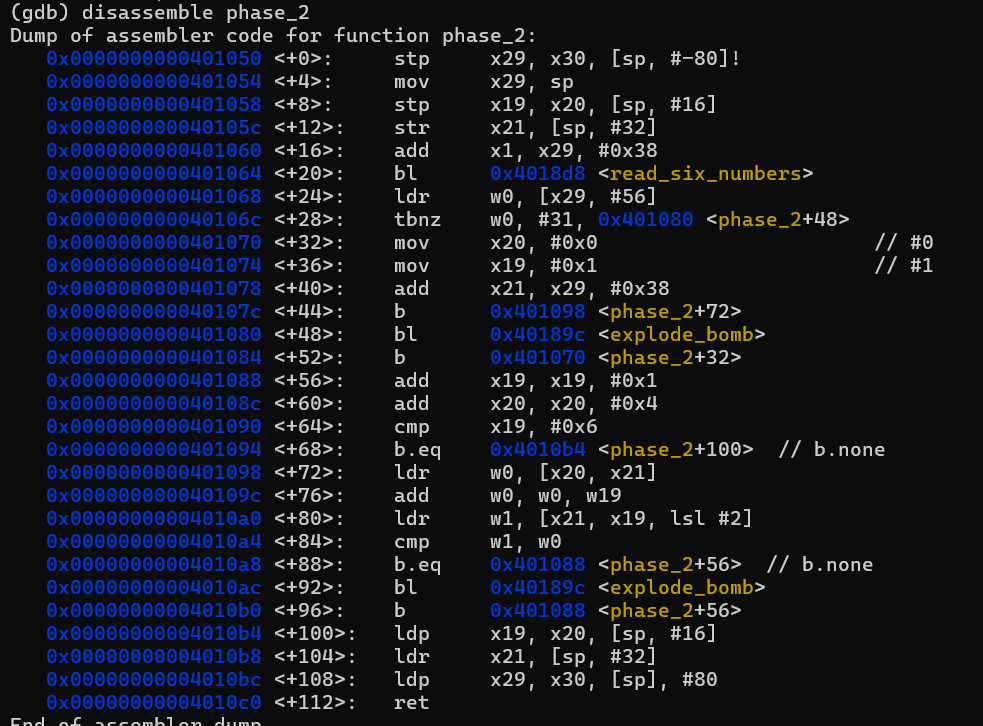


**②确定预期字符串**



**阶段二**

**①先设置断点并反汇编**



**②分析汇编代码并总结 phase\_2 输入要求**

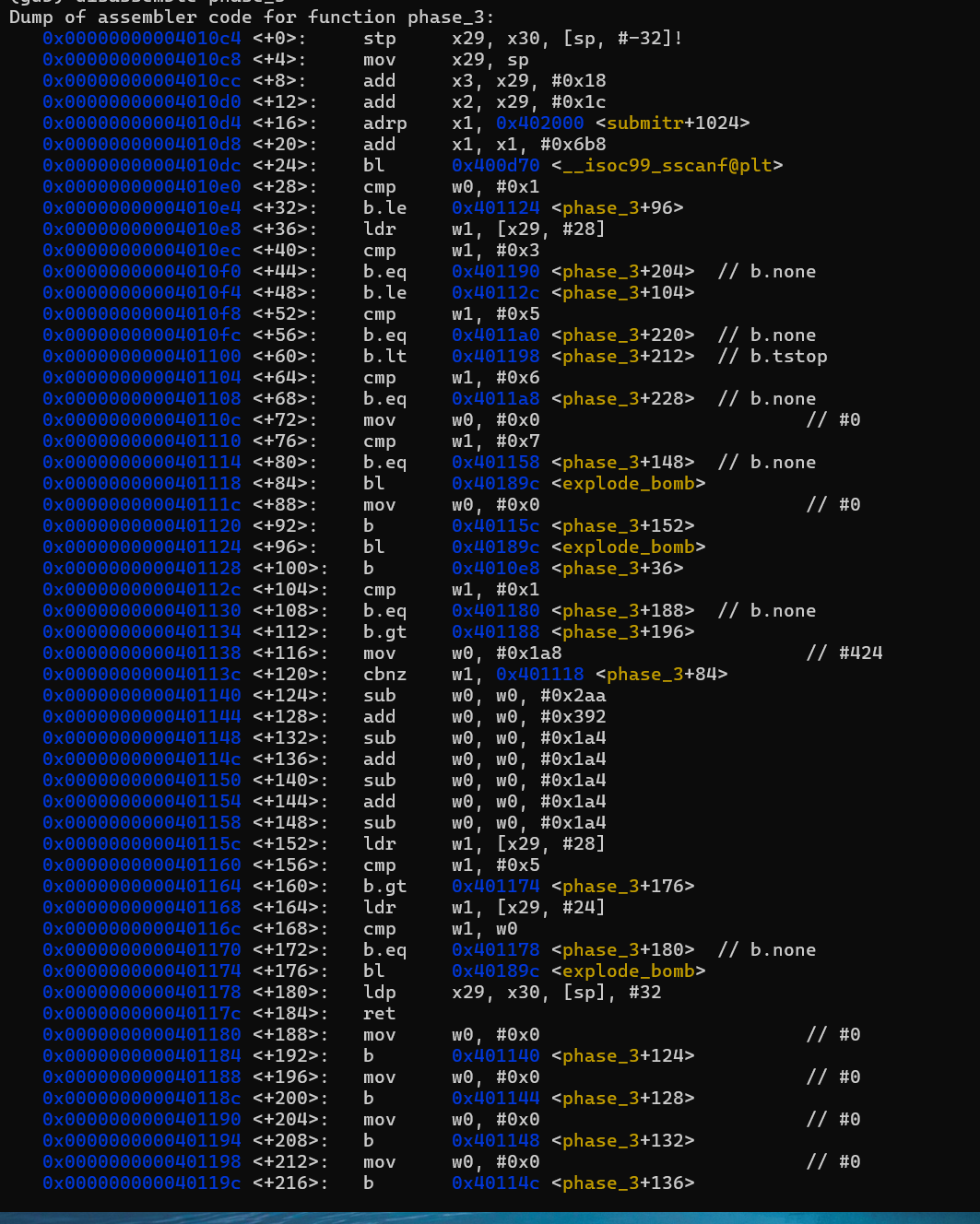
其实和普通版的实验很像，从2开始依次加1 加2 加3

2 3 5 8 12 17

**③通过**

**阶段三**

**①先设置断点并反汇编**

****

**②分析汇编代码**

**0x00000000004010e8 <+36>: ldr w1, [x29, #28]**

**- 将从输入中读取的第一个整数（`val1`）加载到 `w1` 中。**

**- 该值存储在 `[x29 + 28]`，对应于变量 `val1`。**

**函数根据 `val1` 的值使用一系列比较和分支。逻辑如下：**

**情况 `val1 == 0`**

**- 跳转到一段代码，`w0` 被设置为 `424`，然后执行一系列算术运算，最终 `w0 = 236`。**

**- 之后，将第二个输入值 `val2` 与 `236` 比较。**

**情况 `val1 == 1`**

**- `w0` 被设置为 `0`，执行计算，最终 `w0 = -188`。**

**- 比较 `val2` 与 `-188`。**

**情况 `val1 == 2`**

**- 计算得到 `w0 = 494`。**

**- 比较 `val2` 与 `494`。**

**情况 `val1 == 3`**

**- 计算得到 `w0 = -420`。**

**- 比较 `val2` 与 `-420`。**

**情况 `val1 == 4` 或 `val1 == 5`**

**- `w0` 最终为 `0`。**

**- 比较 `val2` 与 `0`。**

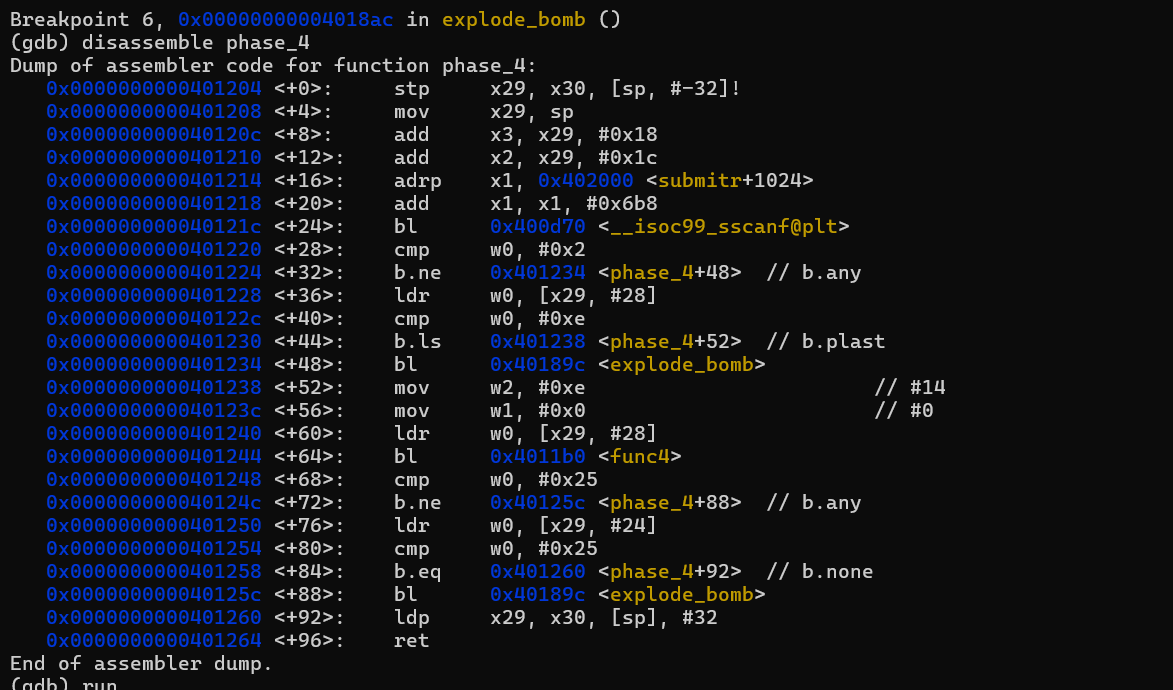
**③确定答案**

**0 236**

**④通过**

**阶段四**

**①设置断点和反汇编**

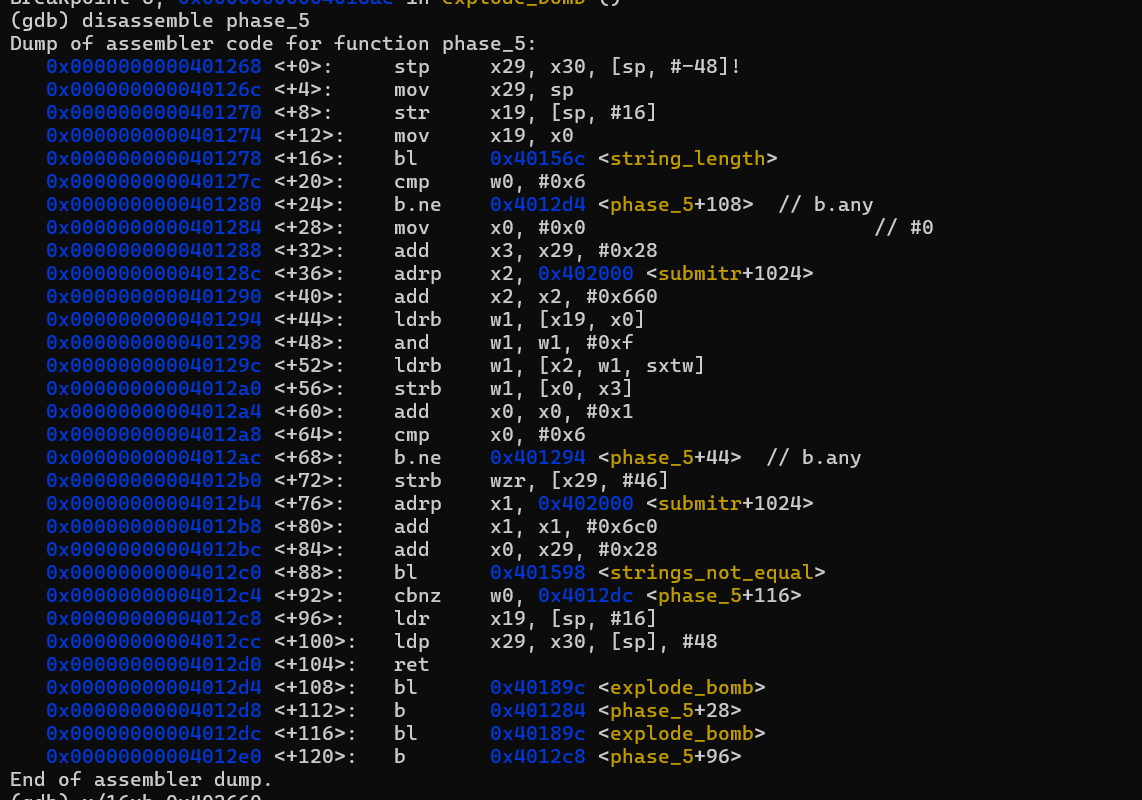
****

**②分析**

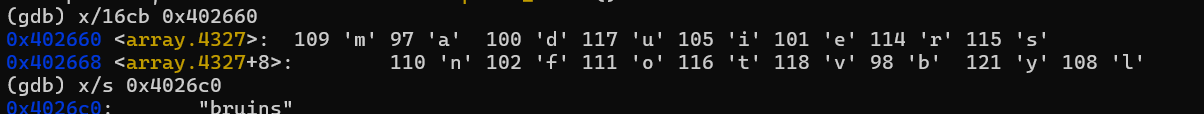
其实和普通版本的一模一样，就是一个二分查找，返回一个固定的函数值，轻松解出来。

**阶段五**

**①设置断点和反汇编**

****

**②查找表内容（地址 0x402660）和目标字符串（地址 0x4026c0）**

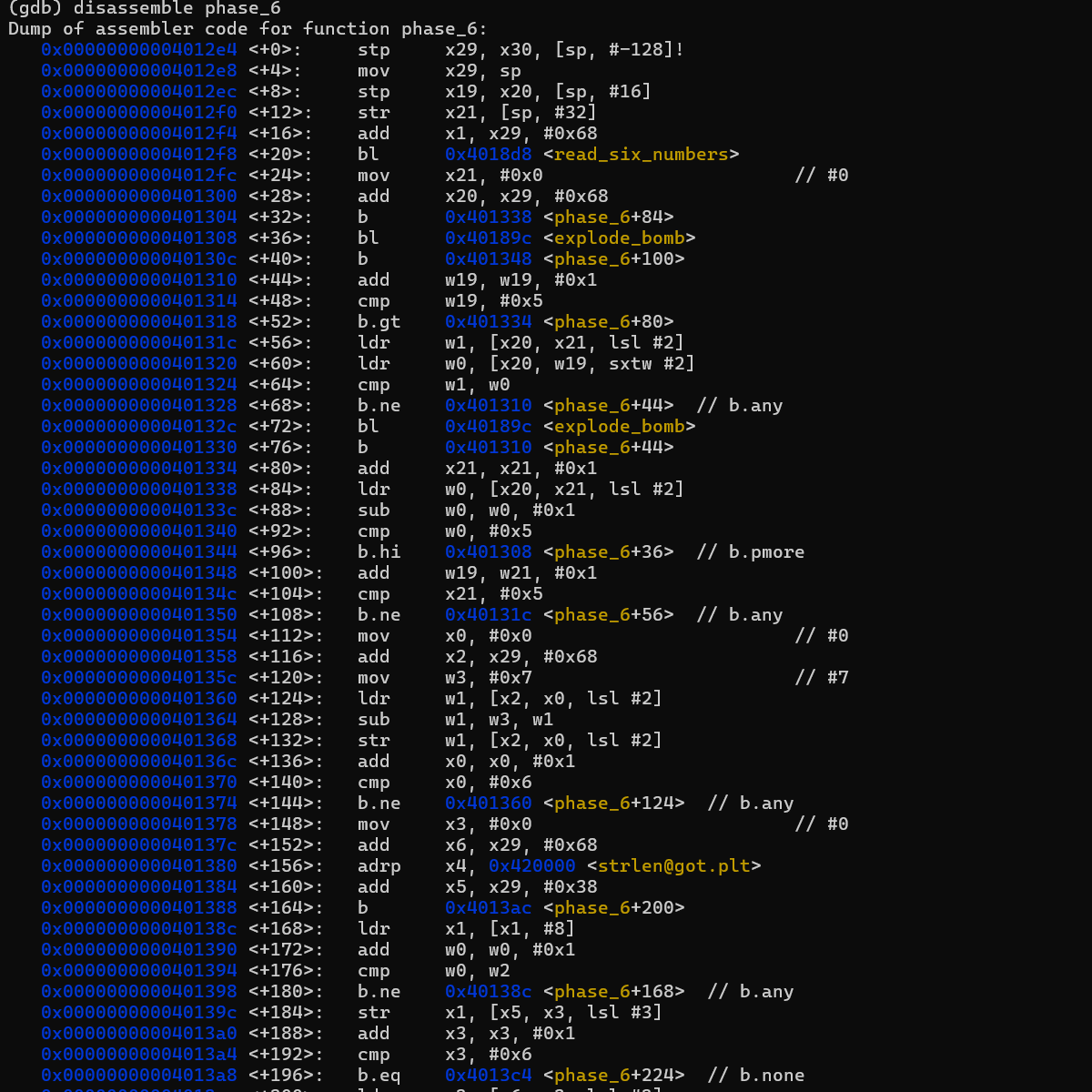


**③由目标字符与查找表索引的对应关系得出最后的答案**

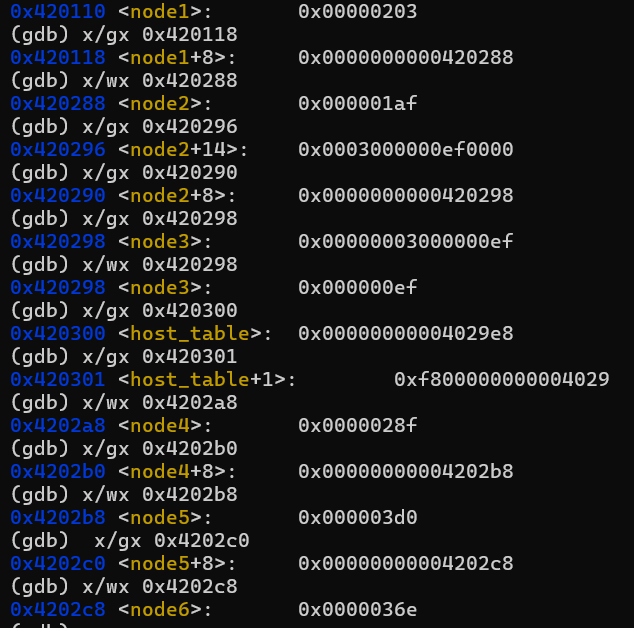
mfsthw

**阶段六**

**①设置断点和反汇编**



**②获取节点值**



按节点值降序排序

根据节点的值从大到小排序如下：

节点 5（值：976）

节点 6（值：878）

节点 4（值：655）

节点 1（值：515）

节点 2（值：431）

节点 3（值：239）

**③计算原始输入数字**

计算原始输入数字：

7 - 5 = 2

7 - 6 = 1

7 - 4 = 3

7 - 1 = 6

7 - 2 = 5

7 - 3 = 4

**阶段七：隐藏阶段**

**①首先找到入口**

第四个输入后面加上 DrEvil

**②做完普通版本就知道了是二叉树**

**③构造二叉树**

36

8 50

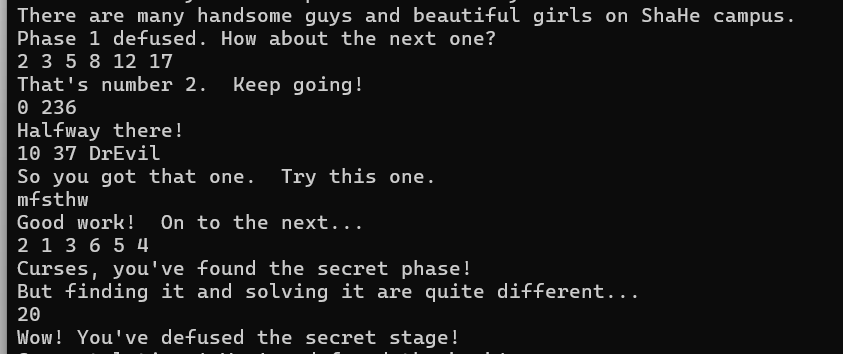
6 22

1 7 20 35

**④反演**

结果是20

**⑤全部通过**



1. 总结体会

1.老问题：太心急，最开始没设置好断点，炸了一次；

2.对汇编指令不熟悉：又忘记了怎么监视；

3.很有成就感：从最开始的不知如何下手到现在三小时完全过关，感觉自己有成长；

4.最后感谢老师和助教给我这次尝试和学习的机会！