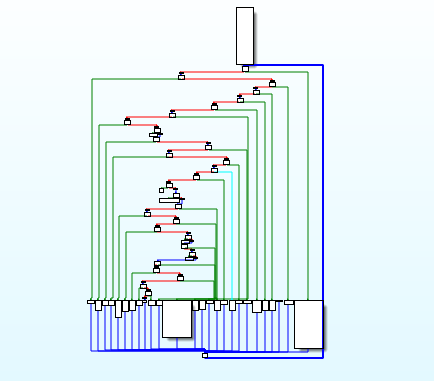
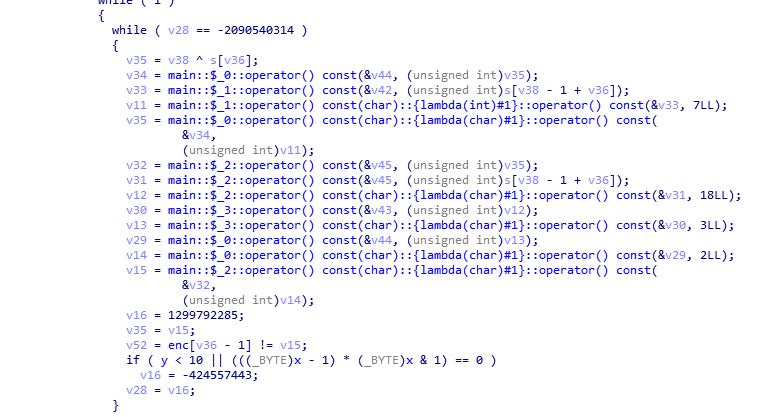
# 题目分析

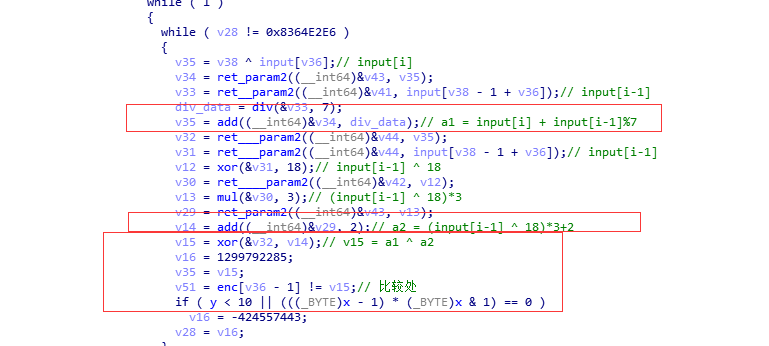
将程序拖进ida后发现是很标准的ollvm的控制流平坦化。



我们找到关键算法部分发现是C++写的，但是点进这些函数会发现并不难分析。



分析后的结果。



其中ret\_param2就是返回参数2，其它的就是些加，乘，除，xor操作了。

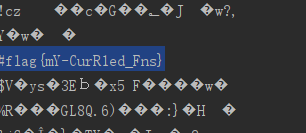
然后可以判断程序逻辑就是通过输入得到a1，和a2，然后v15 = a1 & a2，

然后将v15和enc[i]对比。

现在我们知道了程序的处理流程，但是我们并不知道flag的前几个字母是什么，我们选择进行爆破。

# 脚本

byte = lambda x : x & 0xff  
enc = [243, 46, 24, 54, 225, 76, 34, 209, 249, 140, 64, 118, 244, 14, 0, 5, 163, 144, 14, 165]  
for i in range(33,127):  
 flag = [0] \* (len(enc)+1)  
 flag[0] = i  
 for j in range(len(enc)):  
 flag[j+1] = byte(byte(enc[j] ^ (byte((flag[j] ^ 18) \* 3 + 2))) - flag[j] % 7)  
 print "".join(map(chr,flag))



可以发现flag为flag{mY-CurR1ed\_Fns}