Android

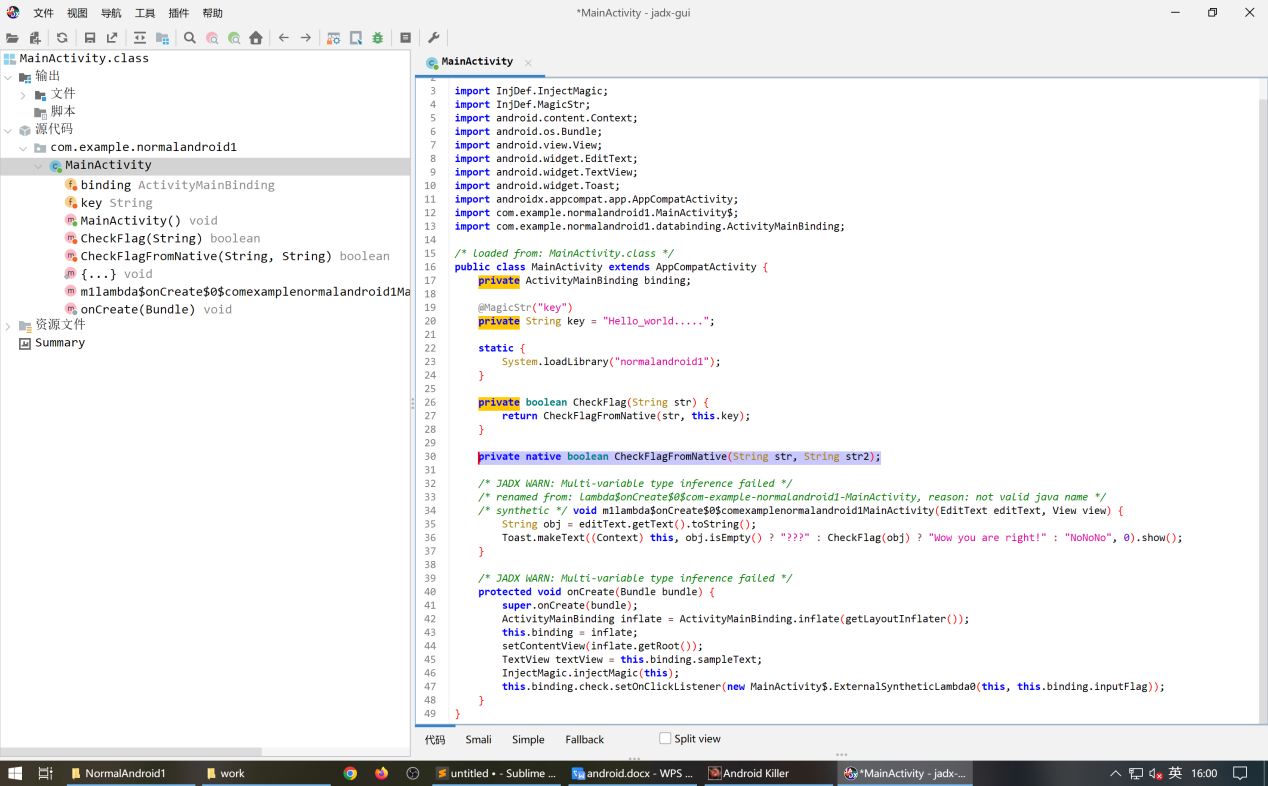
解包反汇编反编译后看一下，大概流程就是调用一个native方法，参数是key和输入的字符，通过native的checkflag返回是否正确。

23行导入so

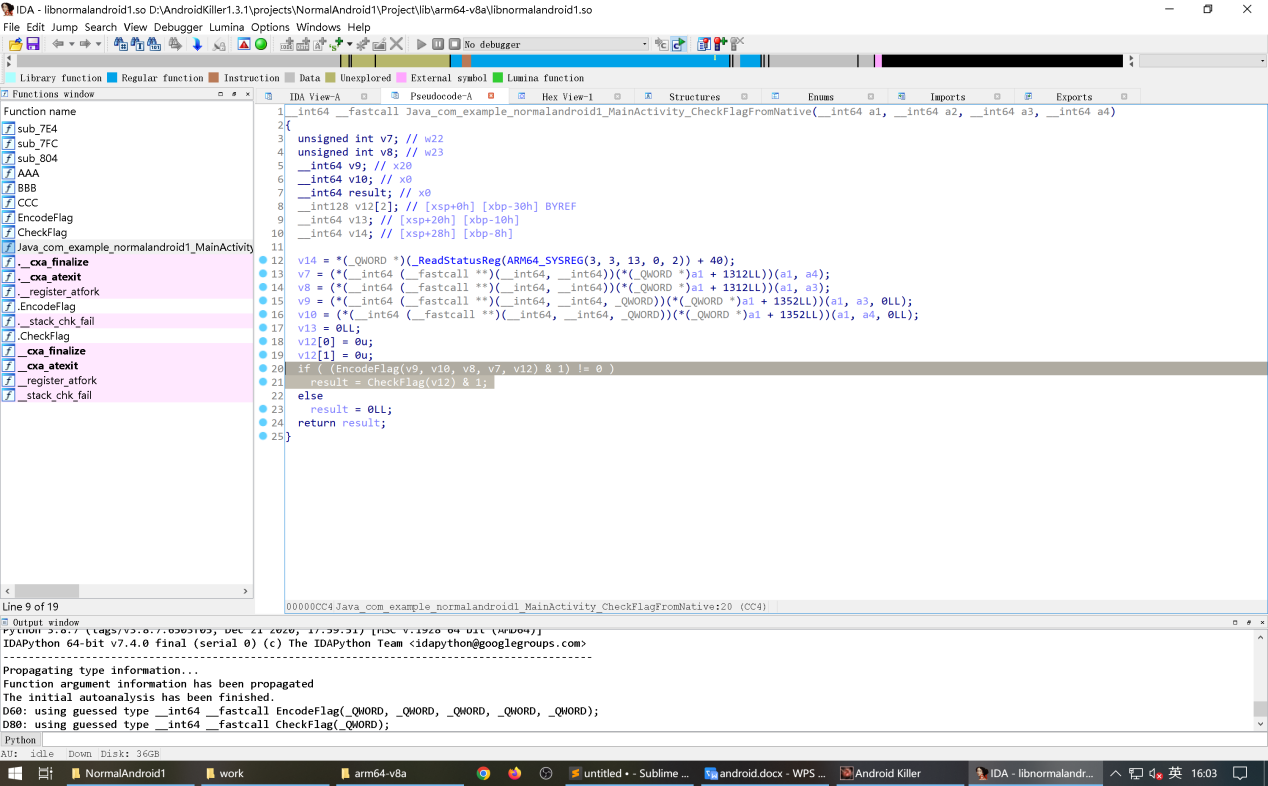
30行声明native方法

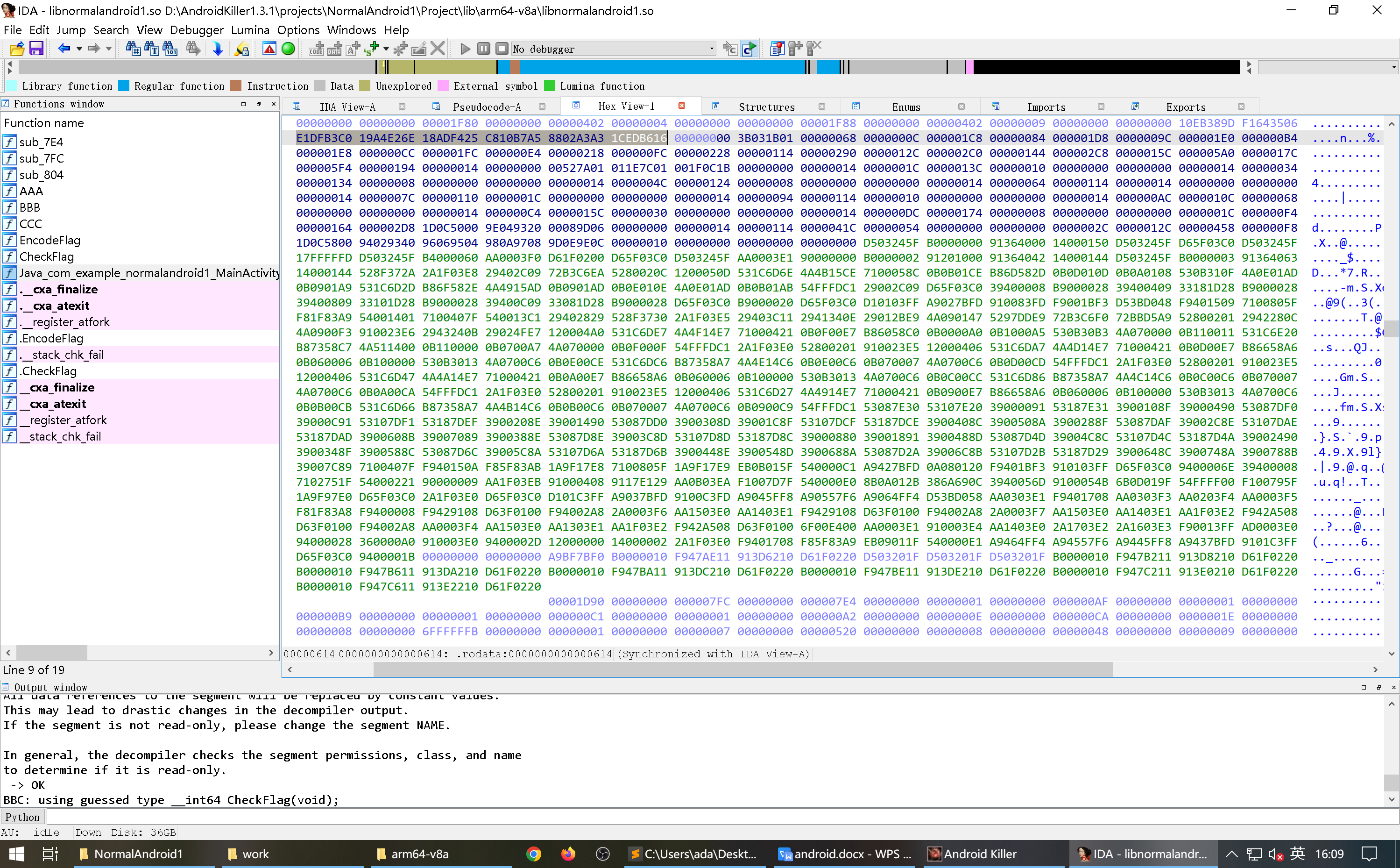
36行调用逻辑

注意第19行修饰符 key实际上是I 10v3 D4SCTF!!! 而不是helloworld...



IDA看一下逻辑也比较简单，调用encode 和check ，check将传入参数与内部存的数据对比，主要看encode



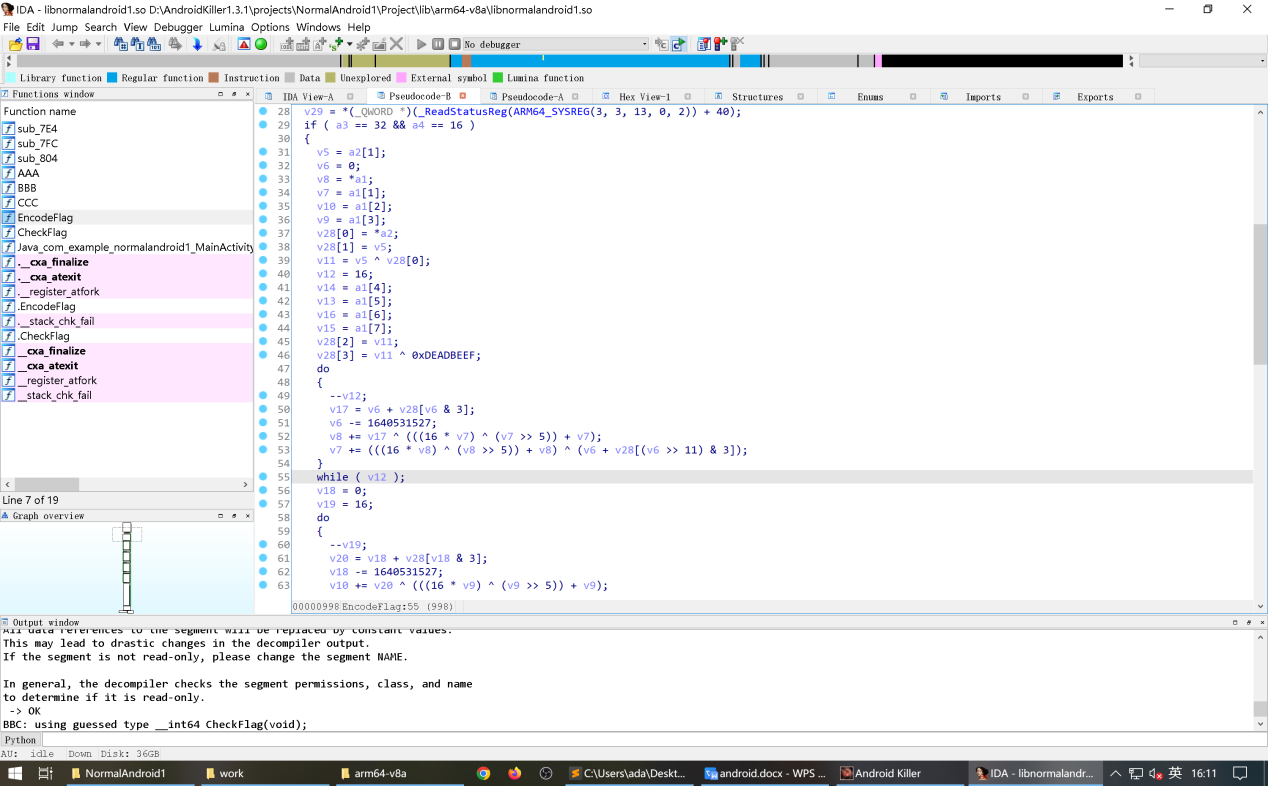


最后的check就是和这些数据对比

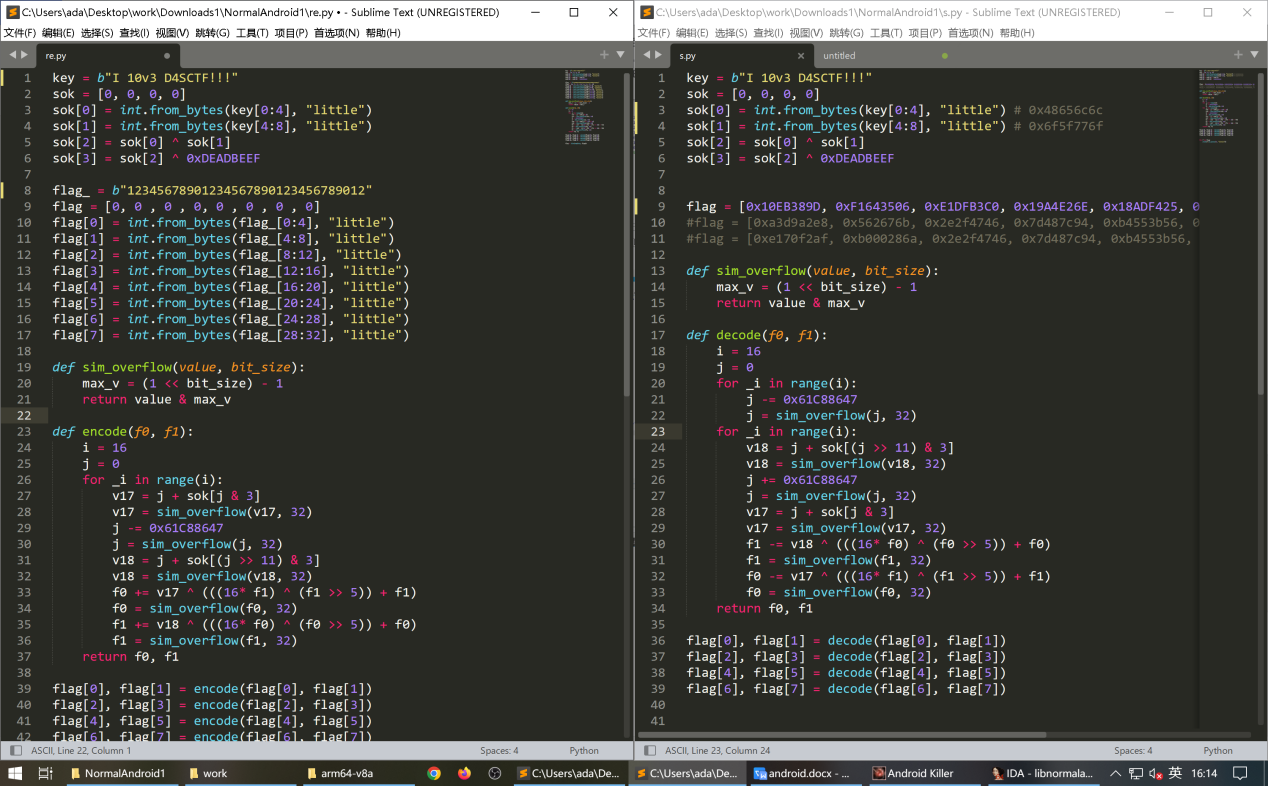
看encode，29可以知道key长16 flag长32

31到46行对输入的key和flag做一些变换，主要是拆分成4字节的数组

47到57是一组循环，下面的也是一样，总共四组，将encode逻辑python代码写出来如下

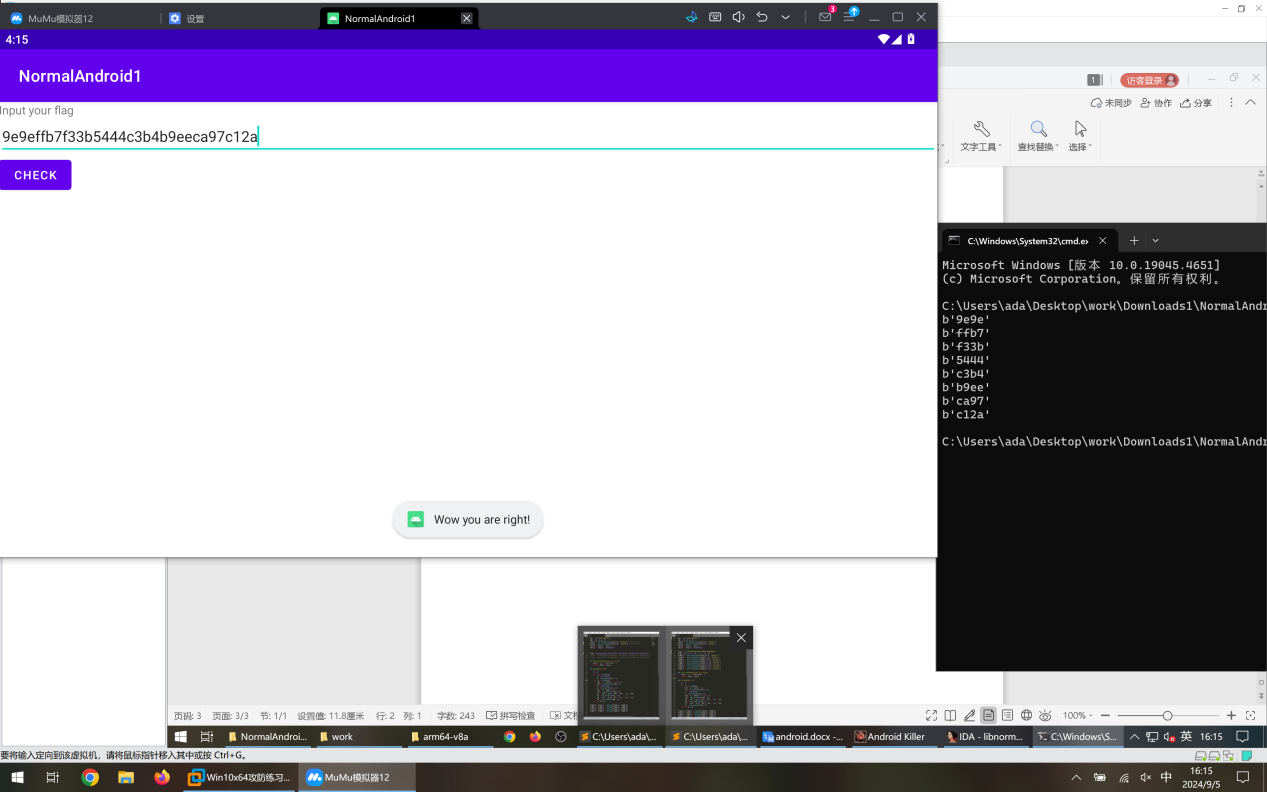


下面两个脚本左边是模拟encode逻辑，右边是将encode逻辑反向解密,实用python的时候要注意手动处理一下溢出，如果写c的话就不用了



最后将解出来的flag放到模拟器里试一下，you are right。

最后加上DASCTF{}



这个题目走了比较多弯路，对于数据在内存中的存储要注意端序，同时也先不用着急动调，ida的伪代码慢慢看还是比较容易的。