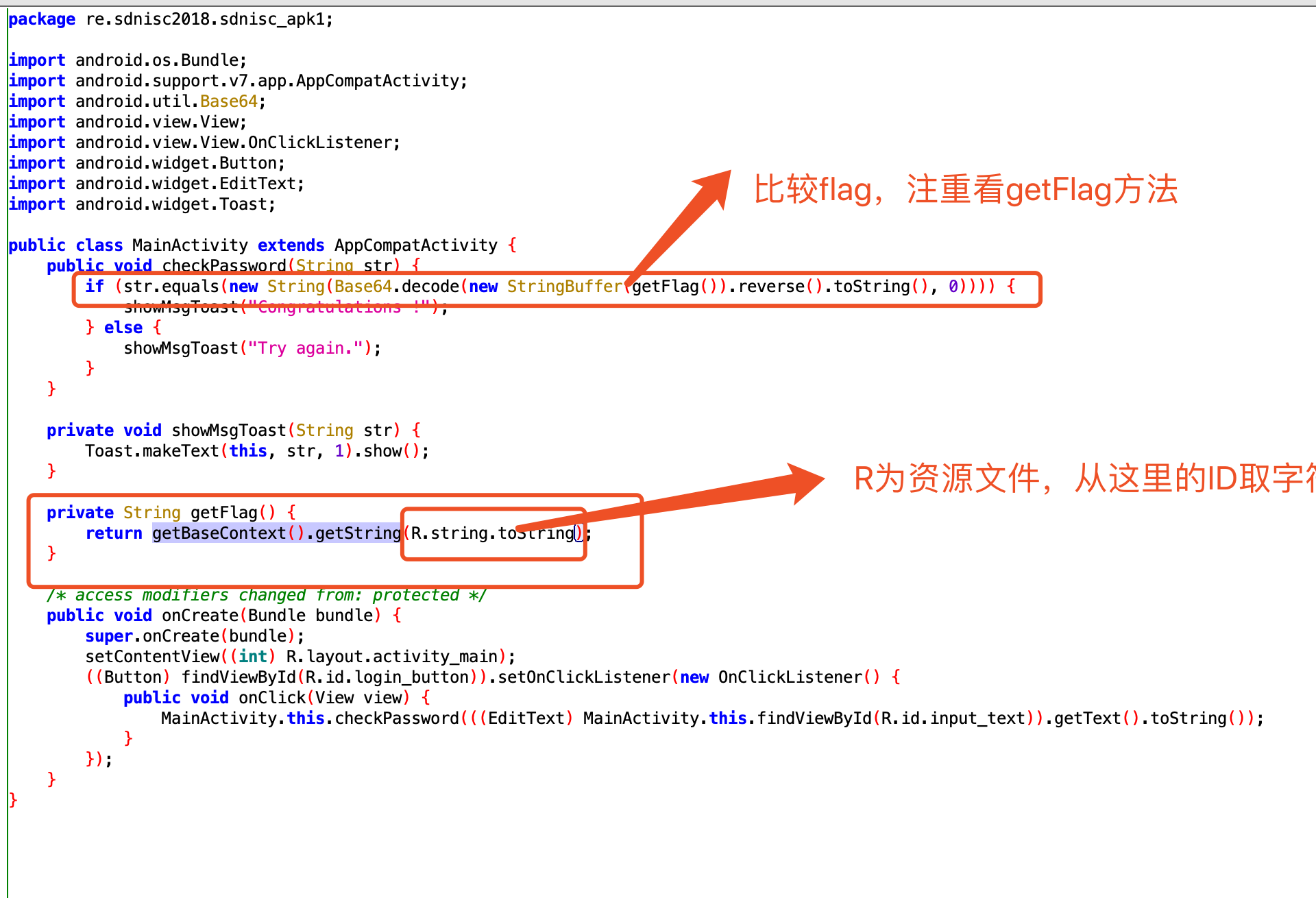
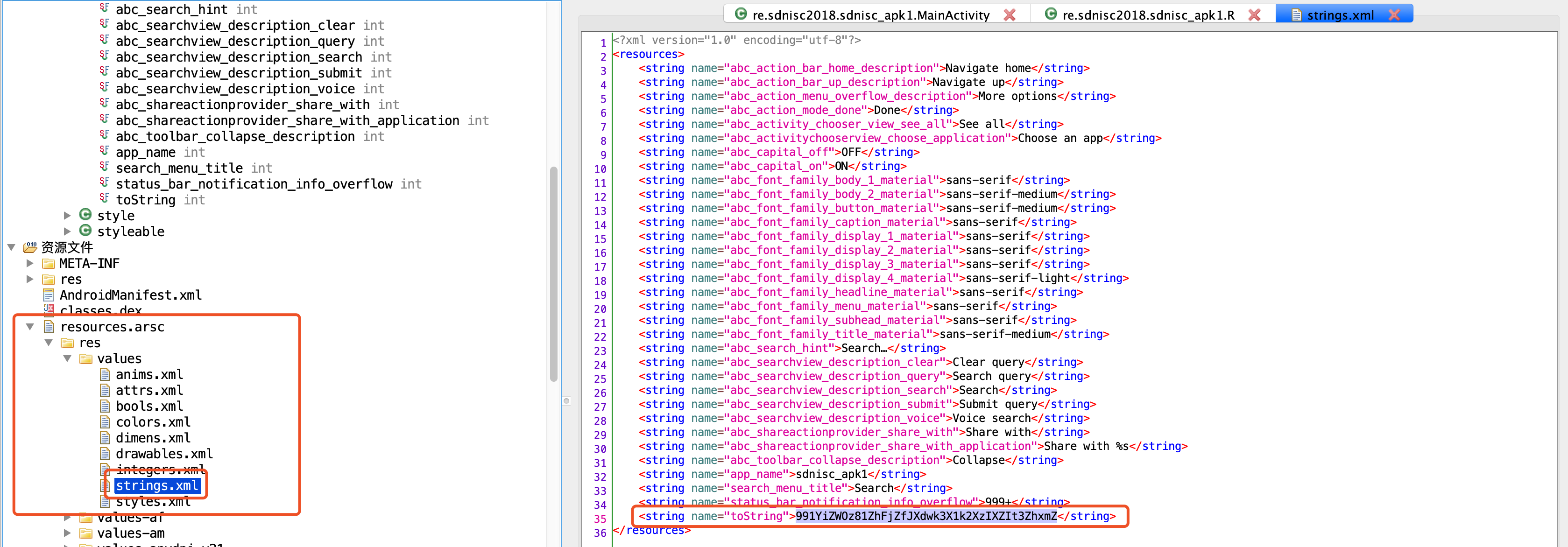
BugKu Android 题目详解

1.sign\_in

下载APK 拖进JADX（反编译工具）或者AndroidKIller或者JD-GUI 你喜欢就好



toString方法从String.xml读取字符串





倒过来读

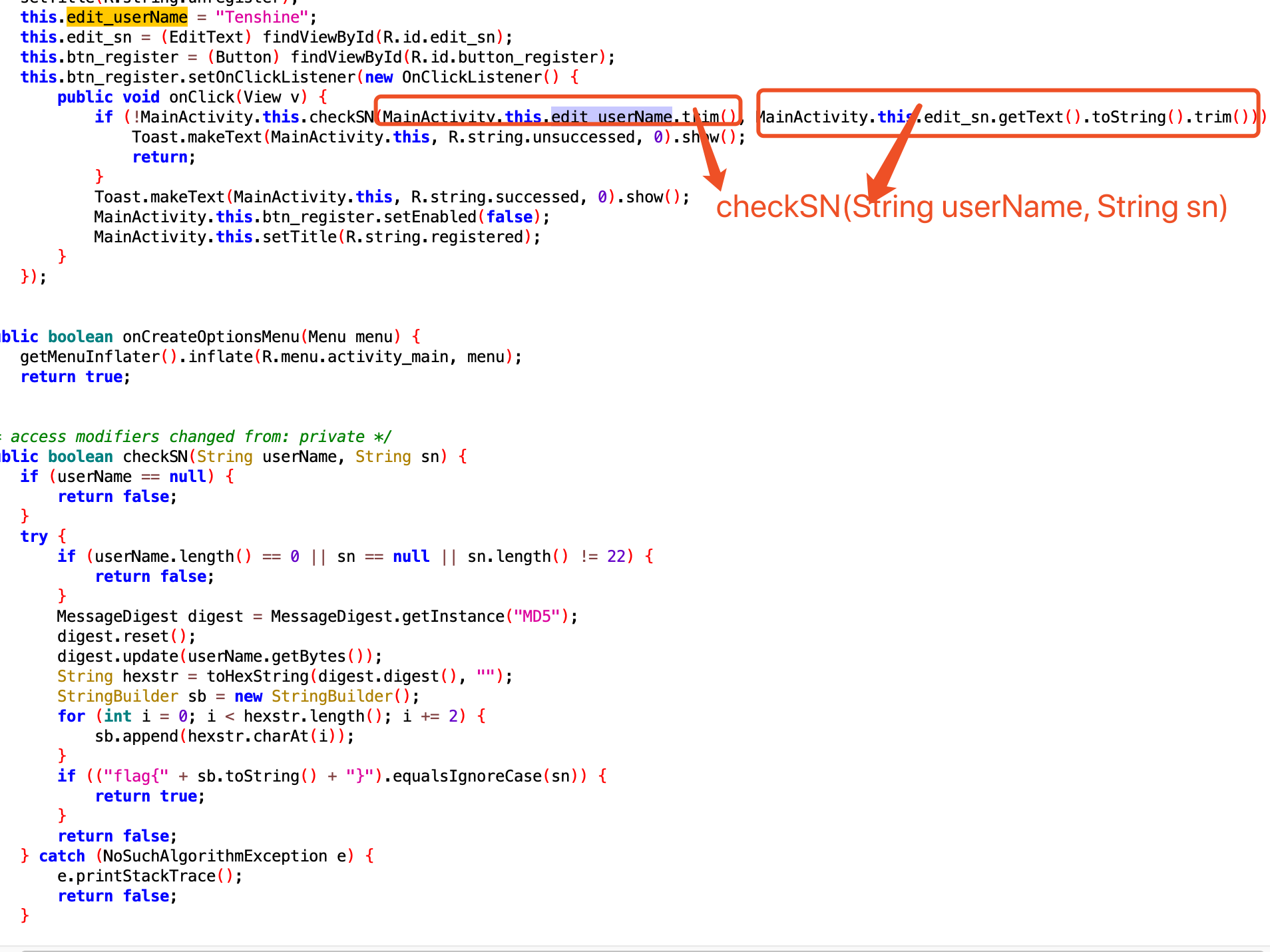
ZmxhZ3tIZXIzX2k1X3kwdXJfZjFhZ18zOWZiY199

Base64 decode

flag{Her3\_i5\_y0ur\_f1ag\_39fbc\_}

2.mobile1

按例 拖进去 强来再说



userName= "Tenshine"

sn=???





Flag{bc72f242a6af3857}

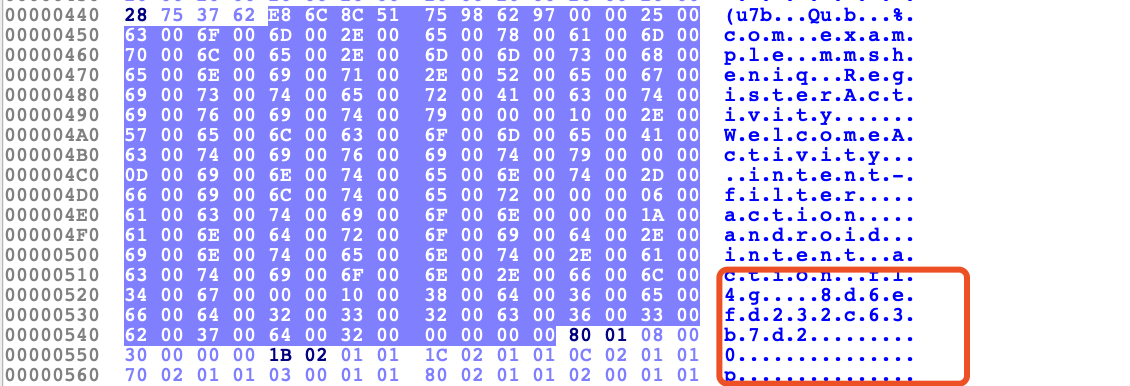
## 3. mobile2

## 有时候解压zip出来不是apk的时候，放进IDA里面看一下，这个就是解压出来不是，把后缀改apk就行

## 查看MainActivity发现一堆垃圾代码 啥用都没有，尝试查找String.xml 也无果 最后在AndroidManifest.xml里面发现了东西

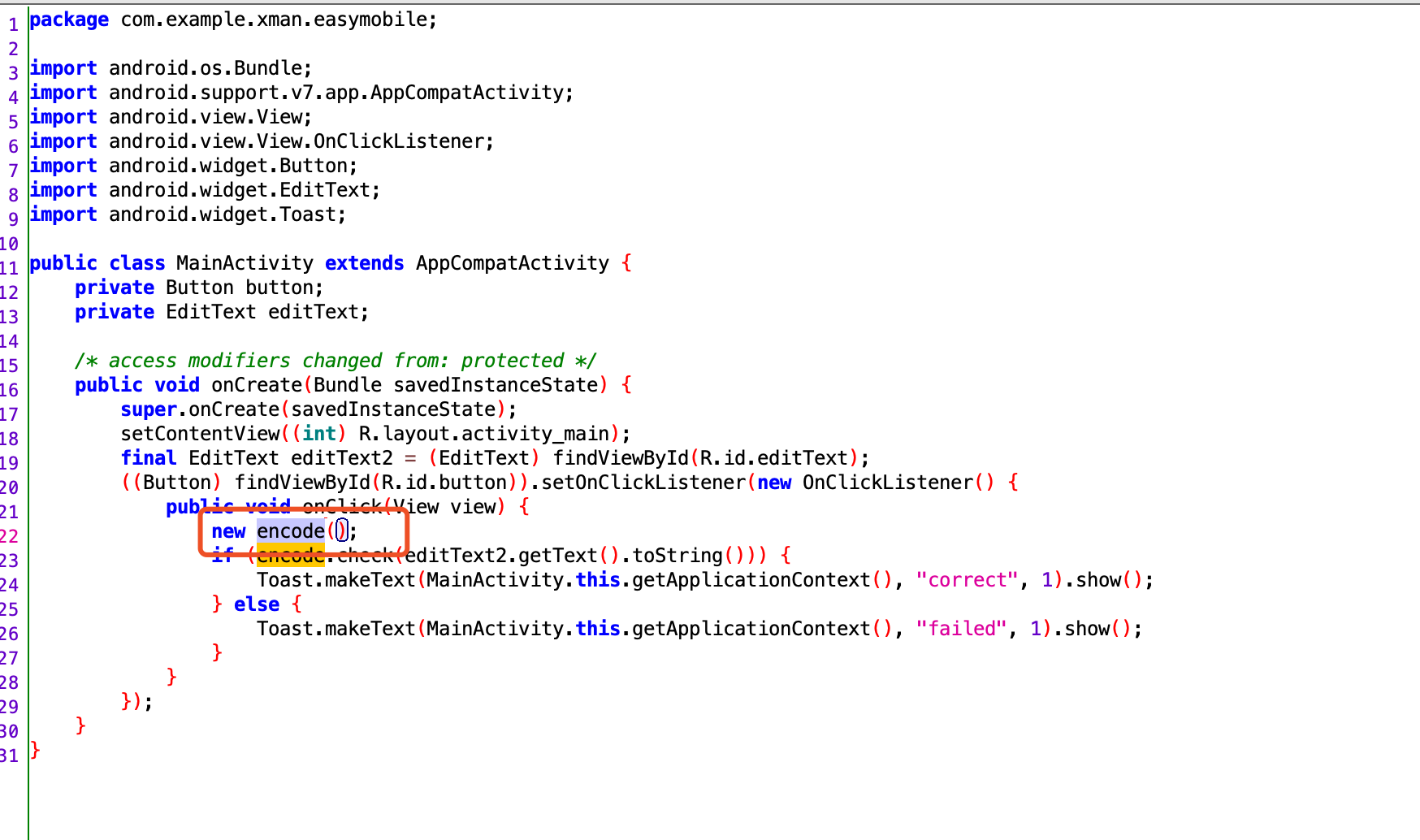
## 回敬一张图

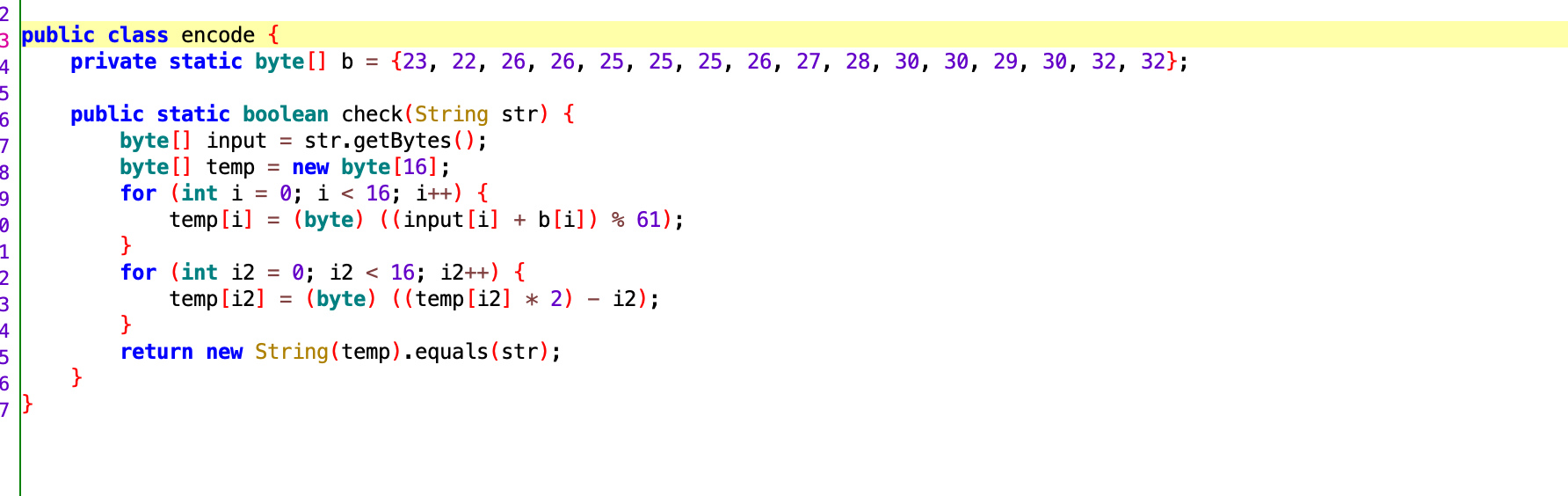
## 



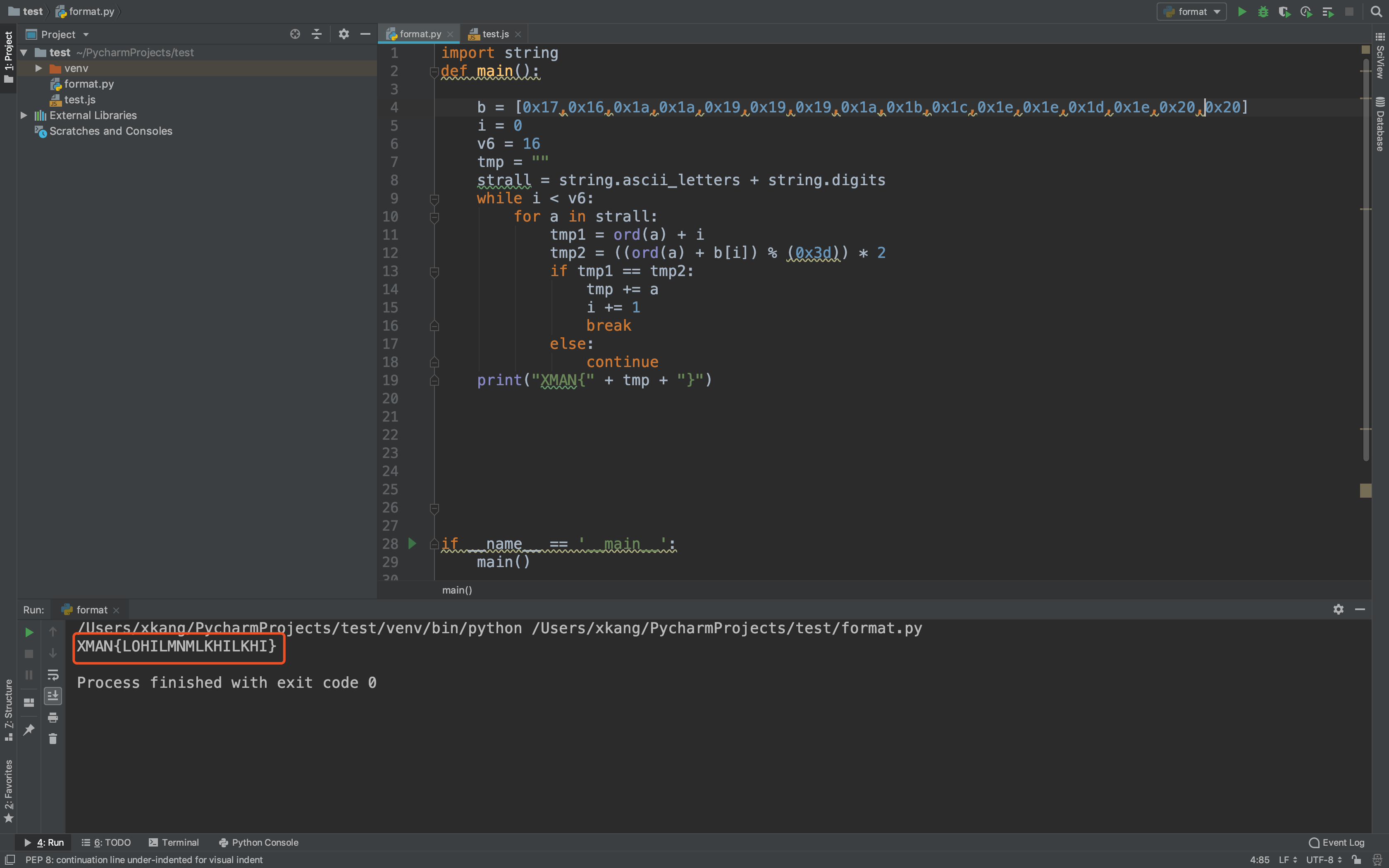
flag{8d6efd232c63b7d2}

4、First\_Mobile

加密函数去看看本体

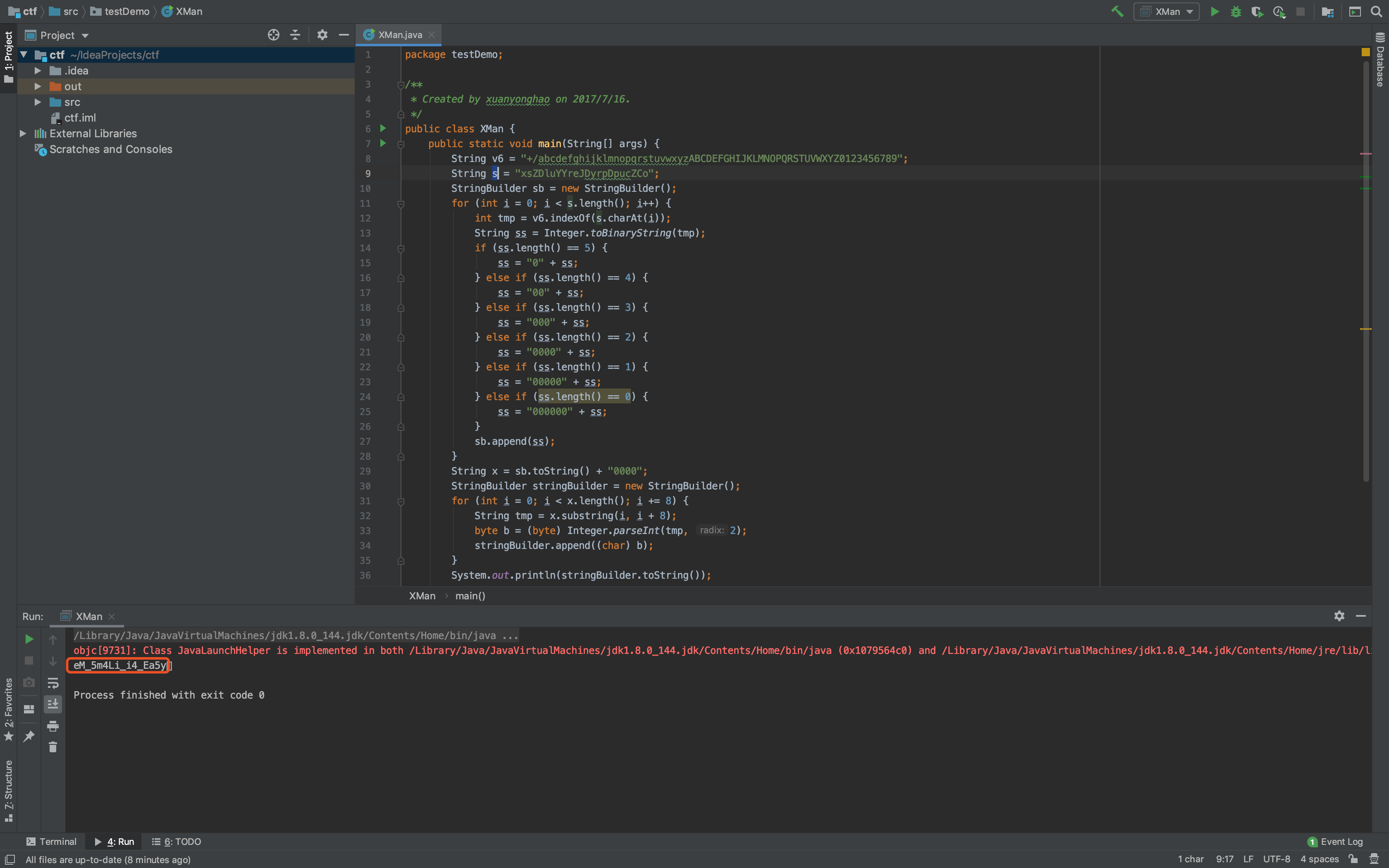


逆向推演



5、HelloSmali2

下载java源码 运行一下 Flag就得到了



提交格式为XMAN{eM\_5m4Li\_i4\_Ea5y}

本人用flag{ eM\_5m4Li\_i4\_Ea5y }提交了一次又一次 ☺



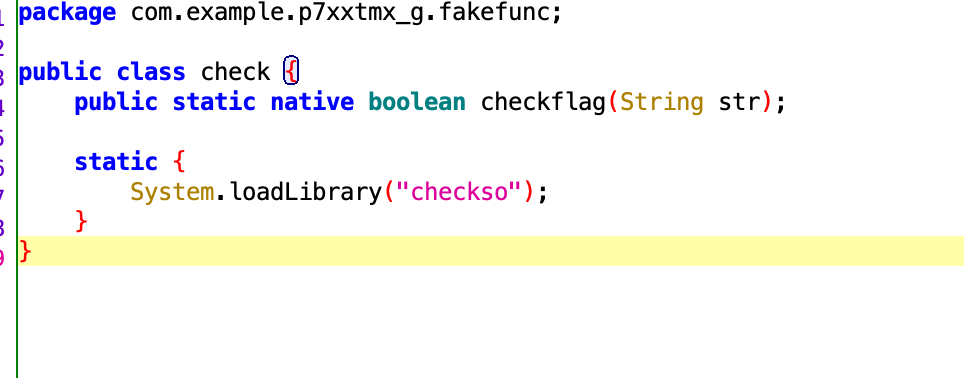
6、unity3d

过于复杂

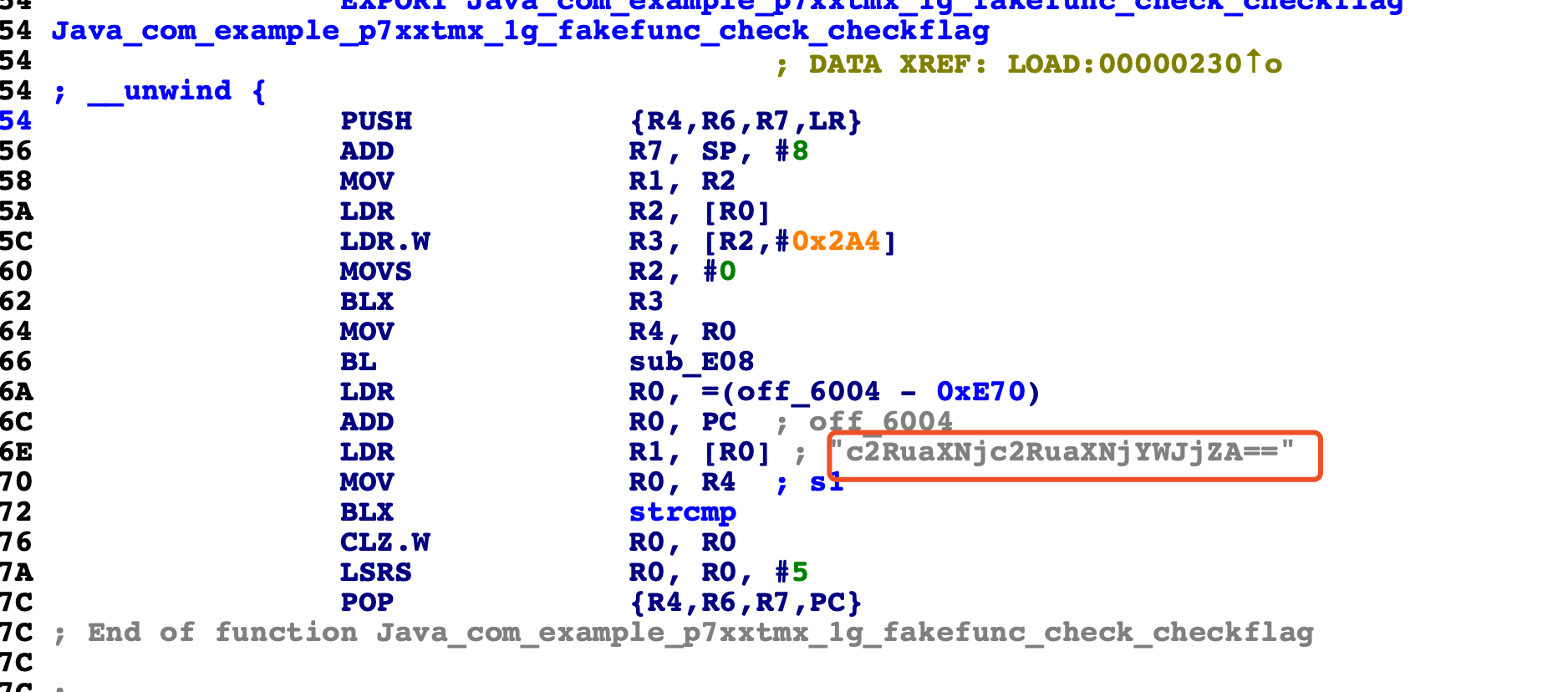
参考大佬链接：<https://veritas501.space/2018/03/28/%E4%B8%A4%E6%AC%A1CTF%E6%AF%94%E8%B5%9B%E6%80%BB%E7%BB%93/>

7、fake-func





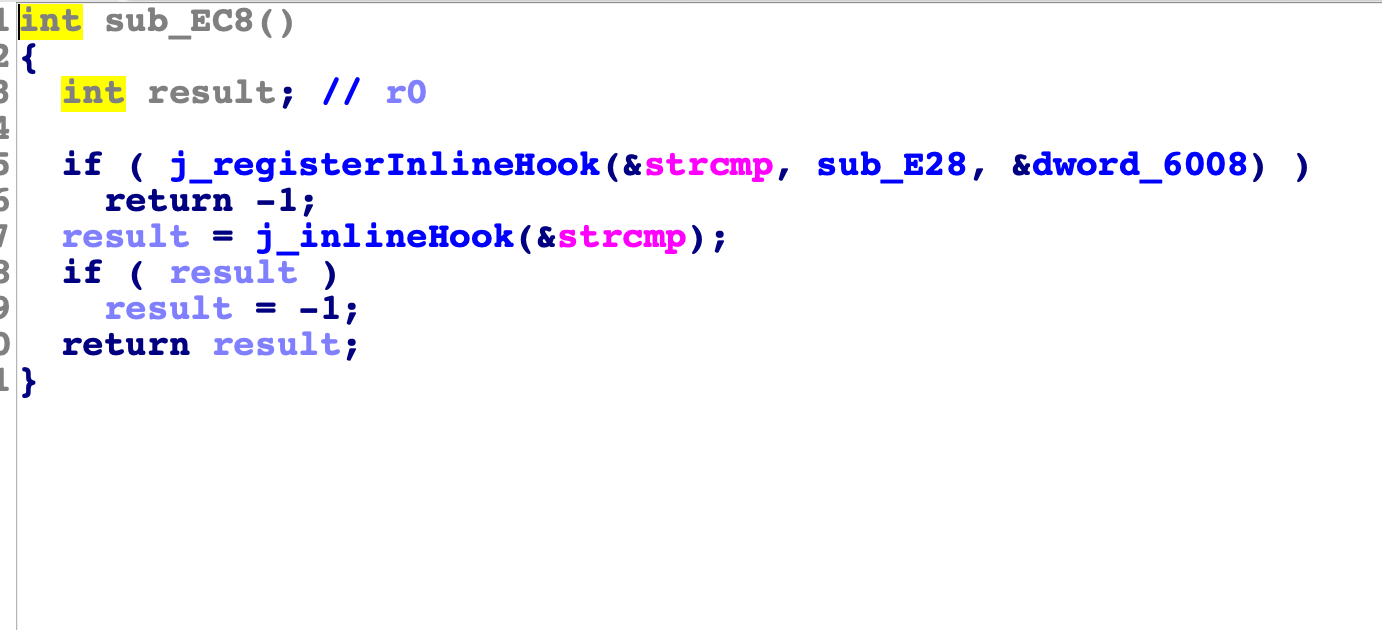
发现是加载so，直接去lib找so并载入IDA中



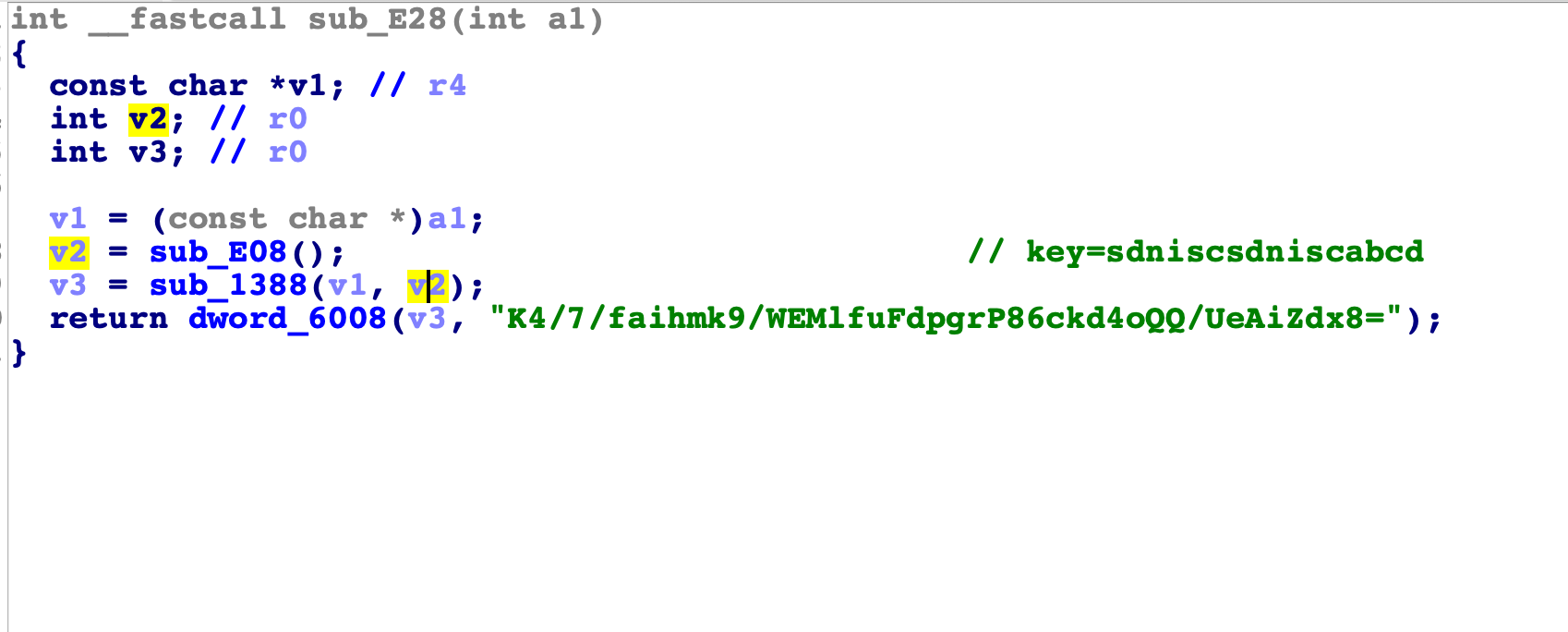
乍一看 喜滋滋 base64加密 解码 获得flag 实际上



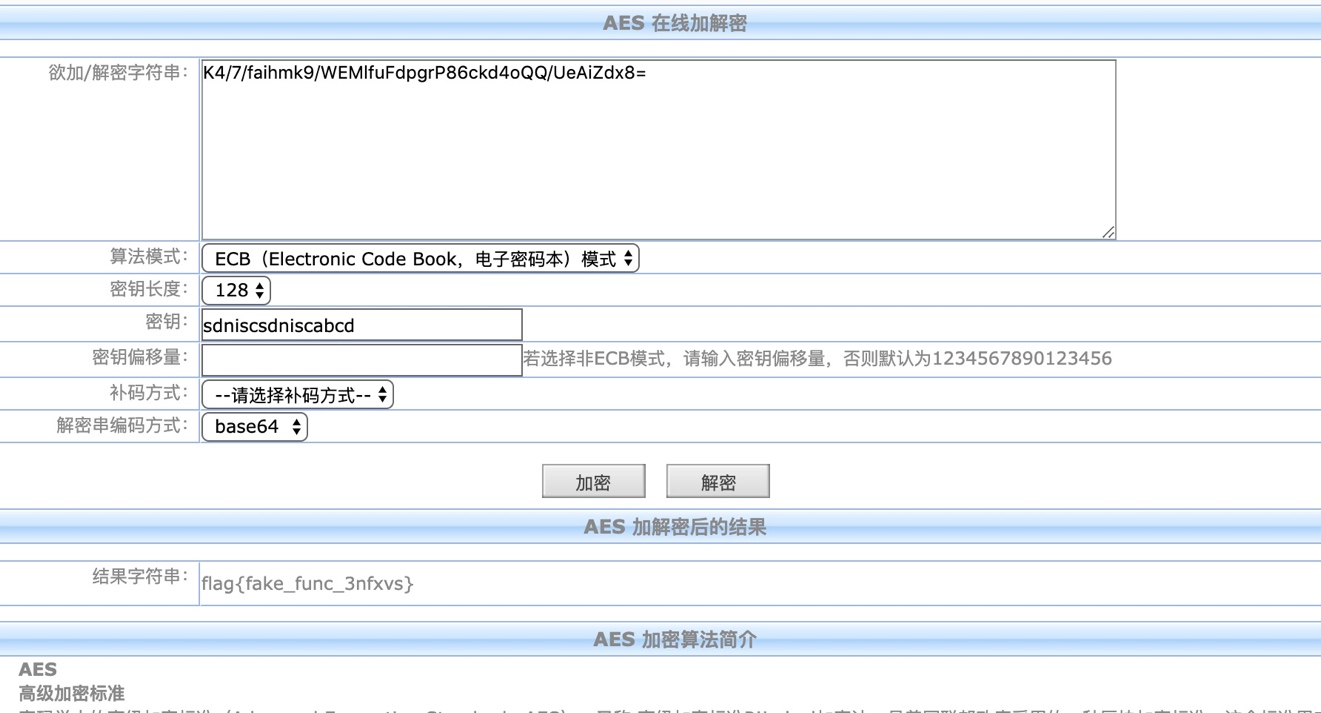
输入不对 在理清一下思绪



发现inlinehook hook了strcmp 事有蹊跷 看sub\_E28



发现是AES加密 key为刚刚base64加密字符串解密后的字符



喜滋滋 搞定

### 8、MagicImageViewer







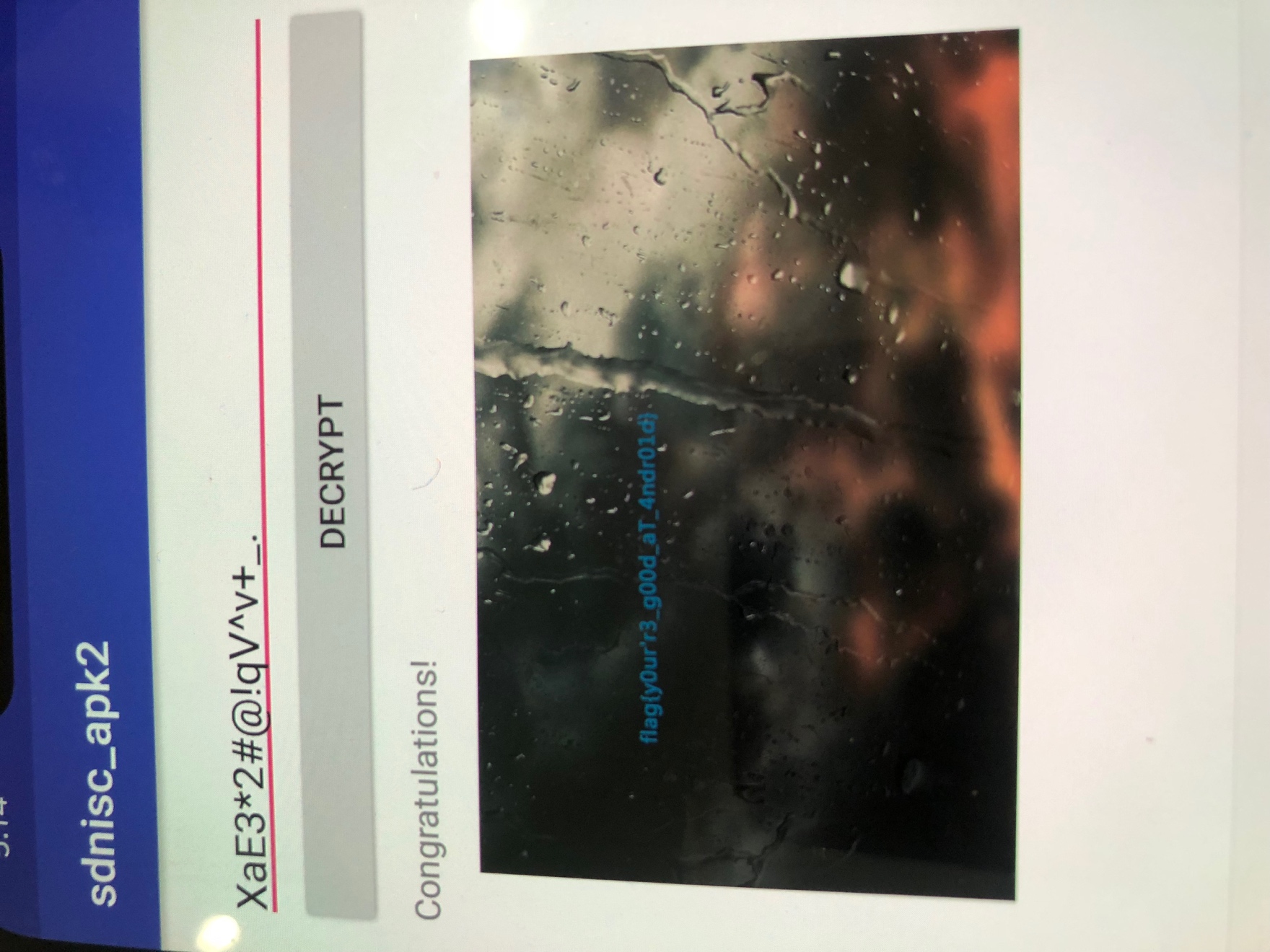




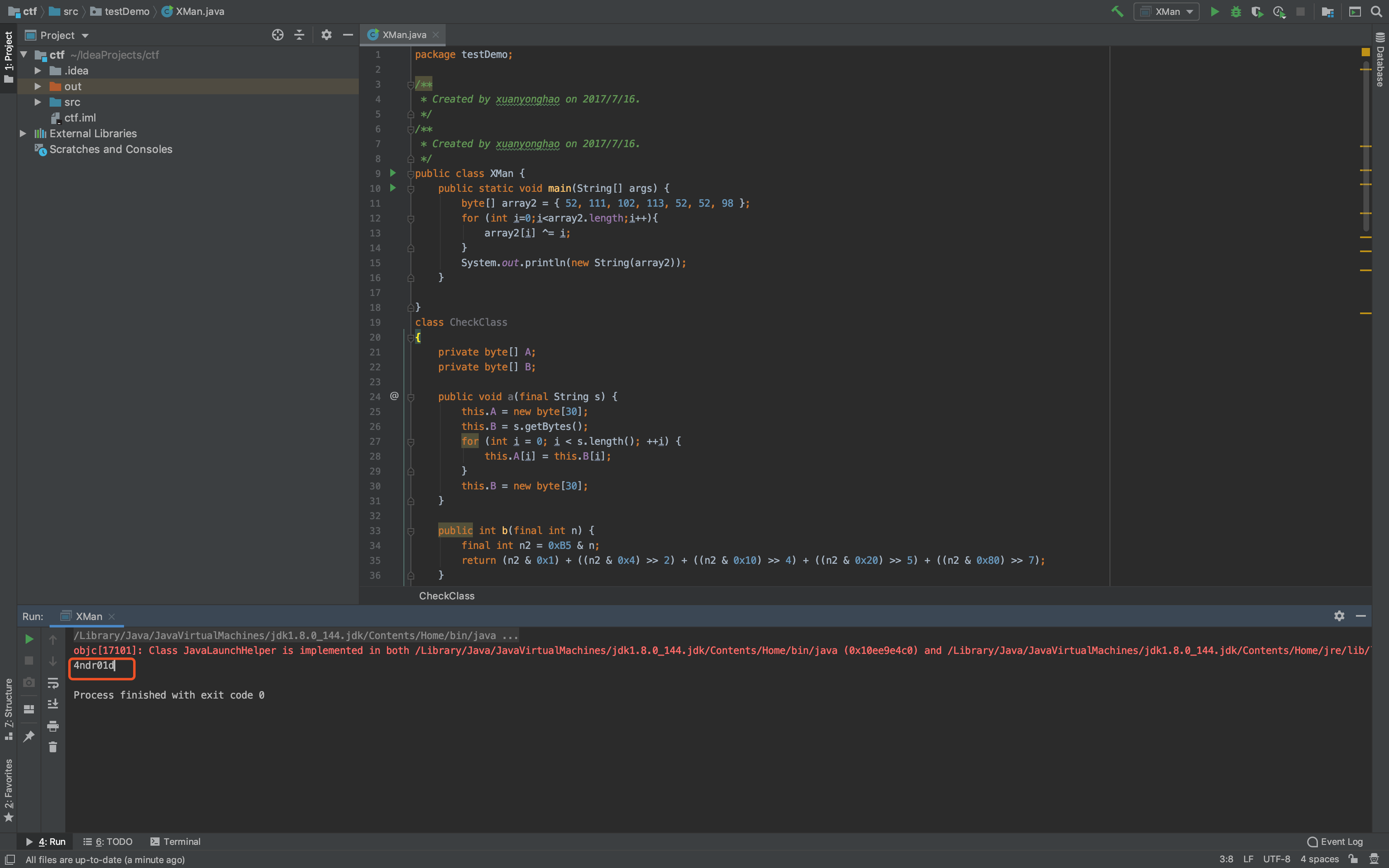
获得key

继而输入进去

获得flag



9、rev1（XMAN）



下载下来 运行一下 ok

150的题？？？？？？？？？

