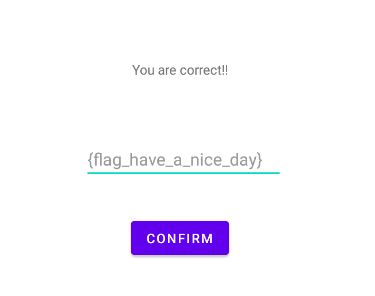
**Lab 3. Deobfuscate Android Apps**

* **Task A**

(1) Your Answer

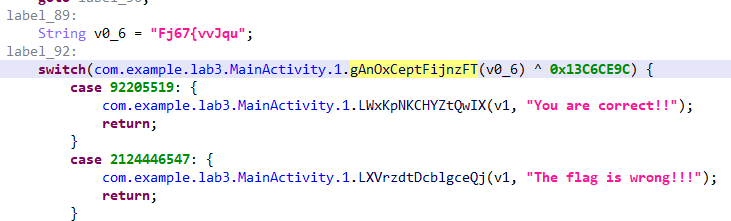
输入{flag\_have\_a\_nice\_day}可以得到正确的反馈，倘若中间有字符输错，则会反馈错误：

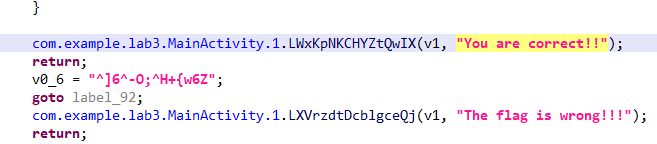


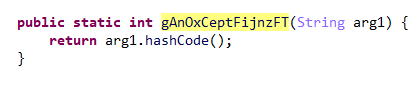
(2) Writeup

[Record how you solve this task here.]

首先在代码当中寻找输出”correct”，然后顺着找能够使得switch选择该正确输出的内容，如下图所示：

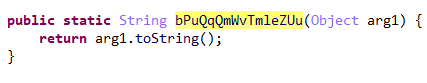


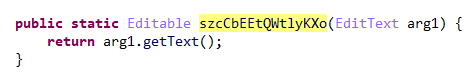


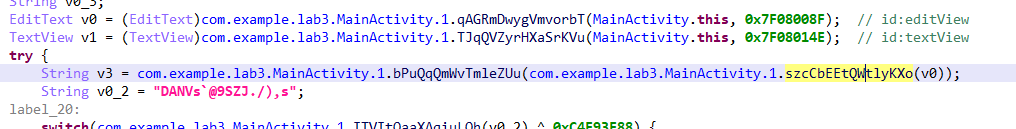


可以看到该函数运用了hashCode函数，hashCode方法返回该对象的哈希码值，这就要求我们再寻找v0\_6的相关内容，或者说查找程序如何跳转到这一部分label\_92或者label\_89。

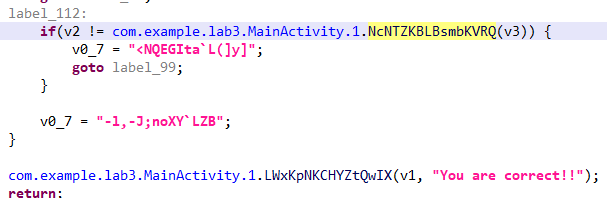
回过头来看输入的内容，可以根据代码看出来我们的输入内容将会被转化为文本文件、字符串，从v0传递到了v3, 接下来去寻找v3这个关键人物。

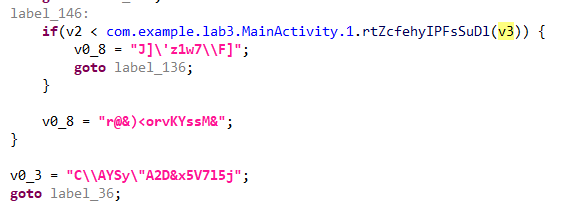


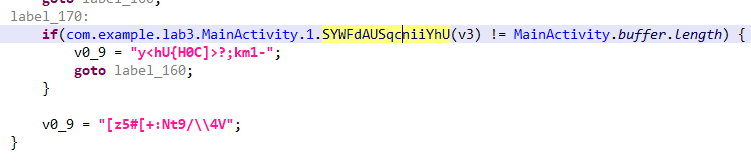


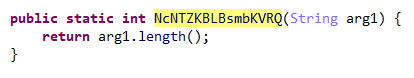


寻找到以v3为关键参数的几个函数，都是返回了v3字符串的长度：

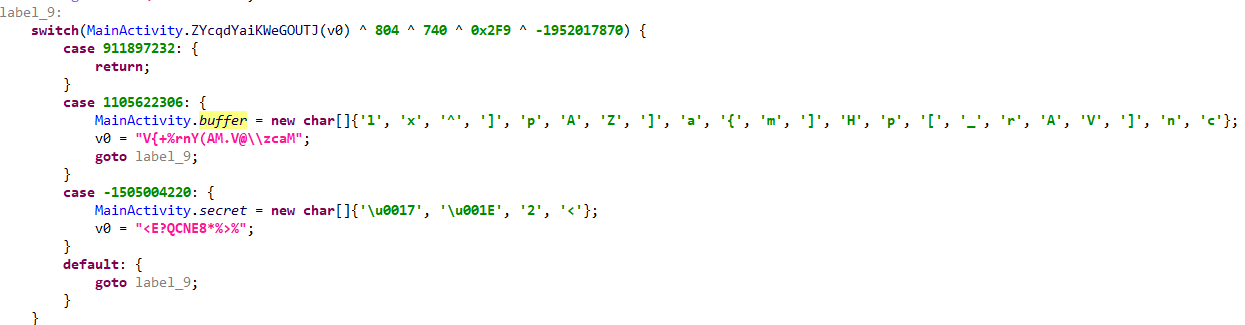






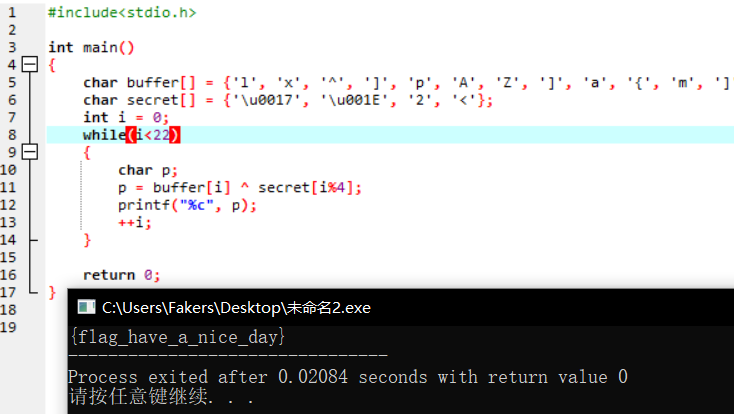


考虑到有个地方用到了buffer、secret，我们寻找关于他们的相关内容，以及和v2 v3相关的地方，下图中对于v2 v3的运用是为了得到v3字符串当中位于v2的字符char值。接下来就是继续寻找v2曾经走过的地方。





大胆猜想关键信息buffer和secret的用法是 ^ 运算，同时利用编写的C语言程序跑出运算结果如下：

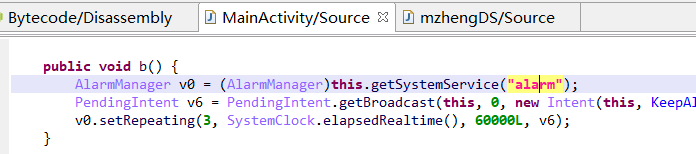


* **Task B**

(1) Your Answer

decrypted string1

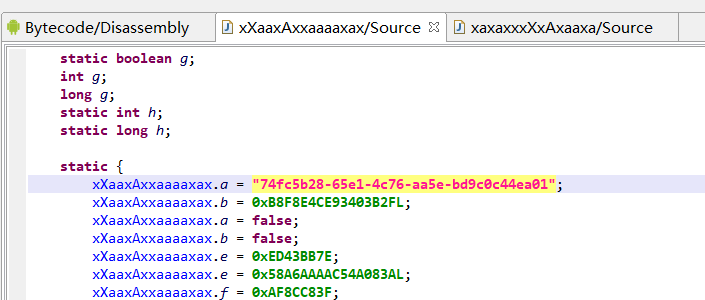
原来的字符串应该是alarm，主程序中位置和解密程序如下：

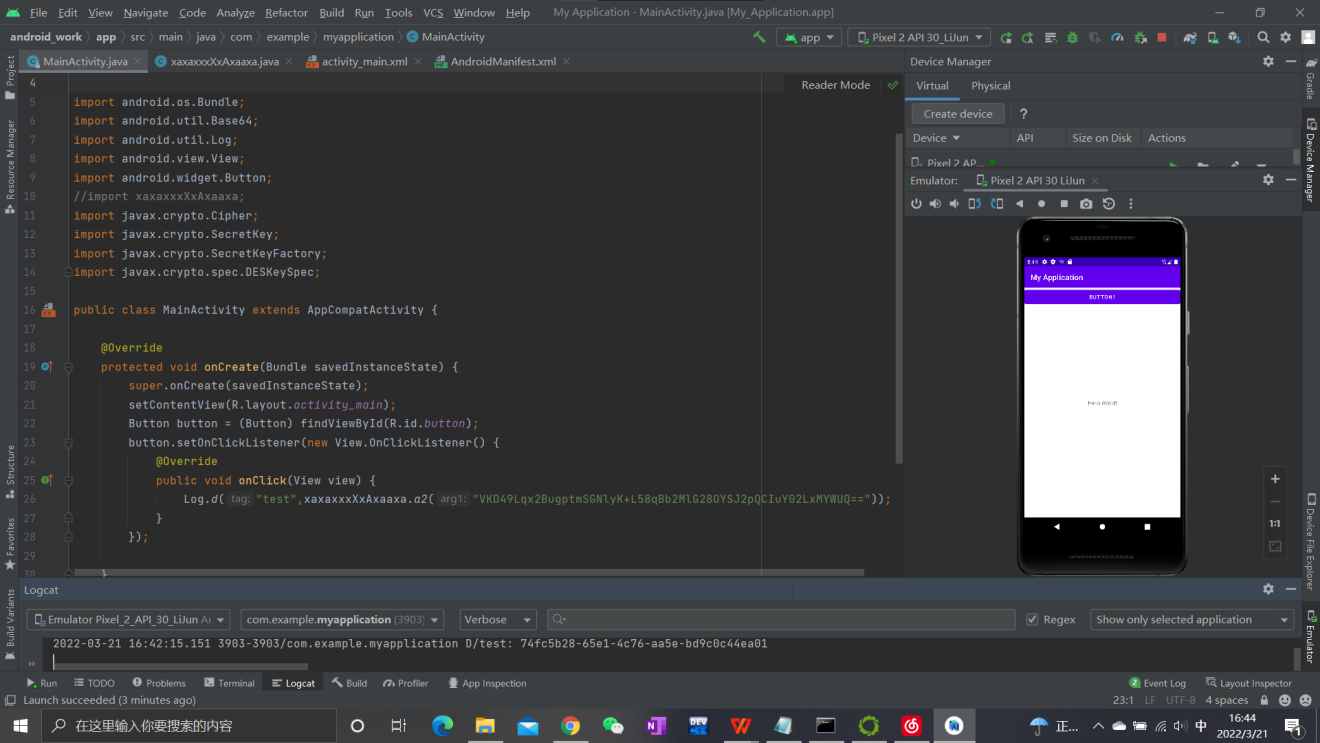




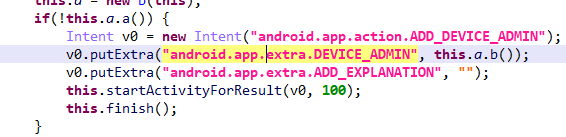
decrypted string2，程序名是4DBDE00A61FEF8AC53AF420F6ABC7A1A

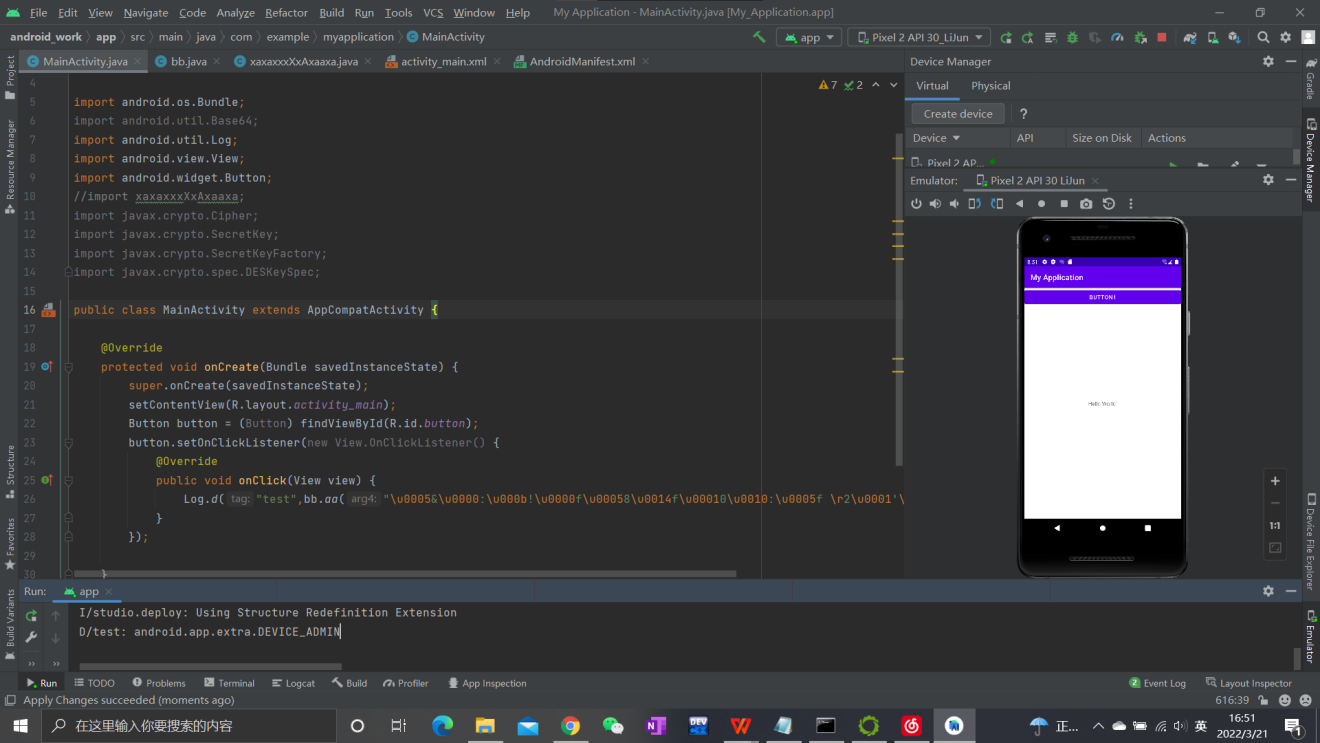
解密之后的字符串应该是74fc5b28-65e1-4c76-aa5e-bd9c0c44ea01，在原处寻找以及在android stdudio上运行得到结果如下：





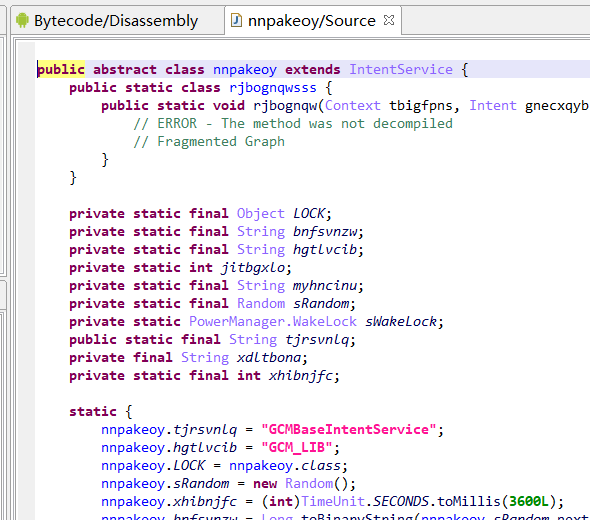
decrypted string3，程序名为6F9BFC9AE52562A2139CCDE94A2365F3

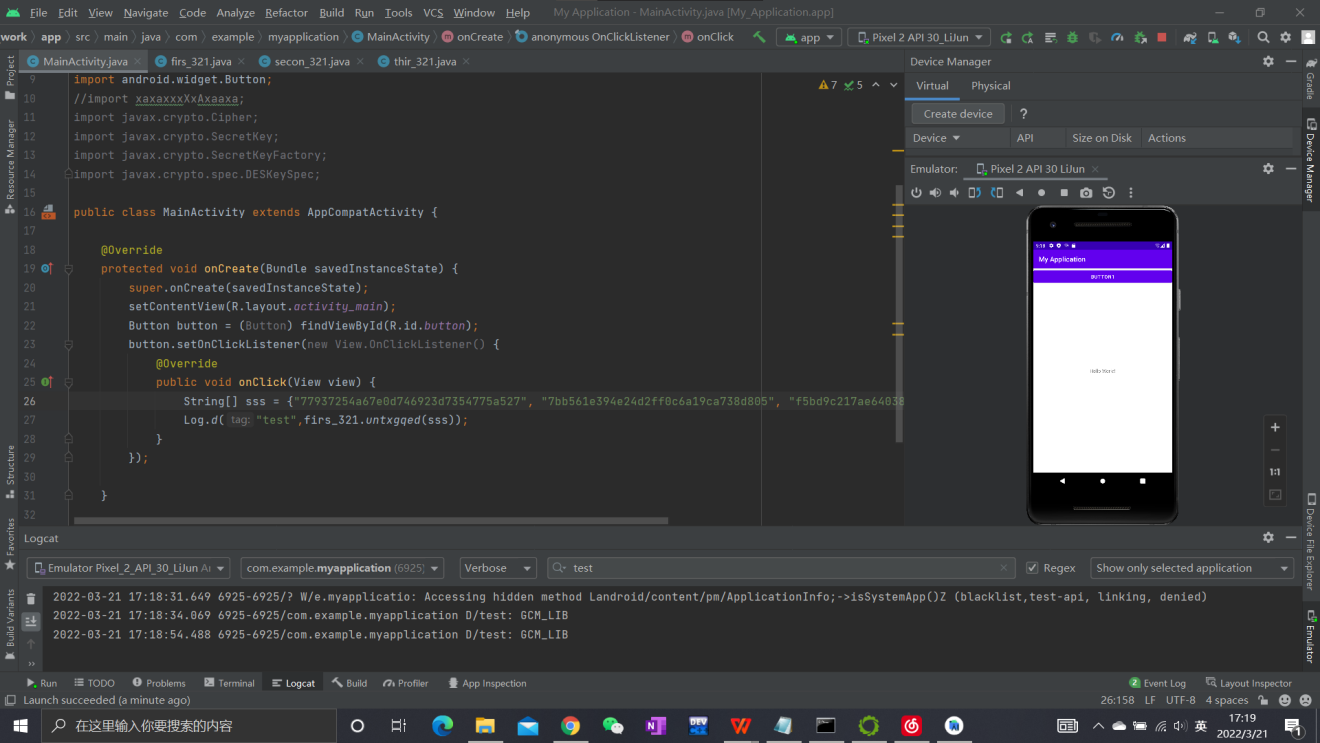
原来的字符串应该是android.app.extra.DEVICE\_ADMIN，



decrypted string4，程序名07F1393D405F0A1A6D951A19428B9DDB

揭秘之后是GCM\_LIB





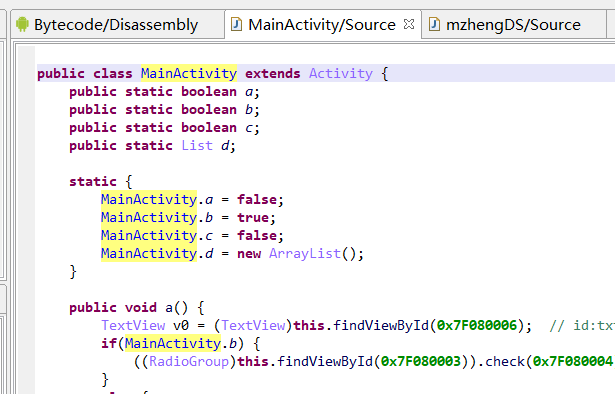
(2) Writeup

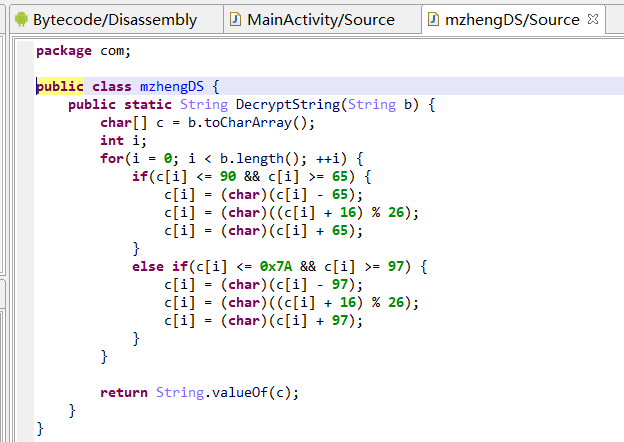
[Record how you solve this task here.]

**Task B实验内容的解决方法主要是根据题目条件找到相应的解密函数，编写安卓程序来解密**

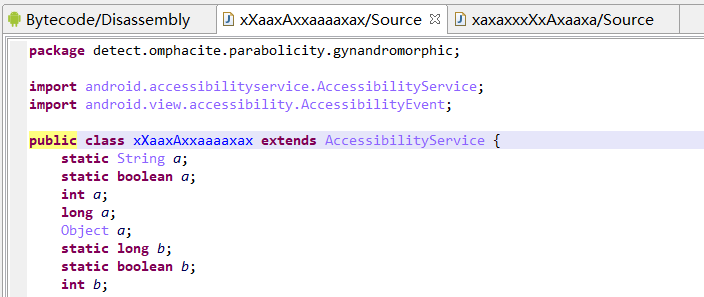
1.程序名2D858B227616E6FFAC53B2A4BA918B5F

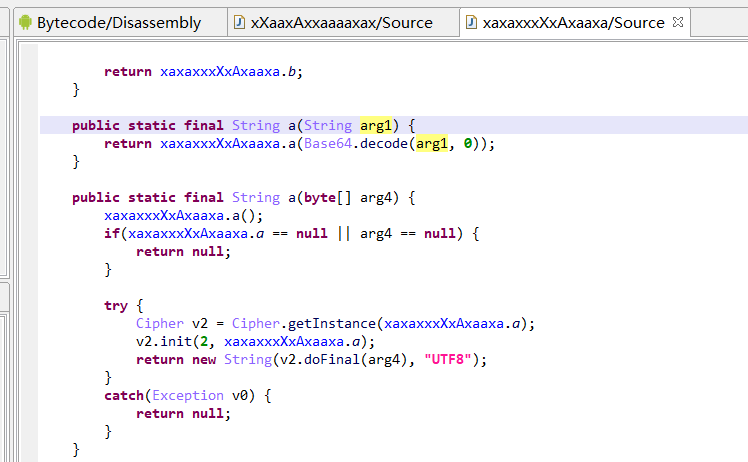
根据题目找到字符串所在的类，以及解密函数的位置和内容：



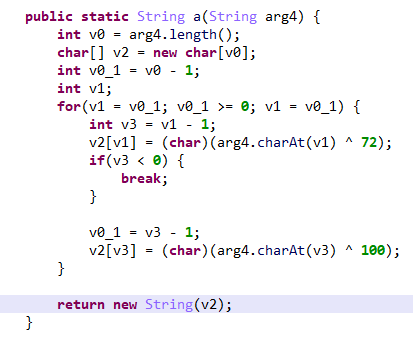


2.程序名是4DBDE00A61FEF8AC53AF420F6ABC7A1A，根据题意找到其所在的类，以及解密函数所在的地方，如下图所示：





3.程序名6F9BFC9AE52562A2139CCDE94A2365F3的程序，由题意可知目录下面的解密函数，然后在android studio中可以运行得到结果



4.程序名07F1393D405F0A1A6D951A19428B9DDB，

奔着解密函数去：

