1、

我选择的是生成256bit的素数，具体的素数生成时间取决于米勒罗宾算法的重复检测次数。

要精确测试的话最好设为26，但这样会花费相当长的时间计算。

这里兼顾精准和时间，设置为10，一般等待2~3min即可得到两个素数

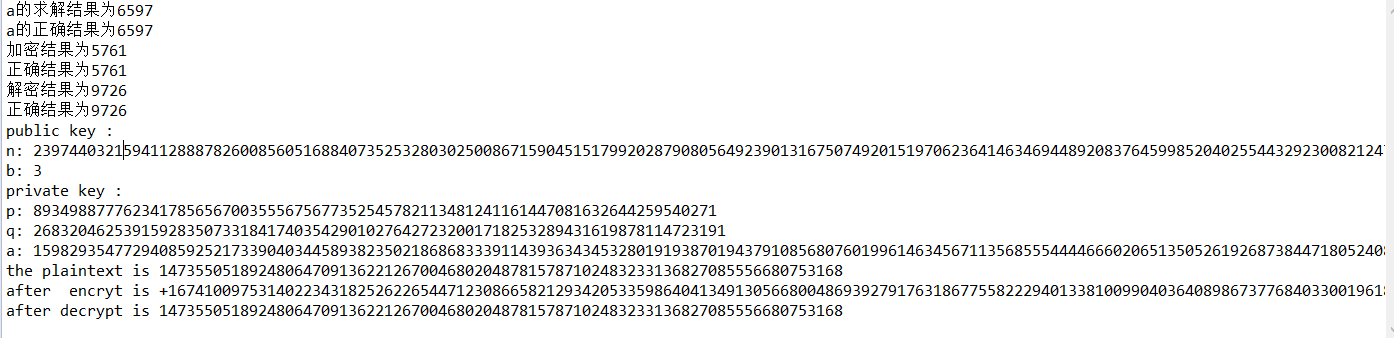
3、

得到密钥后对明文加密用时较短，这里未加记录。

4、

正确性也通过了老师给的数据。

程序截图1：

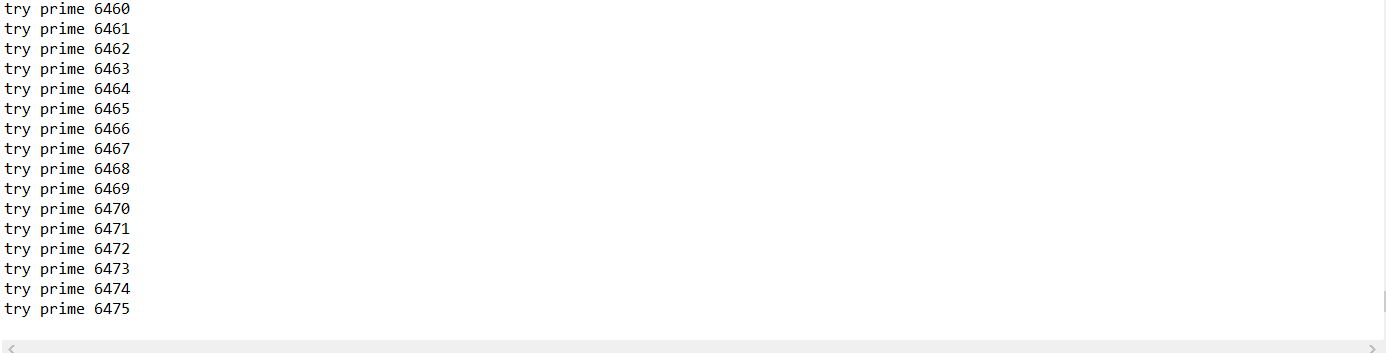


前面7行是利用老师给的数据对RSA设计的正确性进行的检测，可以看出与期望的结果完全一致。

下面输出公钥n和b,私钥p,q和a，并且对随机生成的明文进行了加密，并对加密后的结果进行了解密。

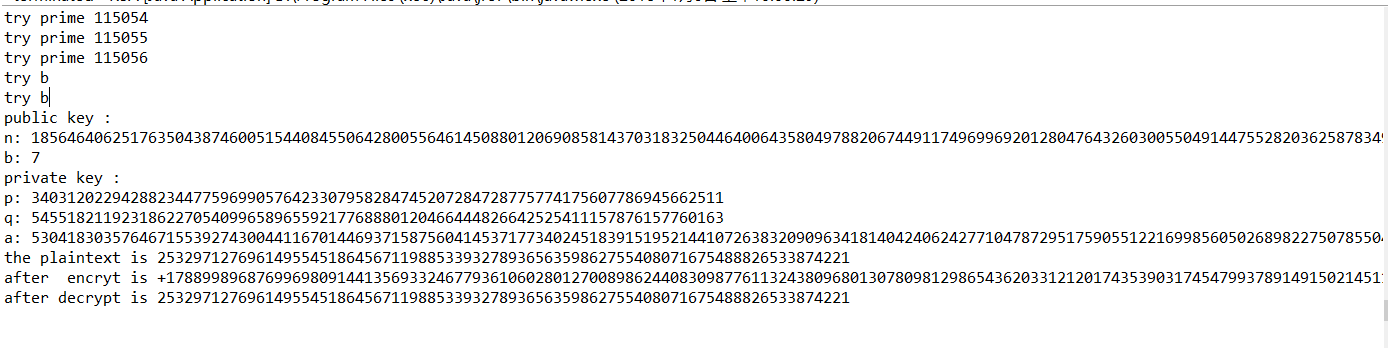
从最后3行可以看出，对加密后的结果解密与加密之前的明文完全相同，保证了解密是加密的逆运算，能够得到原来的明文。

程序截图2：



程序第169行有一个ignoreGetPrime的布尔变量，设置其为false，程序会自己去生成一对素数而不是调用之前得到的素数。生成素数可能会费时几分钟，不停输出的try prime 表示测试的次数

最终得到了需要的密钥部分，并且对随机生成了的明文进行了加密



Try prime和public key之间输出的try b表示程序正在试图寻找一个跟fn互素的b。