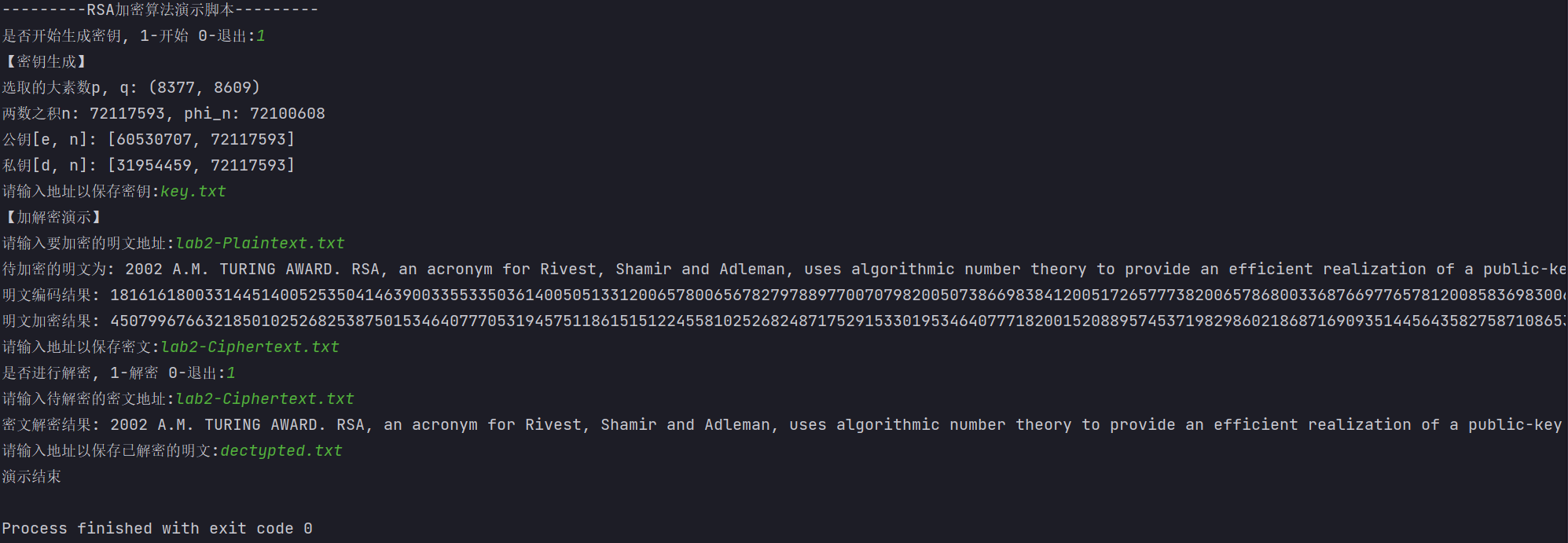
**实验2 RSA密码算法**

**姓名： \_王木一\_ 学号：\_200210231\_**

1. ****运行截图**
2. **实验过程中遇到的问题有哪些？你是怎么解决的。**
3. 分组编码

每次加密一个4位的十进制数字，但加密后的数字位数并不唯一，就会导致解密时不知如何分组，本次实验采用一种很简单的方法具体请见下方。

1. **请说明你的字符分组方式，以及关键的算法例如扩展欧几里德，素数检测，快速幂等。**
2. **分组与编码**

编码：采用ASCII编码，两个字符一组组成一个4位的十进制数字进行加密，若明文长度为奇数则在最后加上字符‘X’。由于部分字符的ASCII十进制码超过3位，故将每个字符编码值减去32，保证加密的数字小于10000。例如：‘z’对应ASCII码为122，‘y’对应ASCII码为121 ‘zy’对应的编码就为9089。若分组编码不足4位将在前面补0

加密：每组加密的结果位数不一定相等。由于选择的两个大素数p，q分别为14-bit的小于10000的数（范围8192~10000），两者之积n的位数为8位。故加密的结果

C的值最大为8位十进制数，但也有可能不足8位，故加密结果不为8位的在前面补0.

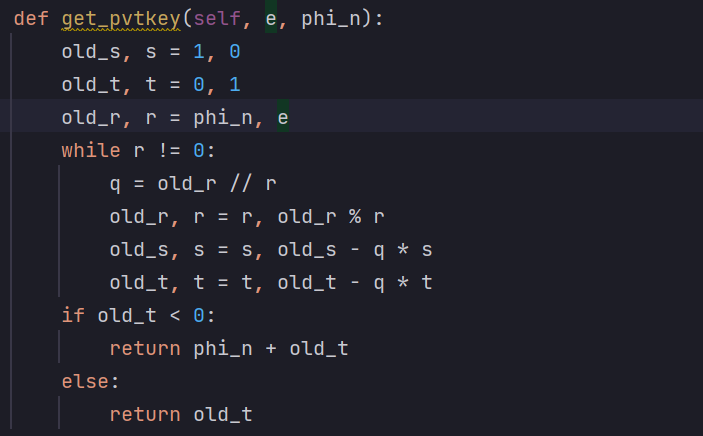
解密：由于每组加密的结果都已经规范化为8位，解密时按8位一组进行解密即可。

1. **加解密算法**

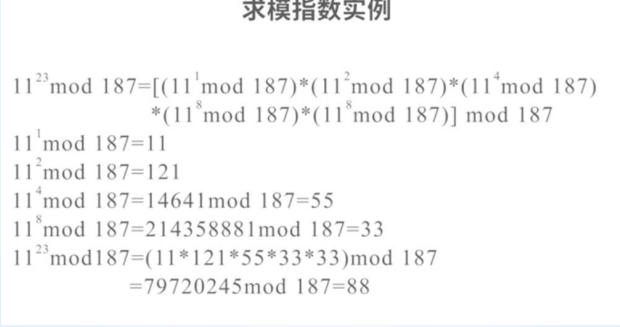
**扩展欧几里得算法（求d）：**根据裴蜀定理， 在RSA加密中，有. 即为e的乘法逆，即d。

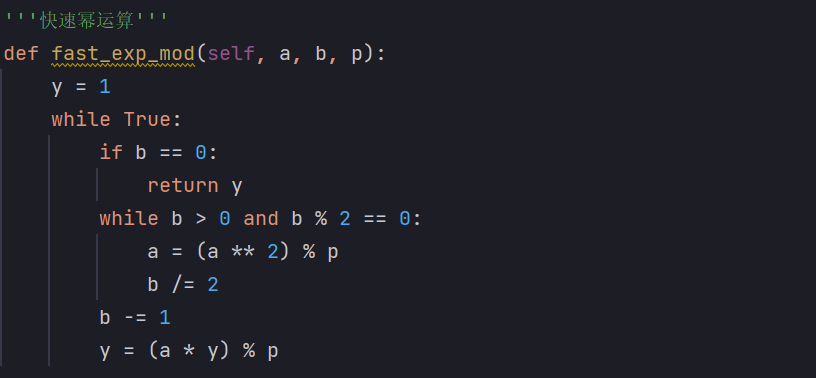
例如gcd(7,3) = 1 = 7 \* 1 + 3 \* (-2)， -2（5）就是3的模7乘法逆

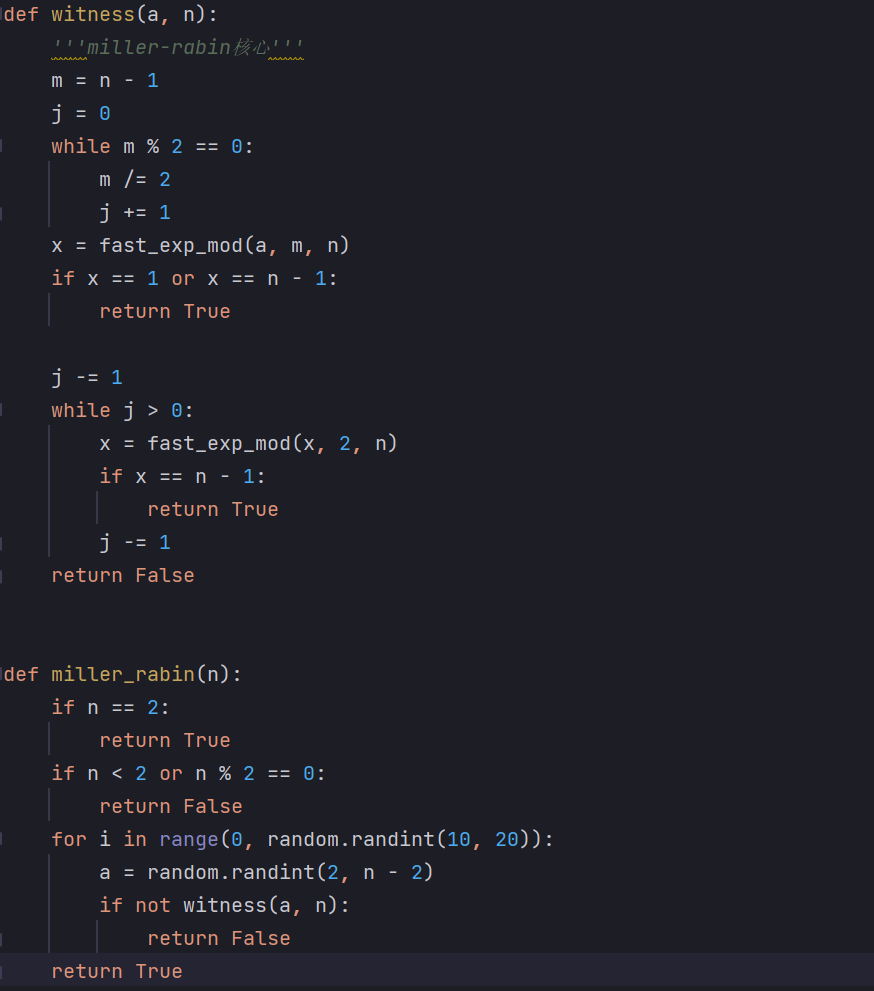
|  |  |  |  |
| --- | --- | --- | --- |
| 商q | 余数r | s | t |
|  | 7 | 1 | 0 |
|  | 3 | 0 | 1 |
| 7 / 3 = 2 | 7 – 3 \* 2 = 1 | 1 – 0 \* 2 = 1 | 0 – 1 \* 2 = -2 |
| 3 / 1 = 3 | 3 – 1 \* 3 = 0(结束 |  |  |

具体代码：

**快速幂算法：**根据模运算化简规则: 根据此思想进行快速幂运算。设m = a mod n, 那么a2 mod n = m2 mod n.

例如：

具体代码：

**素数产生：**首先选取随机数，再使用miller-rabin算法检测其是否为素数。算法检测至少10轮以降低素性检测错误概率。

注：因为老师未要求必须使用miller-rabin算法进行素数生成，故最开始使用的是pycrypto包来生成素数（Util.number.getPrime()）和判断素数（Util.number.isPrime()）。后考虑到提交后复现的测试环境，还是实现了miller-rabin算法，不再额外使用需要单独配置的工具包。

本次实验环境：python==3.7