**密码学实验五实验报告**

姓名：孟衍璋 学号：16337183

1. **实验目的**

实现RSA算法。

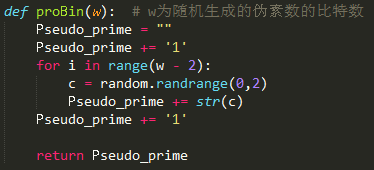
1. **实验内容**

将自己学号作为私钥，并生成对应的公钥。加密：SUN YAT-SEN UNIVERSITY（包含空格），给出密文，并给出解密函数，进行验证。

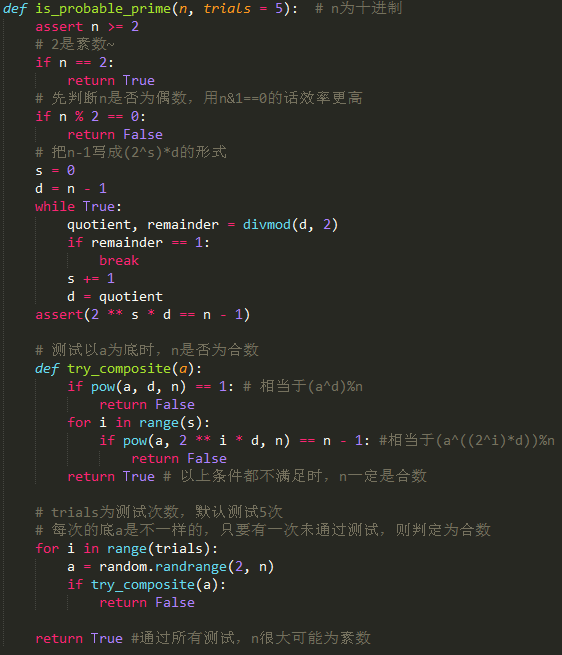
1. **实验及算法原理**

首先，需要考虑的第一件事就是如何生成512位的大素数。基本思路为，先随机生成一个512位的伪素数，确保其为奇数，再利用Miller-Rabin素性检测判断该数是否为素数，如果不是，在附近不断+2应该就能很快地找到一个素数。

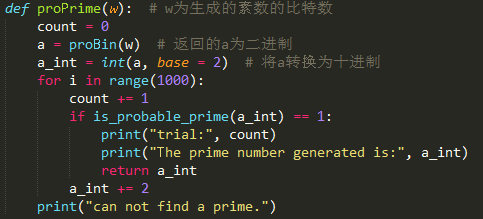
其中生成伪素数的代码为：



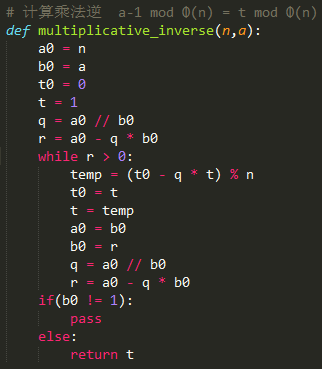
用Miller-Rabin素性检测来判断生成的伪素数是否为素数：



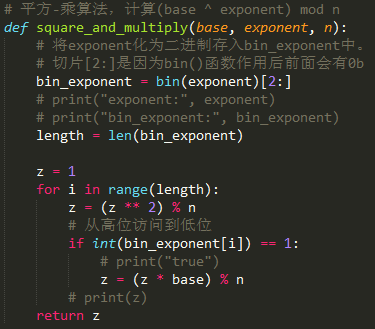
最后得到生成的素数：



得到生成的512位的大素数p与q之后，接下来要做的事就是生成公钥与私钥。其中私钥就是p、q与a，且选取自己的学号作为a的值。再根据私钥a生成公钥b，这里需要计算a\*b≡1(mod Φ(n))，即a-1 mod Φ(n)，根据书上的乘法逆的算法，实现如下：

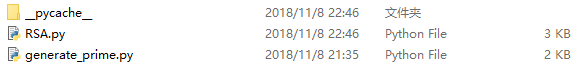


生成私钥与公钥之后，便可以开始加密与解密。加密与解密都是模指数运算，即xc mod n。如果c非常大，效率就会十分低下，而使用平方-乘算法，可以把计算xc mod n所需模乘的次数降低为最多2L次，其中L是c的二进制表示的比特数。于是xc mod n可以在时间O（Lk2）内算出。其中平方-乘算法实现如下：

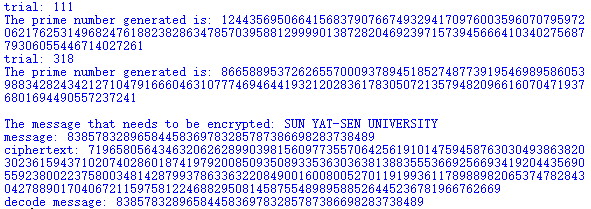


实现了平方-乘算法之后，便可以完成加密与解密操作。

1. **实验代码**



1. **运行截图**



1. **结果分析**

在结果中，消息经过加密之后得到密文，密文解密之后得到的消息与之前的相匹配，证明加密与解密的过程均正确。

1. **总结**

这次实验是实现RSA算法，实现的过程还算比较顺利，没有遇到什么大问题，而且经过多次阅读书上的相关内容，也能很清楚地了解RSA算法的实现过程。

在写代码的过程中，还是有遇到一些小问题，比如在实现平方-乘算法的时候，最开始将指数转化为二进制的时候，考虑到要从高位访问到低位，便将其倒叙输出，所以出现了问题。仔细思考之后，原来本身将指数转化为二进制的时候，得到的是一个字符串，按字符串下标从小到大的顺序刚好是二进制数从高位到低位的顺序。

这次的实验难度不算太大，最主要是搞清楚RSA算法实现的过程，实现过程中用到的最主要的两个算法便是平方-乘算法与计算乘法逆的算法，只要搞清楚了这两部分，其他的问题都不是太大。总而言之，这次实验还是有很大的收获。