**西安电子科技大学网信院**

**信息安全基础与密码学**

**综合实验**

**实 验 报 告（一）**

**Fermat素性检测算法**

**班级：2018039**

**姓名：叶振雷**

**学号：20009101064**

**日期：2022.10.30**

一、实验目的（包括实验环境、实现目标等等）

实验环境：windows11+python3.9+vscode2020

实现目标：

1.通过搭建环境并且实现算法，熟悉运用math、random库的基本函数操作；  
2.通过实现fermat[素数](https://so.csdn.net/so/search?q=%E7%B4%A0%E6%95%B0&spm=1001.2101.3001.7020" \t "_blank)检测算法，加深对于fermat小定理的理解与运用；  
3.体会密码学与数论的紧密联系，将理论化为实践。

二、方案设计

（包括背景、原理、必要的公式、图表、算法步骤等等）

Fermat素性检验素性检验：设 n 是一个大的奇数，

检验n是否为素数的测试。利用命题的检验算法：

输入n：

随机选取b，0<b<n，利用广义欧几里得除法计算b和n的最大

公因数d＝(b, n)。

(i) 如果d >1，则n不是素数。

(ii) 如果d＝1，则计算r＝b^(n－1)(mod n) 。

(a) 如果r≠1，则n不是素数；

(b) 如果r＝1，输出n。

如果经过多轮Fermat素性检验都通过，那么这个数为素数的概率就很接近1了。

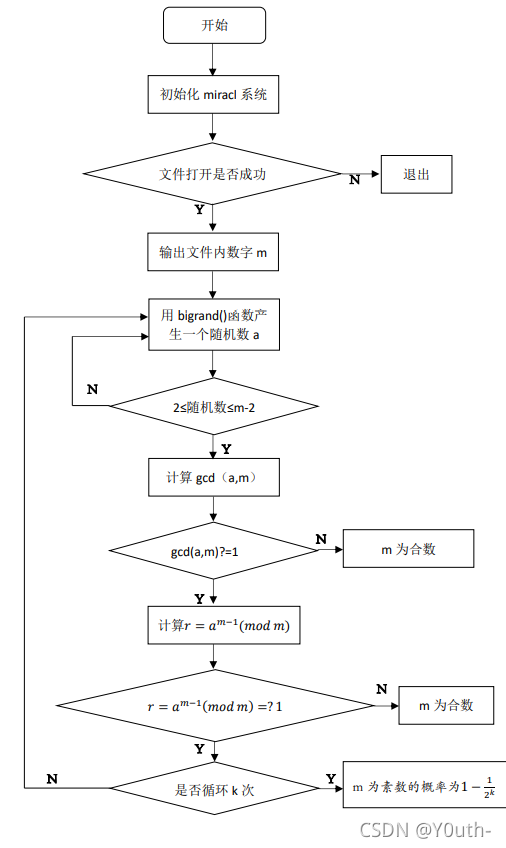
比较有意思的是：

设n为合数，如果对所有的正整数b， (b, n)＝1，都有同余式b^(n-1）≡1 (mod n) 成立，称n为Carmichael数。存在无穷多个Carmichael数。

这说明，就算所有小于n的正整数Fermat素性检验都通过，也无法完全保证这是一个素数，Fermat素性检验仅是保证很大概率为素数的一种方式（在实际应用中出错的概率及其小）

三、方案实现

（包括算法流程图、主要函数的介绍、算法实现的主要代码等等）



引入了math库和random库：

Math：使用pow函数以及math.gcd()求最小公约数

Random：主要是使用random.randint从2-（n-1）中取一个随机数a进行判断素性

主要函数：

def su\_test(case=0,k=0):

print(f"Get m = {case}")

print(f"m 的长度为:", len(str(case)))

**for** t in range(k):

a = randint(2, case-2) # 2 <= a <= (m-2)

print(f"Choose a{t} = {a}")

print()

g = gcd(a, case)#先求最大公因数

**if** g != 1:

print("m 为合数")

print("-----------------------------------------")#分隔符

**return**

r = pow(a, (case-1), case) # 快速幂取模

**if** r == 1:

**continue**

**else**:

print("m 为合数")

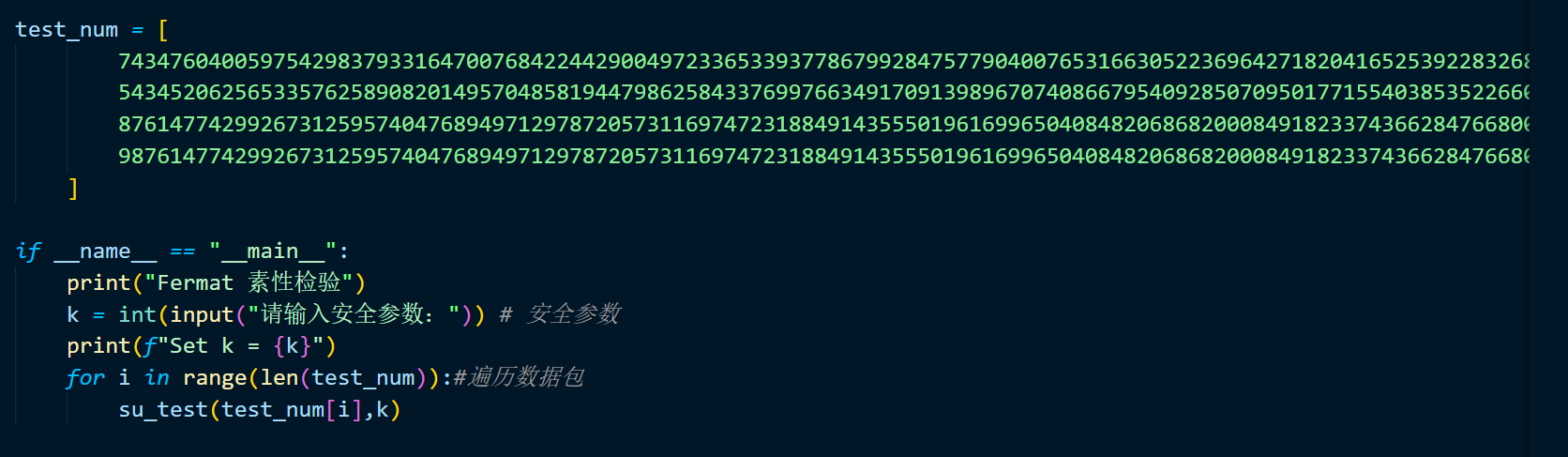
print("-----------------------------------------")#分隔符

**return**

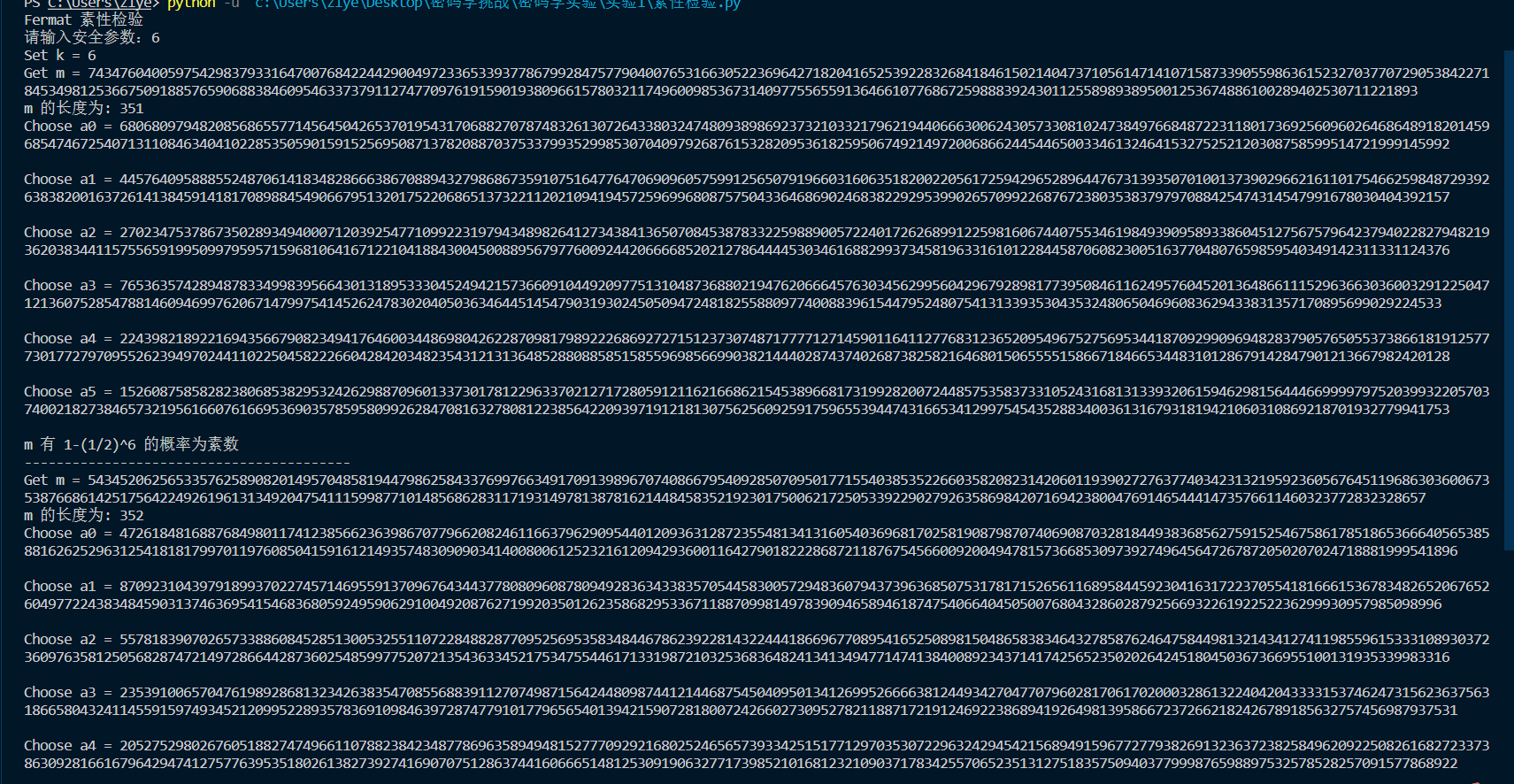
print(f"m 有 1-(1/2)^{k} 的概率为素数")

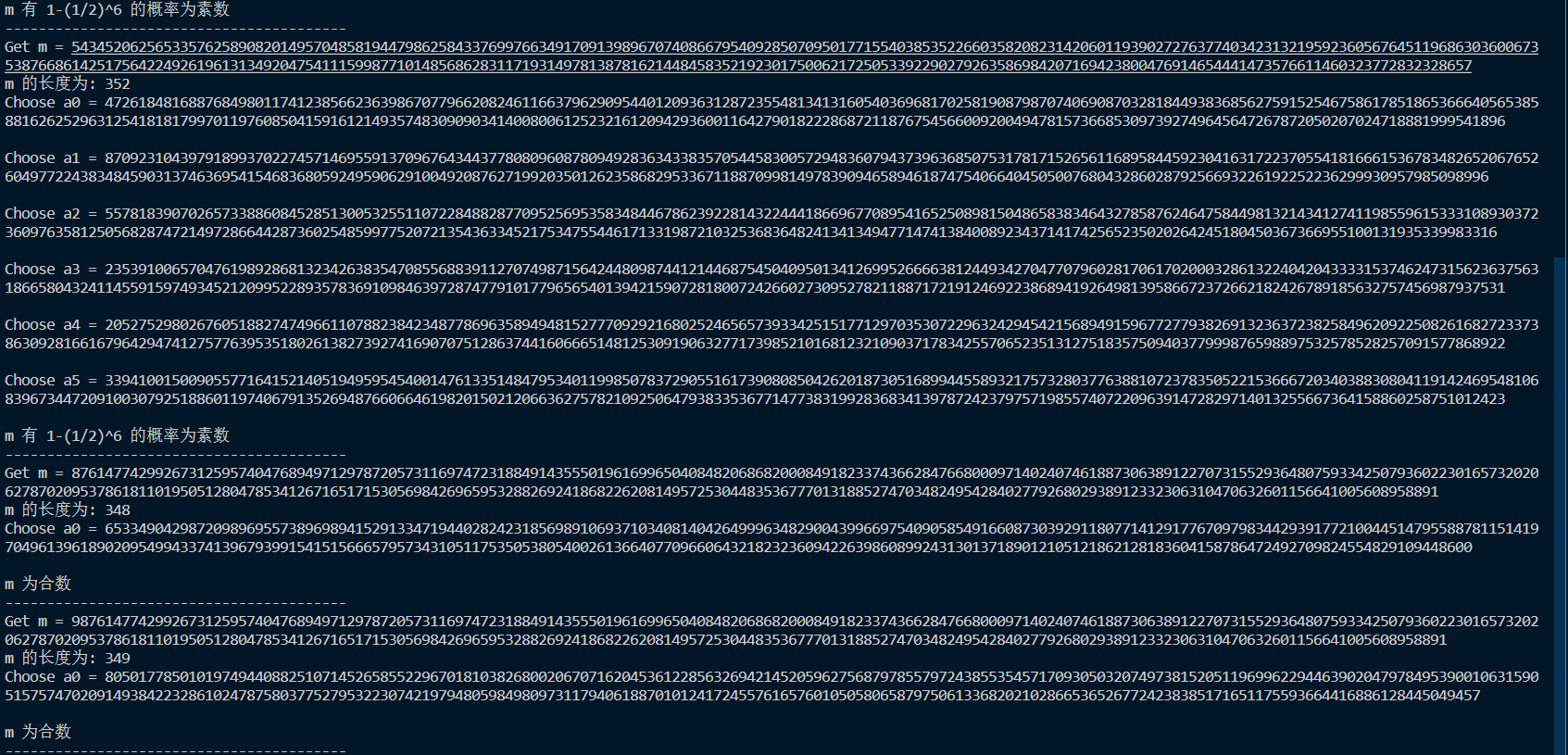
print("-----------------------------------------")#分隔符

四、数据分析(包括算法测试数据的分析，运行结果截图等等)

一共有四组数据，我们将其放在python列表中，然后遍历循环即可：

结果如下图：





五、思考与总结

1. 如果有一个整数𝒂，(𝒂,𝒎)=𝟏，使得𝒂𝒎−𝟏≡𝟏 𝒎𝒐𝒅 𝒎 则𝒎一定是一个素数吗？为什么？（请简述并举例说明，不能只简单回答“是”或“不是”）

不一定，比如说m=4 a=5；a^(m-1)=125≡1 mod 4 m就不是一个素数

1. Fermat素性检测中都用到了哪些运算？分别实现什么功能？请简述。

Gcd()求最大公因数：

#方法一：

a=int(input("请输入一个数："))

b=int(input("请输入另外一个数："))

**while**(a%b!=0):

MOD=a%b

a=b

b=MOD

print("gcd(a,b)=",b)

#方法二：

def gcd(a,b):

**if** b!=0:

**return** gcd(b,a%b)

**else**:

**return** a

print(gcd(15,8))

库中已经有实现，在这里自己提出两种实现

Pow()求快速模指数：

def quick\_algorithm(a,b,c):

a=a%c

ans=1

**while** b!=0:

**if** b&1:

ans=(ans\*a)%c

b>>=1

a=(a\*a)%c

**return** ans

print(quick\_algorithm(21,39,100))

自己复现了快速模指数算法。

1. 你还了解哪种素性检测算法？请简述，并分析其与Fermat素性检测算法的区别与联系。

1.爱拉托斯散 (Eratosthenes)筛法(确定检验)，但不适用于检验大素数。

2.利用素数的性质和概率方法即我们的Fermat素性检测算法，可以检验大的(拟/伪)素数。(概率检验)

由于2的时间花费远小于1，所以在实际运用中，我们将具有素数部分性质的一些数称为伪素数。

设n 是一个奇合数, 如果对整数b， (b, n)＝1，使得 b(n-1)≡1 (mod n) 成立，则称n为以b为基的拟素数(伪素数)。

1. 实验过程中还遇到了什么问题，如何解决的？通过该实验有何收获？

基本上没碰到什么大问题，这个实验作为第一个入门实验，难度较低，这次实验我体会密码学与数论的紧密联系，将理论化为实践。

**实验报告提交说明**：

1. 实验报告同时提交word文档与源代码（.c或者.py）。
2. 实验报告与源代码命名规则：实验1-学号-姓名，例如：实验1-20009200400-张三。
3. 请于截止日期前在西电智课平台（学在西电）提交相关文档，逾期未提交，该部分成绩记为0分，周知。