**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：**

**姓名：**

**学号：**

**日期：2024-10-20**

一、实验目的（包括实验环境、实现目标等等）

Python环境

实现一个函数is\_probable\_prime (number, repetitions, mode)

Number为待检验的数字，repetitions为检验重复次数，Mode为检验使用的方法，如fermat或millar-rabin

二、方案设计

（包括背景、原理、必要的公式、图表、算法步骤等等）

一个数是素数是这个数满足费马小定理的充分必要条件。



因此，费马小定理可以用于检验素数。给定一个数m，随机生成一个整数a（3<a<m-1）,如果a,m满足费马小定理，则m有概率是素数。如果不满足，则m一定是合数。

重复检验多次，可以以很高的准确性确定m是否为素数。

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）

算法流程：  
Step1：随机生成一个合适的。

Step2: 用扩展的欧几里得算法EEA计算gcd(a, m), 如果gcd=1，则转到下一步，如果gcd不为1，则跳出，返回m不是素数。

Step3：计算

Step4: 如果结果为1，则回到Step1，如果结果不为1，则跳出，返回m不是素数。

循环最多执行repetitions次

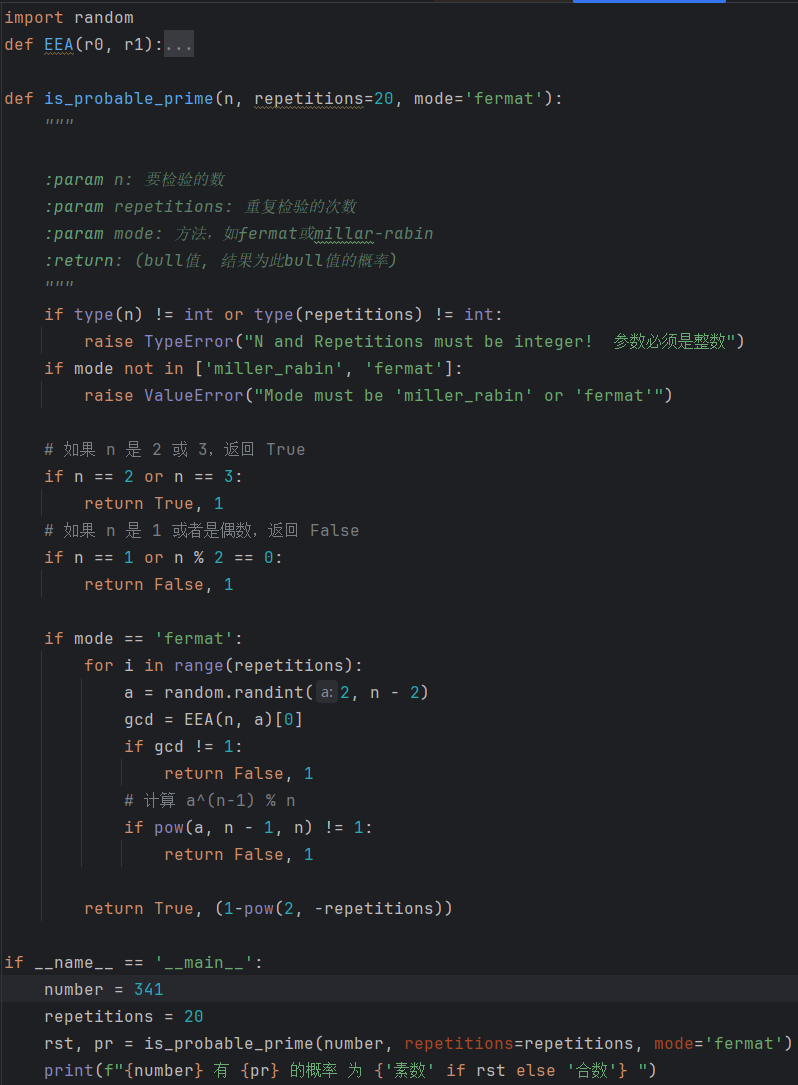
主要函数：

Pow(base, exp, mod) Python内置函数，其中mod可以省略，如果mod省略，则只计算幂，如果mod给出，则会运用快速模幂的方法计算模幂。本实验中，将使用后者的功能。

EEA(r0,r1) 运用扩展的欧几里得算法，输入两个整数r0,r1，将返回一个三元组

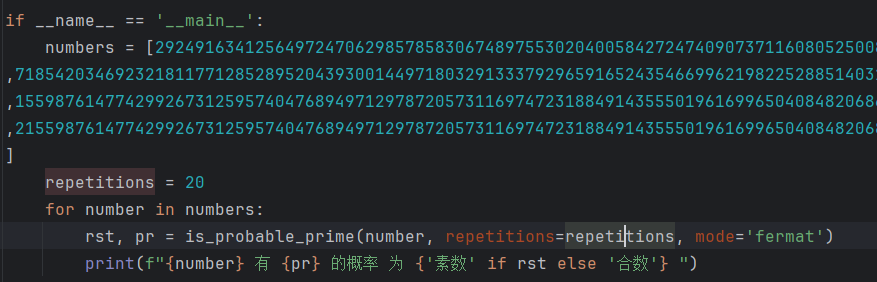
(gcd, s, t). 其中gcd = r0 \* s + r1 \* t

is\_probable\_prime (number, repetitions, mode) 实验要求实现的函数，返回结果为二元组，(bool\_val，pr)。Bool\_val为布尔值，表示检验结果，pr表示为此结果的概率。

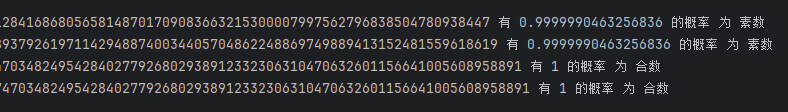


四、数据分析(包括算法测试数据的分析，运行结果截图等等)

测试数据都是极大的整数，其数量级为2^1160左右，因此，用常规方法(从1到n/2测试能否被整除)完全不可行。

用此main函数运行脚本。  


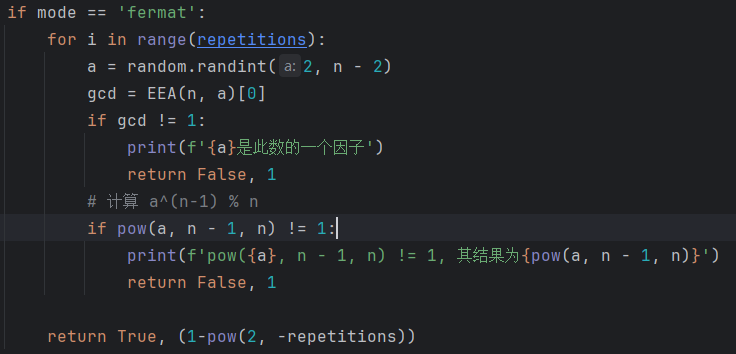
运行结果为



在检验20次的情况下，得到测试数据1、2有极大概率是素数，而测试数据3、4是合数。

选择测试数据3分析，我在is\_probable\_prime函数中加入一些代码，使得函数返回“合数”结果时，打印判定number为合数的原因。

代码如下所示：



运行结果：  
pow(1054607955307120636434226398804503519679821924057149421160387283154493566172612035908872757929230823137206600527202237640955768674401911148512511658814500607019045370467202764565143835716837537028228454316543312906747323503525023358528542560231438401415395751331311074236681799764585850304559014776934656677224830383945388511287065771833801413501149338, n - 1, n) != 1, 其结果为1400896696982005708987521241431563715111152479921066704822643418217607908633556519170501486592002180168887549776183021548860911178625352697077784883647054157062851972481059182386080366594408009987316758666497430929796000735520253728522057595110626275948734021662028924650789270432278013350550586887872061999182938804699725450892659773650986278963981517

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”）

不是，这是费马小定理的表述公式。素数一定满足费马小定理，但是满足费马小定理的数不一定是素数。

例如2^340 mod 341 = 1,但341是合数

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。

随机数生成算法，用于生成随机数

扩展的欧几里得算法EEA，用于计算gcd

快速模幂算法，用于计算指数极大时的模幂。当指数极大时，先算出幂，再计算余数不现实，而快速模幂算法可以大大提升效率，复杂度为log2(n)

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。

Millar-rabin素性检测。

联系：

两者都是基于费马小定理，通过模幂运算来检测素性，可以说费马素性检测是millar-rabin素性检测的基础

区别：

Miller-Rabin素性检测额外进行一系列平方运算。将数字分解为d \* 2^s

Fermat素性检测有更高的概率误判一个合数为素数（伪素数），Miller-Rabin素性检测则通过多次测试降低了这种概率。

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？

实验过程较为简单，没有遇到其他问题。

在密码学课程中，生成RSA密钥需要选择两个大素数p,q，而常用的办法就是随机生成大整数，使用素性检验算法检测之，如果是素数就采用。

理论与实践在本次实验中融会贯通，使我不仅学到了检测素数的办法，而且还温习了密码理论课的知识，受益匪浅

**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。