

第一周实验报告

- 课程名称：进阶式挑战性综合项目 I
- 组号：第二组
- 成员：韦博，张一博，张颖，胡焯维
- 指导教师：文淑华
- 实验日期：2025-9-3
- 提交日期：2025-9-8

第一步

经过我们组内讨论，我们决定使用 python 作为我们的开发语言，因为其具备较为完善的库环境，且代码风格简洁，也易学习，更有利于我们之后对项目的扩展。之后我们按照要求进行了 python 的下载，组内成员都为 Windows 环境，也利于后续代码的编写。

第二步

在 cmd 中运行 `pip install cryptography` 后，可以实现 python 加密库的安装

```
C:\Users\Lenovo>python
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cryptography
>>> print(cryptography.__version__)
45.0.7
>>> from cryptography.fernet import Fernet
>>> exit()
```

在终端中运行 `print(cryptography.__version__)` 时，能正确返回版本值；

在执行 `from cryptography.fernet import Fernet` 时没有报错，证明安装成功了；

运行 `pip install gmpy2` 后，可以实现数学运算库的安装，同样的

```
C:\Users\Lenovo>python
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import gmpy2
>>> print(gmpy2.__version__)
2.2.1
>>> from gmpy2 import mpz, mpq, mpfr, mpc
>>> a = mpz(12345678901234567890)
>>> b = mpz(98765432109876543210)
>>> print(a * b)
1219326311370217952237463801111263526900
>>> exit()
```

我们也可以用上述方法验证 gmpy2 的安装

另外，我们也可以进一步测试其基本功能，图片中大整数 a 和 b 相乘

除此以外，我们还可以安装别的库，例如：

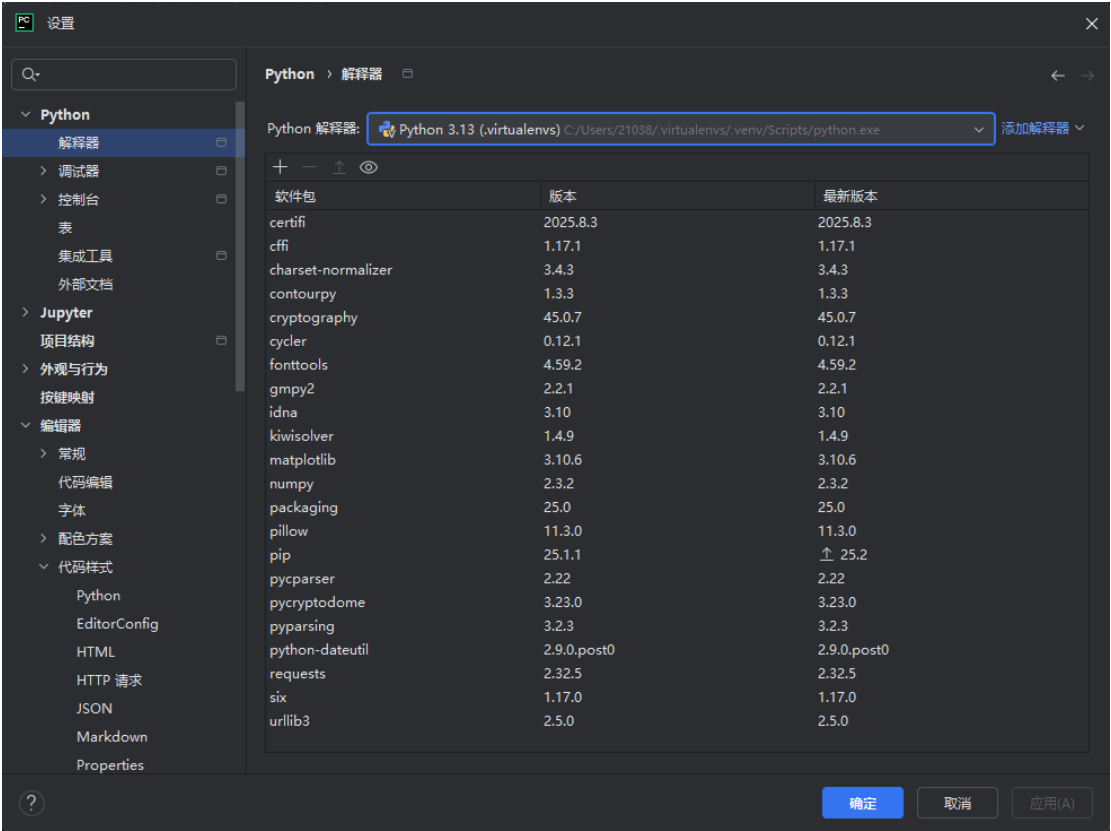
`pip install numpy` # 数值计算

`pip install matplotlib` # 图表绘制

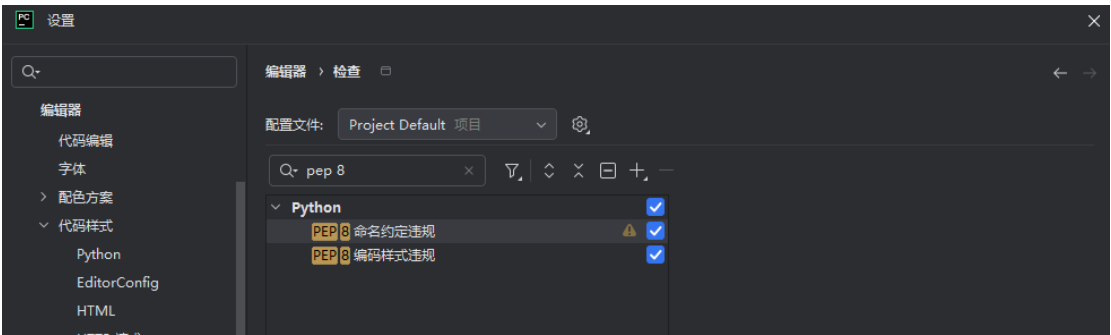
`pip install requests` # HTTP 请求

第三步

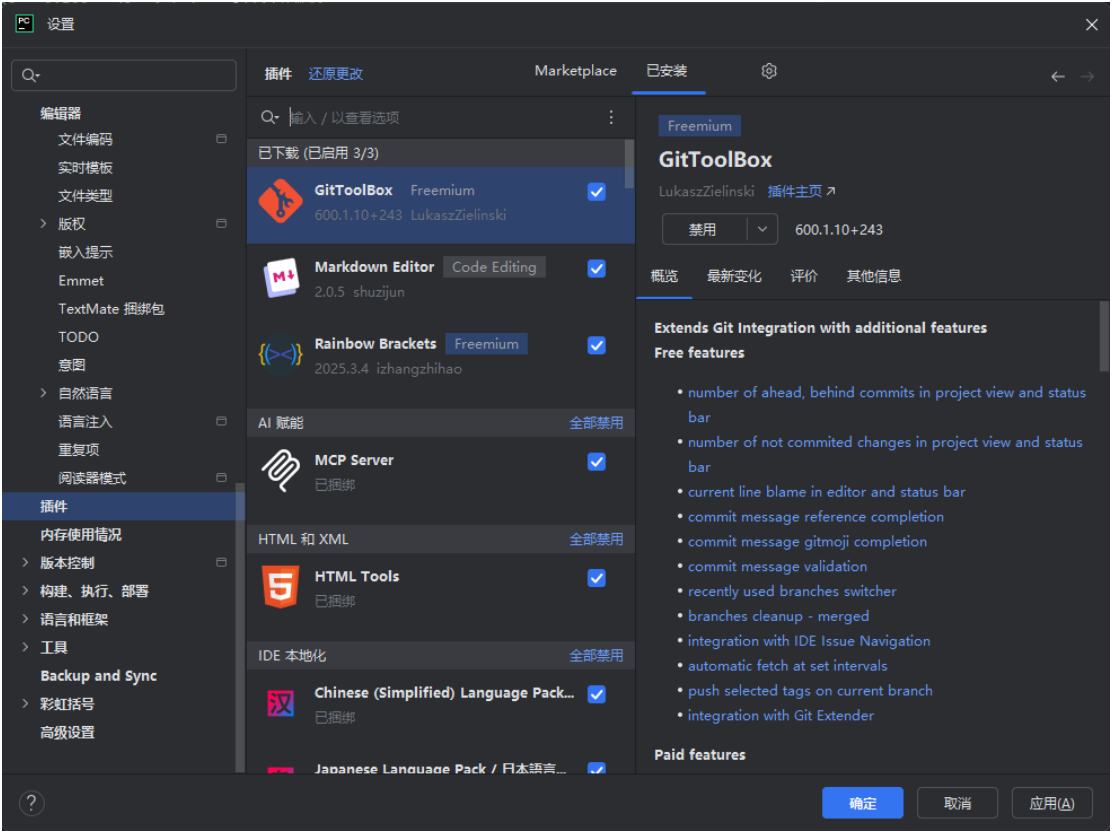
安装好 Pycharm 后，可从设置页面打开 python 的解释器选项，依循先前安装 python 解释器的文件路径添加解释器，即完成了 python 解释器路径的设置。



设置页面当中往下可以通过搜索 PEP 8 以达到配置代码风格的目的。

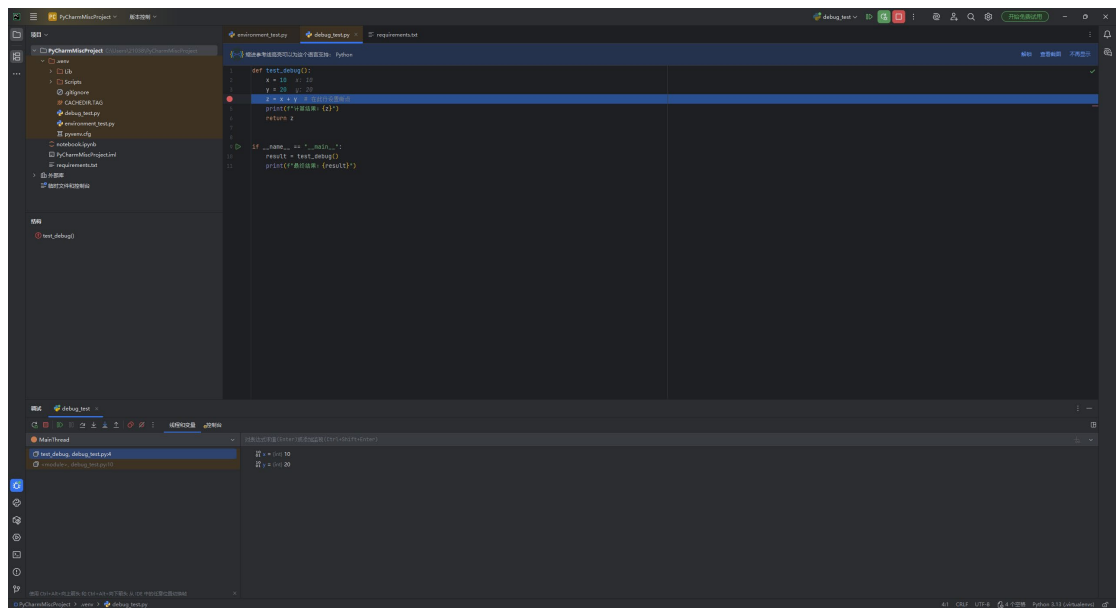


同样的，在设置页面选择插件选项，搜索并安装有用的插件。

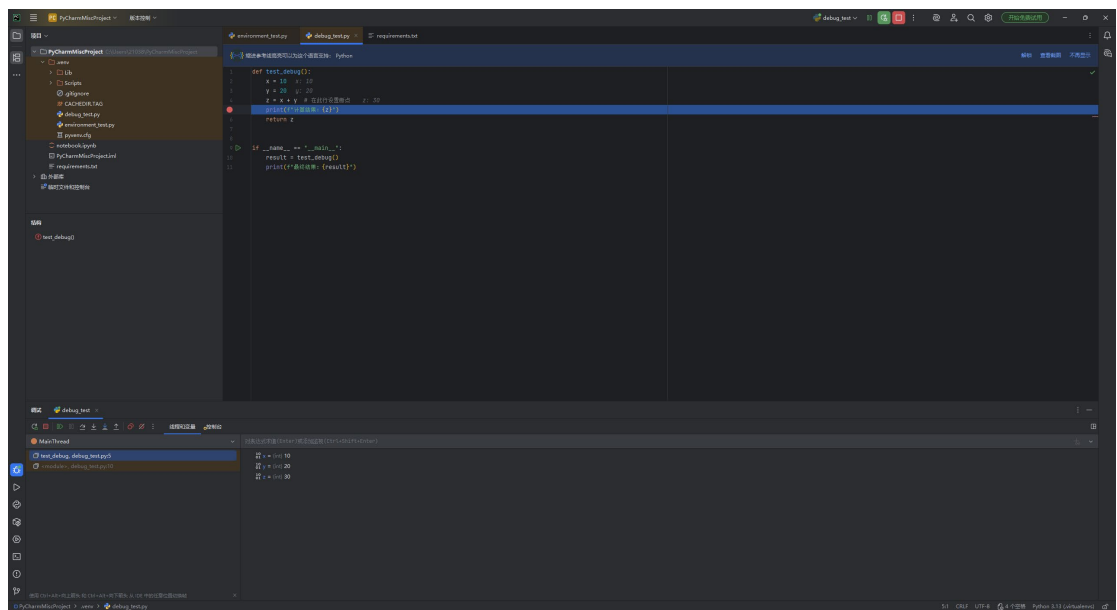


完成路径配置等前置工作后，我们可在编辑器中新建 python 文件做调试环境的验证。

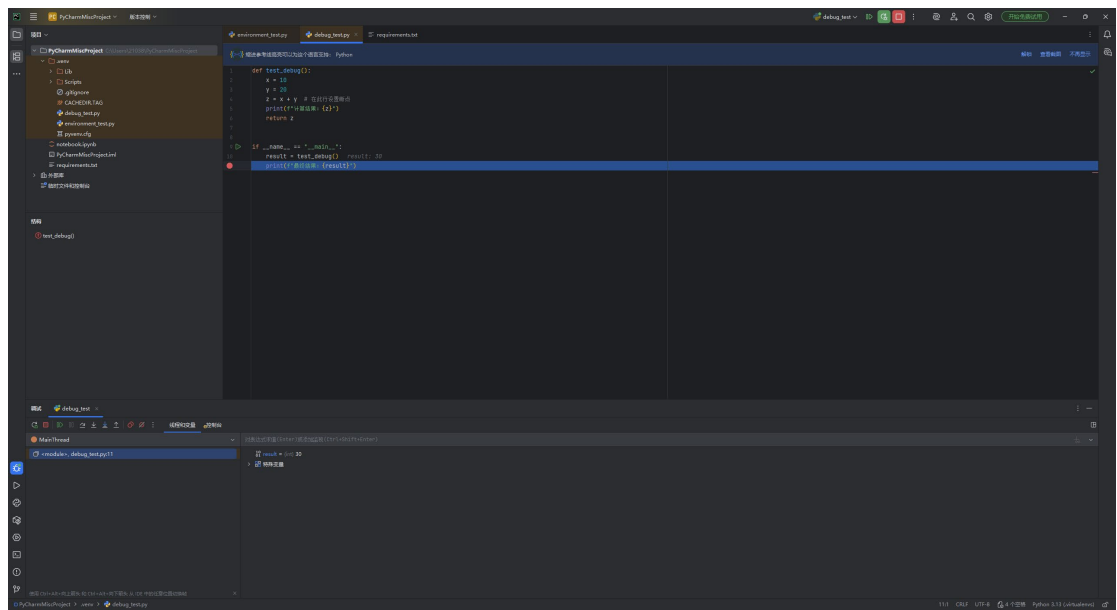
如图，在代码第四行设置断点并启动调试模式，可观察到下方终端显示此时 x, y 变量值分别为 10 和 20。



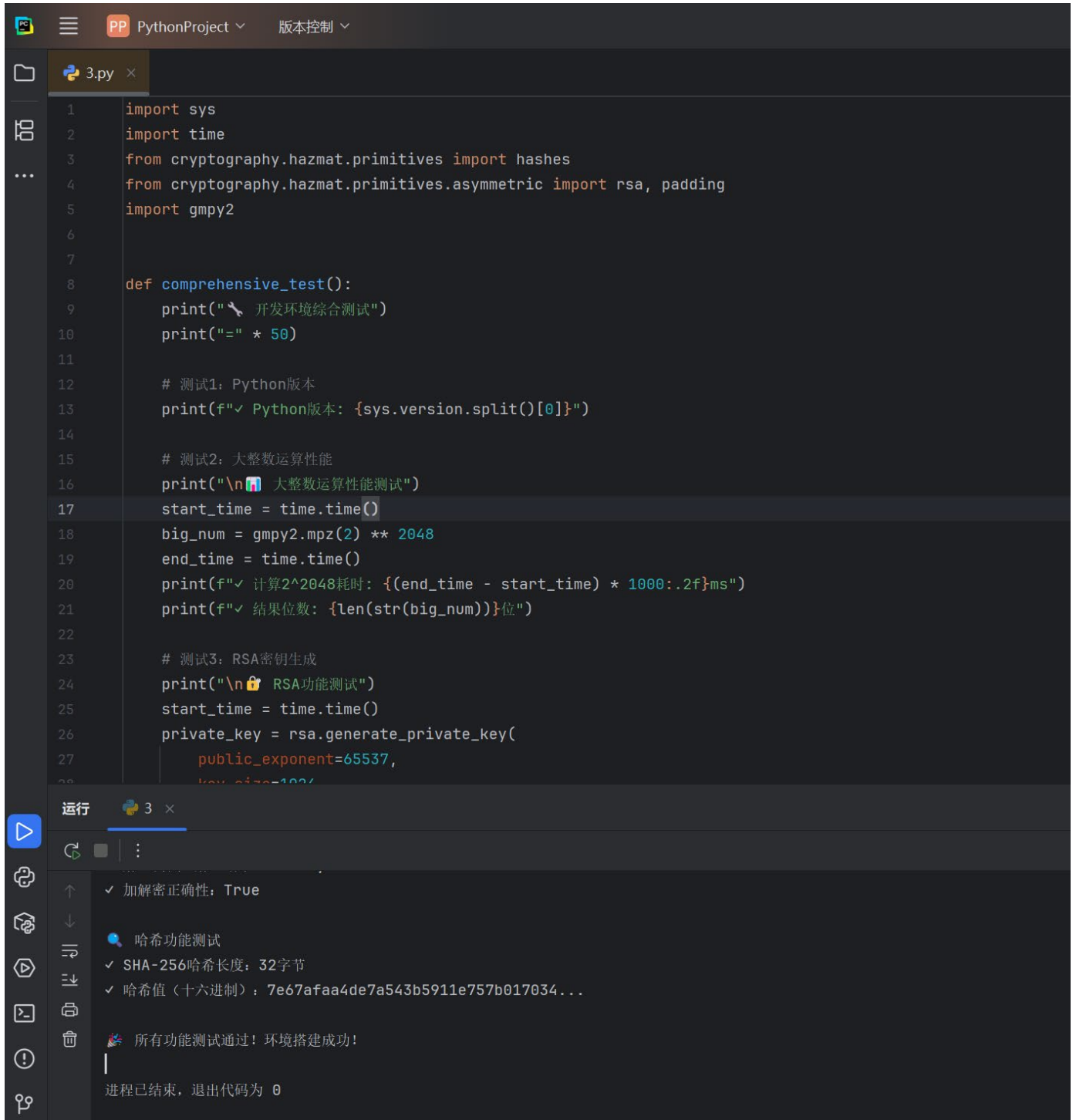
接下来我们单步执行代码，当把断点设置到第五行后，可观察到终端显示此时 z 变量值为 30。



最后再把断点设置在最后一行，可以看到终端显示此时 result 变量值为 30，x，y，z 变量因处在 test_debug () 函数内部，而此函数在最后已被调用完毕，所以终端调试窗口并没有显示 x，y，z 的值。



第四步



The screenshot displays the PyCharm IDE interface. The top toolbar shows icons for file operations, a menu, and project settings. The main editor window is titled '3.py' and contains the following Python code:

```
1 import sys
2 import time
3 from cryptography.hazmat.primitives import hashes
4 from cryptography.hazmat.primitives.asymmetric import rsa, padding
5 import gmpy2
6
7
8 def comprehensive_test():
9     print("\n 🚧 开发环境综合测试")
10    print("=" * 50)
11
12    # 测试1: Python版本
13    print(f"✓ Python版本: {sys.version.split()[0]}")
14
15    # 测试2: 大整数运算性能
16    print("\n 🧮 大整数运算性能测试")
17    start_time = time.time()
18    big_num = gmpy2.mpz(2) ** 2048
19    end_time = time.time()
20    print(f"✓ 计算2^2048耗时: {(end_time - start_time) * 1000:.2f}ms")
21    print(f"✓ 结果位数: {len(str(big_num))}位")
22
23    # 测试3: RSA密钥生成
24    print("\n 🗝️ RSA功能测试")
25    start_time = time.time()
26    private_key = rsa.generate_private_key(
27        public_exponent=65537,
```

The bottom panel, titled '运行' (Run), shows the execution output for '3.py':

```
✓ 加解密正确性: True
🔍 哈希功能测试
✓ SHA-256哈希长度: 32字节
✓ 哈希值 (十六进制): 7e67afaa4de7a543b5911e757b017034...
🎉 所有功能测试通过! 环境搭建成功!
|
进程已结束, 退出代码为 0
```

实验结果：

✓ Python 版本: 3.12.3

大整数运算性能测试

✓ 计算 2^{2048} 耗时: 1.03ms

✓ 结果位数: 617 位

RSA 功能测试

✓ 生成 1024 位 RSA 密钥耗时: 13.12ms

✓ 加密测试：原文 'Hello, Secure E-commerce!' -> 密文长度 128 字节


✓ 解密测试：解密结果 'Hello, Secure E-commerce!'

✓ 加解密正确性：True

哈希功能测试

✓ SHA-256 哈希长度：32 字节

✓ 哈希值（十六进制）：7e67afaa4de7a543b5911e757b017034...

 所有功能测试通过！环境搭建成功！

实验步骤：

1.python 官网下载相应版本 python，并进行环境和包管理器配置。

2. 下载 pycharm，安装插件，进行解释器配置，创建一个 python 项目。
3. 配置相关库，numpy，gmpy2，cryptography 等。
4. 利用测试函数测试 python 环境和版本，大整数运算性能测试，RSA 功能测试，
哈希功能测试，最终全部通过，结果已给出。

第五步

我们都在自己的环境中创建了项目文件夹，并进行了 git 初始化，便于之后的项目合作，也配置了项目的.gitignore 文件，又创建了 README.md 文件。在项目初始化完成后，我们又创建了 github 仓库，并进行了远程仓库的连接，最后成功上传，为之后的项目工程合作铺平了道路。

secure-ecommerce-systemPublic

PinWatch0

main2 Branches0 Tags

Go to file

Add file

Code

myLeiMu [docs]更改READMEe1c715e · 20 minutes ago3 Commits

docs/requirements	[docs]完善格式	2 hours ago
src/tests	[docs]完善格式	2 hours ago
.gitignore	初始化项目：环境搭建完成	last week
README.md	[docs]更改README	20 minutes ago

README

安全电子商务系统项目

项目简介

本项目是《进阶式挑战性综合项目 I》的实践项目，目标是设计并实现一个安全的电子商务系统。

技术栈

- 开发语言：Python 3.9+
- 密码学库：cryptography, gmpy2
密码学库：cryptography, gmpy2
- 开发工具：PyCharm/VSCode

团队成员

- myLeiMu：算法实现
- diomand-bo：测试验证
- wnsjdh：文档编写
- Eddiez0616：系统集成