

作业 1

1. 定义一个函数，用欧几里得算法寻找两个数的最大公约数。

- 算法：

- (1) 将两个数赋值给 a 和 b，使 $a > b$;
- (2) 将 b 的值赋值给 a;
- (3) 将没有改变的 a 的值除以 b 的余数赋给 b，重复步骤 (2) 和 (3) 直到 b 值为 0，停止计算，此时 a 的值为这两个数的最大公因数。

- 提示：

- (1) x 除以 y 的余数可以用 $x\%y$ 得到;
- (2) 可以使用循环或者迭代来计算。

参考： 12 和 18 的最大公约数为 6

2. 设计一个矩阵相乘函数，可以完成矩阵之间的相乘计算，具体框架如下所示。

```
import numpy as np

def matrix_multiply(a,b):
    #判断两矩阵是否可以相乘
    #如果可以，利用循环进行计算，返回结果

# 主函数
def main():
    a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]], dtype = np.float32)
    b = np.array([[5, 8, 3], [9, 4, 2], [1, 6, 7]], dtype = np.float32)
    # 调用相乘函数
    result = matrix_multiply(a,b)

    # 比较结果
    print(result) # 输出自己的计算结果
    print(np.dot(a, b)) # 输出 numpy 计算的结果

# 进入主函数
if __name__ == '__main__':
    main()
```

作业要求：

1. 完成时间为2周，上完课两周内提交代码；
2. 提交文件为*.py格式代码、公约数代码与矩阵相乘代码打包成一个文件夹；
3. 文件夹命名规则：学号-姓名-作业编号，示例：123456-张三-作业01
4. 作业提交到邮箱(458121574@qq.com)；