L1506：数字选取 ✩✩✩

题目描述

给定正整数 n，现在有 $1,2,\ldots,n$ 共计 n 个整数。你需要从这 n 个整数中选取一些整数，使得所选取的整数中任意两个不同的整数均互质（也就是说，这两个整数的最大公因数为 1）。请你最大化所选取整数的数量。

例如，当 $n=9$ 时，可以选择 $1,5,7,8,9$ 共计 5 个整数。可以验证不存在数量更多的选取整数的方案。

输入输出格式

输入：一行，一个正整数 n，表示给定的正整数。

对于 40% 的测试点，保证 $1 \le n \le 1000$。对于所有测试点，保证 $1 \le n \le 10^5$。

输出：一行，一个正整数，表示所选取整数的最大数量。

输入示例	输出示例
6	4

样例解释

当 $n=6$ 时，可以选择 $1,2,3,5$ 或 $1,3,4,5$ 等方案，都能选取 4 个整数。可以验证无法选取 5 个整数。

算法分析

1. **问题分析**

   我们需要从 1 到 n 中选取尽可能多的数，使得任意两个数互质。

2. **关键观察**

   - **数字 1 与任何数都互质**： 1 必须选
   - **质数之间必定互质**，因为质数的定义是只有 1 和自身两个正因子
   - **一个重要的策略**：我们可以选择所有的质数和 1，因为它们两两互质
   - **合数一定不能选**：因为合数必然能拆成若干个质数相乘。例如 $6=2\times 3$，并且这些质数必然小于该合数，所以这些质数（例如 2,3）已经选取了，再选取 6 会导致 6 和 3 不互质。

3. **贪心策略**

   （1） 选择数字 1（与所有数互质）

   （2） 选择所有不超过 n 的质数（质数之间互质）

4. 试除法判断质数

- 原理：检查 2 到 $\sqrt{n}$ 之间是否有 n 的因子
- 时间复杂度：$O(\sqrt{n})$
- 代码实现：

```cpp
bool is_prime(int n) {
    if (n < 2) return false;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) return false;
    }
    return true;
}
```

5. 完整算法步骤

（1） 初始化计数器 $cnt = 1$（包含数字 1）

（2） 遍历 2 到 n 的所有整数

（3） 对每个整数 i，使用试除法判断是否为质数

（4） 如果是质数，计数器 cnt 加 1

（5） 输出最终的 cnt 值

✍ 其他常见的质数筛

方法名称	时间复杂度	空间复杂度	适用场景
试除法	$O(\sqrt{n})$	$O(1)$	单个质数判断
埃拉托斯特尼筛法	$O(n \log\log n)$	$O(n)$	批量质数筛选
欧拉筛法	$O(n)$	$O(n)$	高效批量筛选
米勒-拉宾素性检验	$O(k \log^3 n)$	$O(1)$	大数质数判定
分段筛法	$O(n \log\log n)$	$O(\sqrt{n})$	大范围质数筛选

拓展思考

- 如果数字范围扩大到 10^7 或更大，算法需要如何优化？
- 还有很多其他的筛选质数的方法，比如线性筛、欧拉筛等等，你能解释一下它们的相较于其他方法的优缺点吗？