

1 功能需求

年月日问题（NextDate），核心需求是：给定 1800 年到 2050 年之间的某个有效日期，计算其下一天的日期；对于无效日期，则提示输入有误，直接返回。

2 测试用例设计

边界值测试可从某输入条件的最大取值范围开始，然后在该范围中寻找其他较为明显的边界，且应遵循独立性假设。

第二日问题的输入可拆分为 3 个输入条件：年份、月份、日期，其中，

- 1. 年份的最大取值范围是在需求中明确指出的，从 1800 年到 2050 年；
- 2. 月份的最大取值范围可根据常识得到，从 1 月到 12 月；
- 3. 日期的最大取值范围仍根据常识得到，从 1 号到 31 号（尽管不同月份包含的日期数不同，但独立性假设要求日期的边界应忽略年份和月份对其产生的影响，可接受的日期最大值应为 31）。

测试用例清单如下：

| 输入 | | | 预期输出 | 备注 |
|-------|-------|-----|--------------------------------|-------|
| yeaar | month | day | | |
| 1799 | 5 | 21 | 提示“年份请输入一个在 1800 到 2050 之间的整数“ | 年份的边界 |
| 1800 | 5 | 21 | 1800-5-22 | |
| 1801 | 5 | 21 | 1801-5-22 | |
| 2049 | 5 | 21 | 2049-5-22 | |
| 2050 | 5 | 21 | 2050-5-22 | |
| 2051 | 5 | 21 | 提示“年份请输入一个在 1800 到 2050 之间的整数“ | |
| 2022 | 0 | 14 | 提示“月份请输入一个在 1 到 12 之间的整数“ | 月份的边界 |
| 2022 | 1 | 14 | 2022-1-15 | |
| 2022 | 2 | 14 | 2022-2-15 | |
| 2022 | 11 | 6 | 2022-11-7 | |
| 2022 | 12 | 6 | 2022-12-7 | |
| 2022 | 13 | 6 | 提示“月份请输入一个在 1 到 12 之间的整数“ | |
| 2023 | 4 | 0 | 提示“日期请输入一个在 1 到 31 之间的整数“ | 日期的边界 |
| 2023 | 4 | 1 | 2023-4-2 | |
| 2023 | 4 | 2 | 2023-4-3 | |
| 2023 | 4 | 30 | 2023-5-1 | |
| 2023 | 4 | 31 | 提示“该日期不存在“ | |

| | | | | |
|------|---|----|---------------------------|--|
| 2023 | 4 | 32 | 提示“日期请输入一个在 1 到 31 之间的整数” | |
|------|---|----|---------------------------|--|

3 再分析

我们坚持缺陷具有独立性的前提是否正确，是否有待考证，我们需要从系统业务层面进行考虑。独立性假设排除了平年闰年边界的影响，排除了不同月份天数的影响。实际上从系统业务角度来说，覆盖率十分的低，难以发现程序潜在的错误。

我们如果考虑到这些因素，我们可能会重新得到这样合法的数据范围。（假设平年）

| | |
|----|--|
| 0 | |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |
| 13 | |

所以从业务的需要出发，我们需要加入这样几种可能的数据：

1. 某一典型平年月日边界的测试用例。（设所取邻域仍为 1）
 - a) 1, 3, 5, 7, 8, 9, 10, 12 月的 30, 31, 32 日等测试数据。
 - b) 4, 6, 9, 11 月的 30, 31, 32 日等测试数据。
 - c) 2 月的 27, 28, 29 日测试数据。
2. 闰年 2 月的测试用例（选择某些典型闰年 N）
 - a) $N \bmod 4 = 0$, 27, 28, 29, 30 日测试数据。
 - b) $N \bmod 100 = 0$, 27, 28, 29, 30 日测试数据。
 - c) $N \bmod 400 = 0$, 27, 28, 29, 30 日测试数据。