

全国青少年信息学奥林匹克竞赛模拟赛

CCF NOIP2025 模拟赛

时间：2025 年 10 月? 日 07:30 ~ 12:00

题目名称	Happy · Lovely · Everyday!	敬启, 致那时的 我	Lead to shine more	永恒的盛宴
题目类型	传统题	传统题	传统题	传统题
目录	a	b	c	d
可执行文件名	a	b	c	d
输入文件名	a.in	b.in	c.in	d.in
输出文件名	a.out	b.out	c.out	d.out
测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	4	4	6	6
是否捆绑测试	是	是	是	是

提交源程序文件名

对于 C++	a.cpp	b.cpp	c.cpp	d.cpp
--------	-------	-------	-------	-------

编译选项

对于 C++	-lm -std=c++14 -O2 -Wl,--stack=998244353
--------	--

注意事项与提醒（请选手务必仔细阅读）

1. 选手提交的源程序必须存放在已建立好的，且带有样例文件和下发文件的的文件夹中，文件名称与对应试题英文名一致；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 ulimit -s unlimited 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 若无特殊说明，每道题的代码大小限制为 100KB。
9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。

10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

【样例输出 1】

```
1 4
2 7
3 92
4 2445
5 52038
6 514228
7 3183186
```

【样例 2】

见选手目录下的 *a/a2.in* 与 *a/a2.ans*。
该样例满足子任务 3 的限制。

【样例 3】

见选手目录下的 *a/a3.in* 与 *a/a3.ans*。

【样例解释】

对于样例一，能得到的本质不同的序列为 00, 0, 11, 110；
对于样例二，能得到的本质不同的序列为 0, 00, 011, 101, 11, 110, 1111。

【测试点约束】

本题使用捆绑测试且开启所有合理的子任务依赖。
对于 100% 的数据， $1 \leq n, \sum n \leq 2 \times 10^6$ 。
每个子任务的具体限制见下表：

子任务编号	$\sum n \leq$	特殊性质	分数
1	10	无	20
2	20	无	30
3	2×10^6	<i>a</i> 序列元素全为 1	30
4	2×10^6	无	20

敬启，致那时的我 (b)

【题目描述】

实乃理给你两个整数 S, k ，你需要帮她求出以下式子的值对 $1,000,000,007$ 取模的结果：

$$\sum_{X=0}^S [\text{popc}(X) = k] F(X)$$

其中 F 为斐波那契数列，即 $F(0) = F(1) = 1, F(n) = F(n-1) + F(n-2)$ 。 $\text{popc}(X)$ 表示 X 的二进制表示中 1 的个数。

【输入格式】

从文件 ***b.in*** 中读入数据。

第一行两个整数 n, k 。

第二行一个长度为 n 的 01 串 S_2 ，代表 S 的二进制表示。保证 S_2 没有前导零。

【输出格式】

输出到文件 ***b.out*** 中。

输出一行一个整数表示答案对 $1,000,000,007$ 取模的结果。

【样例输入 1】

```
1 3 0
2 110
```

【样例输出 1】

```
1 1
```

【样例输入 2】

```
1 3 1
2 110
```

【样例输出 2】

1 8

【样例输入 3】

1 3 2

2 110

【样例输出 3】

1 24

【样例输入 4】

1 17 8

2 11011111101010010

【样例输出 4】

1 894224396

【样例 5】

见选手目录下的 *b/b5.in* 与 *b/b5.ans*。

该样例满足特殊性质 B 的要求。

【样例 6】

见选手目录下的 *b/b6.in* 与 *b/b6.ans*。

该样例满足特殊性质 C 的要求。

【样例 7】

见选手目录下的 *b/b7.in* 与 *b/b7.ans*。

【样例解释】

对于样例 1，答案为 $F(0) = 1$ ；
对于样例 2，答案为 $F(1) + F(2) + F(4) = 1 + 2 + 5 = 8$ ；
对于样例 3，答案为 $F(3) + F(5) + F(6) = 3 + 8 + 13 = 24$ 。

【测试点约束】

本题使用捆绑测试且开启所有合理的子任务依赖。
对于 100% 的数据， $0 \leq k \leq n \leq 2 \times 10^3$ ， $n, S \geq 1$ 。
每个子任务的具体限制见下表：

子任务编号	特殊性质	分数
1	A	20
2	B	30
3	C	30
4	无	20

特殊性质 A： $S \leq 10^6$ 。
特殊性质 B： $k \leq 2$ 。
特殊性质 C： S 可以表示为 $2^y - 1$ 的形式，其中 y 是一个正整数。

Lead to shine more (c)

【题目描述】

遥有一个正整数 x , 初始时为 0 时刻, 且 $x = 1$ 。每过一个时刻 x 都会变成 $x + \text{popc}(x)$, 给定正整数 n , 请你告诉她 x 最早变为 n 的时刻, 或不存在这样的时刻。其中 $\text{popc}(x)$ 表示 x 的二进制表示中 1 的个数。

【输入格式】

从文件 *c.in* 中读入数据。

本题单个测试点内包含多组数据。

第一行一个整数 T 表示数据组数。

接下来, 对于每组数据, 一行一个正整数 n 。

【输出格式】

输出到文件 *c.out* 中。

对于每组数据, 输出一行一个整数表示答案。若不存在这样的时刻则输出 -1 。

【样例输入 1】

```
1 7
2 3
3 114
4 114514
5 998244365
6 999999999981
7 9999999999999999
8 100000000000000000
```

【样例输出 1】

```
1 2
2 37
3 -1
4 68784060
5 51582151974
```


6 34098371561762190

7 -1

【样例解释】

x 前几个时刻的值为：1, 2, 3, 5, 7, 10, 12, 14, 17, ...。

【测试点约束】

本题使用捆绑测试且开启所有合理的子任务依赖。
对于 100% 的数据， $1 \leq T \leq 10^5$ ， $1 \leq n \leq 10^{18}$ 。
每个子任务的具体限制见下表：

子任务编号	$n \leq$	$T \leq$	分数
1	10^7	1	10
2	10^9	1	10
3	10^{12}	1	20
4	10^{12}	50	20
5	10^{18}	1	10
6	10^{18}	10^5	30

永恒的盛宴 (d)

【题目描述】

有 n 只「新鲜的食物」，分别编号为 $1, 2, \dots, n$ 。现在第 i 只「新鲜的食物」锁定了第 a_i 只「新鲜的食物」为目标（显然 $a_i \neq i$ ），且所有的「新鲜的食物」都是正常状态。

有一个行动顺序 p (p 为一个 $1 \sim n$ 的排列)，它们会按照行动顺序依次行动。假设现在是第 t 回合，则第 p_t （以下以 c 代指 p_t ）只「新鲜的食物」会做如下行动：

- 尝试吃掉第 a_c 只「新鲜的食物」。
- 如果第 a_c 只「新鲜的食物」没有被吃掉，且处于虚弱状态，则第 a_c 只「新鲜的食物」被吃掉，第 c 只「新鲜的食物」状态不变。
- 否则第 a_c 只「新鲜的食物」无法被吃掉，第 c 只「新鲜的食物」因饥饿而进入虚弱状态，第 a_c 只「新鲜的食物」状态不变。

如果行动顺序 p 在所有 $n!$ 种 $1 \sim n$ 的排列里均匀随机，那么期望有多少只「新鲜的食物」会被吃掉？输出答案对 $998,244,353$ 取模的值。

【输入格式】

从文件 **d.in** 中读入数据。

第一行一个整数 n ，代表「新鲜的食物」个数。

第二行一行 n 个正整数，第 i 个数表示第 i 只「新鲜的食物」锁定的目标编号 a_i 。

【输出格式】

输出到文件 **d.out** 中。

输出一行一个整数，表示答案对 $998,244,353$ 取模的结果。

【样例输入 1】

```
1 4
2 2 4 1 3
```

【样例输出 1】

```
1 499122178
```

【样例输入 2】

```
1 5
2 2 3 1 2 3
```

【样例输出 2】

```
1 191330169
```

【样例输入 3】

```
1 6
2 6 1 4 5 1 4
```

【样例输出 3】

```
1 390979040
```

【样例 4】

见选手目录下的 *d/d4.in* 与 *d/d4.ans*。

该样例满足子任务 2 的限制。

【样例 5】

见选手目录下的 *d/d5.in* 与 *d/d5.ans*。

该样例满足子任务 5 的限制。

【样例解释】

样例一的答案为 $\frac{3}{2}$ 。

如果 $p = [4, 1, 3, 2]$ ，那么会发生以下行动：

1. 第 4 只「新鲜的食物」尝试吃掉第 3 只「新鲜的食物」失败，进入虚弱状态；
2. 第 1 只「新鲜的食物」尝试吃掉第 2 只「新鲜的食物」失败，进入虚弱状态；
3. 第 3 只「新鲜的食物」尝试吃掉第 1 只「新鲜的食物」成功；
4. 第 2 只「新鲜的食物」尝试吃掉第 4 只「新鲜的食物」成功。

共有两只「新鲜的食物」被吃掉。

样例二的答案为 $\frac{161}{120}$ 。

如果 $p = [2, 1, 4, 5, 3]$ ，那么会发生以下行动：

- 1. 第 2 只「新鲜的食物」尝试吃掉第 3 只「新鲜的食物」失败，进入虚弱状态；
- 2. 第 1 只「新鲜的食物」尝试吃掉第 2 只「新鲜的食物」成功；
- 3. 第 4 只「新鲜的食物」尝试吃掉第 2 只「新鲜的食物」失败（因为已经被第 1 只吃掉了），进入虚弱状态；
- 4. 第 5 只「新鲜的食物」尝试吃掉第 3 只「新鲜的食物」失败，进入虚弱状态；
- 5. 第 3 只「新鲜的食物」尝试吃掉第 1 只「新鲜的食物」失败，进入虚弱状态。

共有一只「新鲜的食物」被吃掉。

样例三的答案为 $\frac{209}{120}$ 。

【测试点约束】

本题使用捆绑测试且开启所有合理的子任务依赖。

对于 100% 的数据， $1 \leq n \leq 500, \forall i \in [1, n], i \neq a_i$ 。

每个子任务的具体限制见下表：

子任务编号	$n \leq$	特殊性质	分数
1	10	无	10
2	500	A	15
3	40	无	15
4	100	无	15
5	500	B	15
6	500	无	30

特殊性质 A： a_i 是一个 $1 \sim n$ 的排列。

特殊性质 B： 每个 a_i 都在 $[1, i - 1] \cup [i + 1, n]$ 共 $n - 1$ 个整数中等概率随机生成。