

NOI 2025 模拟赛

CDQZ

测试时间：2025.06.16

题目名称	卡德	卡特	秋思
题目类型	传统型	传统型	传统型
目录	card	cat	choose
可执行文件名	card	cat	choose
输入文件名	card.in	cat.in	choose.in
输出文件名	card.out	cat.out	choose.out
提交文件名	card.cpp	cat.cpp	choose.cpp
时间限制	1.0 秒	1.0 秒	2.0 秒
内存限制	1024 MiB	512 MiB	512 MiB
子任务数目	20	10	7
测试点是否等分	是	是	否
编译选项	-O2 -std=c++14		

【注意事项（请仔细阅读）】

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹；
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。
14. 统一评测时采用的机器配置为：12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz，内存 16GB。上述时限以此配置为准。
15. 评测在 Windows 11 下进行，使用 LemonLime 进行评测。如果对此条以及 14 条中的机器配置有疑问，请及时询问。
16. 题目按照字典序排序，不保证题目按照难度排序。

卡德 (card)

【题目背景】

Hanghang 在一个巨大的数轴上。

【题目描述】

给定一个长度为 n 的正整数序列 a ，设 $S = n + \sum_{i=1}^n a_i$ 。小 H 有 S 张卡片，每张卡片上都写着一个数，其中 n 张上面分别写着 a_1, a_2, \dots, a_n ，其余 $\sum_{i=1}^n a_i$ 张卡片上都写着 -1 。

小 H 现在站在数轴的坐标 0 处，他将执行以下操作 S 次：

- 假设小 H 现在站在坐标 x 处，他会选择一张卡片并将其丢弃，设这张卡片上的数为 x ，则小 H 会跳到坐标 $x + v$ 处。如果他刚好跳到了坐标 0 处，则会获得一枚硬币。

现在对于所有 $k \in [1, n]$ ，你要求出有多少种选牌的顺序，使得小 H 最终恰好会获得 k 枚硬币，答案对 998244353 取模。

注意，对于两张写的数相同的牌，先选其中一张和先选另一张算同一种方案。

【输入格式】

从 *card.in* 中读入数据。

第一行一个正整数 n 。

第二行 n 个正整数，表示 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 *card.out* 中。

输出 n 行，每行一个数。第 i 行的数表示当 $k = i$ 时的方案数对 998244353 取模的结果。

【样例 1 输入】

```
1 2
2 1 1
```

【样例 1 输出】

```
1 2
2 4
```

【样例 1 解释】

共有 4 张卡牌，上面的数分别为 $1, 1, -1, -1$ 。一共有 6 种不同的方案，以下两种方案会恰好获得 1 枚硬币： $1, 1, -1, -1$ 和 $-1, -1, 1, 1$ ，其余 4 种方案都会恰好获得 2 枚硬币。

【样例 2 输入】

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

【样例 2 输出】

```
1 140
2 220
3 144
```

【样例 3】

见选手目录下的 *card/card3.in* 与 *card/card3.ans*。
该样例满足测试点编号 2 ~ 3 的限制条件。

【样例 4】

见选手目录下的 *card/card4.in* 与 *card/card4.ans*。
该样例满足测试点编号 7 ~ 8 的限制条件。

【样例 5】

见选手目录下的 *card/card5.in* 与 *card/card5.ans*。
该样例满足测试点编号 9 ~ 10 的限制条件。

【样例 6】

见选手目录下的 *card/card6.in* 与 *card/card6.ans*。
该样例满足测试点编号 13 ~ 14 的限制条件。

【样例 7】

见选手目录下的 *card/card7.in* 与 *card/card7.ans*。
该样例满足测试点编号 17 ~ 18 的限制条件。

【样例 8】

见选手目录下的 *card/card8.in* 与 *card/card8.ans*。
该样例满足测试点编号 19 ~ 20 的限制条件。

【测试点约束】

对于 100% 的数据，满足 $1 \leq n \leq 5000, 1 \leq a_i \leq 10^9$ 。

测试点编号	$n \leq$	$a_i \leq$
1	5	2
2 ~ 3	10	5
4 ~ 6	12	10^9
7 ~ 8	20	
9 ~ 10	500	1
11 ~ 12		2
13 ~ 14	50	50
15 ~ 16	150	150
17 ~ 18	500	500
19 ~ 20	5000	10^9

卡特 (cat)

【题目描述】

有一天，无聊的猫猫们想出了一个新游戏。由一只睿智的猫给出了一个长度为 n 的排列，它们想知道这个排列是不是一个好排列。

一个排列是好的，当且仅当对于所有正整数 m 都满足，排列中所有长度为 $2m + 1$ 的子串的中位数都不在这个子串的第 $m + 1$ 个位置。

但是，这个给出排列太长了，小猫只记住了一些位上的数。所以它会给你一个一些数被替换成 -1 的排列，你需要将 -1 填入所有可能的值后，统计好的排列数量。

答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 *cat.in* 中读入数据。

第一行一个正整数 t ，表示有 t 组数据。

接下来对于每组数据，第一行一个正整数 n 。

第二行 n 个整数，表示一个长为 n 的排列。

【输出格式】

输出到文件 *cat.out* 中。

对于每组数据，输出一行一个整数表示可能的好的排列数量，答案对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

```
1 5
2 2
3 -1 -1
4 3
5 -1 -1 -1
6 4
7 1 2 3 4
8 6
9 -1 -1 3 4 -1 -1
10 8
11 -1 -1 -1 -1 -1 -1 -1 -1
```

【样例 1 输出】

```
1 2
2 4
3 0
4 1
5 316
```

【样例 2 输入】

```
1 7
2 6
3 -1 -1 2 -1 -1 -1
4 3
5 -1 -1 -1
6 4
7 3 4 2 1
8 8
9 -1 -1 -1 -1 -1 -1 -1 -1
10 5
11 -1 -1 -1 -1 -1
12 7
13 -1 -1 -1 -1 -1 -1 -1
14 2
15 -1 -1
```

【样例 2 输出】

```
1 13
2 4
3 0
4 316
5 24
6 136
7 2
```

【样例 2】

见选手目录下的 *cat/cat2.in* 与 *cat/cat2.ans*。

该样例满足子任务 1 的限制条件。

【样例 3】

见选手目录下的 *cat/cat3.in* 与 *cat/cat3.ans*。
该样例满足子任务 2 的限制条件。

【样例 4】

见选手目录下的 *cat/cat4.in* 与 *cat/cat4.ans*。
该样例满足子任务 3 的限制条件。

【测试点约束】

对于 100% 的数据，满足 $1 \leq t \leq 10^4, 2 \leq n \leq 10^6, \sum n \leq 10^6, a_i \in \{-1\} \cup [1, n]$ 。保证 $\forall 1 \leq i < j \leq n, a_i \neq -1, a_j \neq -1$ ，满足 $a_i \neq a_j$ 。

测试点编号	特殊限制
1	$\forall 1 \leq i \leq n, a_i = -1$
2	$\sum n \leq 8$
3 ~ 5	$\sum n^2 \leq 10^6$
6 ~ 10	$\sum n \leq 10^6$

秋思 (choose)

【题目描述】

有一张 n 行 m 列的网格图，记从上往下第 i 行、从左往右第 j 列的格子为 (i, j) ，每个格子 (i, j) 上有两个权值 $a_{i,j}, b_{i,j}$ 。

现在你要选出一些格子，使得不存在两个（上下左右）相邻的格子同时被选择，且选择的格子的 $(\sum a_{i,j})^2 + (\sum b_{i,j})^2$ 最大。求出这个值最大是多少。

【输入格式】

从 `choose.in` 中读入数据。

第一行输入两个整数 n, m 。

接下来 n 行，每行输入 m 个整数，第 i 行第 j 个整数表示 $a_{i,j}$ 。

接下来 n 行，每行输入 m 个整数，第 i 行第 j 个整数表示 $b_{i,j}$ 。

【输出格式】

输出到文件 `choose.out` 中。

输出一行一个整数表示答案。

【样例 0 输入】

```
1 2 2
2 1 2
3 3 4
4 1 3
5 1 2
```

【样例 0 输出】

```
1 41
```

【样例 1】

见选手目录下的 `choose/choose1.in` 与 `choose/choose1.ans`。

该样例满足子任务编号 1 的限制条件。

【样例 2】

见选手目录下的 *choose/choose2.in* 与 *choose/choose2.ans*。
该样例满足子任务编号 2 的限制条件。

【样例 3】

见选手目录下的 *choose/choose3.in* 与 *choose/choose3.ans*。
该样例满足子任务编号 3 的限制条件。

【样例 4】

见选手目录下的 *choose/choose4.in* 与 *choose/choose4.ans*。
该样例满足子任务编号 4 的限制条件。

【样例 5】

见选手目录下的 *choose/choose5.in* 与 *choose/choose5.ans*。
该样例满足子任务编号 5 的限制条件。

【样例 6】

见选手目录下的 *choose/choose6.in* 与 *choose/choose6.ans*。
该样例满足子任务编号 6 的限制条件。

【样例 7】

见选手目录下的 *choose/choose7.in* 与 *choose/choose7.ans*。
该样例满足子任务编号 7 的限制条件。

【测试点约束】

本题使用捆绑测试。

对于 100% 的数据，满足 $n, m \geq 1, 1 \leq n \times m \leq 3000, 1 \leq a_{i,j}, b_{i,j} \leq 10000$ 。

子任务编号	$nm \leq$	特殊性质	分值
1	16	无	5
2	81	有	5
3	196		5
4	3000		10
5	36	无	15
6	81		15
7	3000		45

特殊性质： $\forall 1 \leq i \leq n, 1 \leq j \leq m, b_{i,j} = 0$ 。