

# NOIP 模拟赛

QDEZ zxb

题目名称	考试	围棋	相互抵消	串串
题目类型	传统型	传统型	传统型	传统型
输入文件名	test.in	go.in	offset.in	string.in
输出文件名	test.out	go.out	offset.out	string.out
时间限制	2s	3s	1s	1s
内存限制	512MB	512MB	512MB	512MB
子任务数目	20	25	20	20
子任务是否等分	是	是	是	是

## 提交源程序文件名

对于 C++	test.cpp	go.cpp	offset.cpp	string.cpp
--------	----------	--------	------------	------------

## 注意事项

- 1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
- 2. 编译选项：`-std=c++14 O2`
- 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内存限制与题目的内存限制一致。
- 7. 题目不难，AK 了请不要大声喧哗。

# 考试 (test)

## 题目描述

有  $n$  名学生完成了一份包含  $m$  道题目的测试。对于每个学生，已知他答对了哪些题目，哪些没有答对。

每道题的分数为  $p_j$ 。由于某种特殊原因， $p$  是一个 1 到  $m$  的排列。如果某名学生答对了第  $j$  道题，他将获得  $p_j$  分（否则得 0 分）。

对于第  $i$  名学生，他期望在测试中获得  $x_i$  分。定义测试结果的“惊讶值”定义为  $\sum_{i=1}^n |x_i - r_i|$ ，其中  $r_i$  表示第  $i$  名学生实际获得的分数。

请找到一种排列  $p$ ，使得“惊讶值”最大。如果有多种答案，输出任意一种即可。

## 输入格式

本题多测。

第一行包含一个整数  $t$ ，表示测试用例的数量。

每个测试用例的第一行包含两个整数  $n$  和  $m$ ，分别表示学生人数和题目数量。

第二行包含  $n$  个整数  $x_1, x_2, \dots, x_n$ ，其中  $x_i$  表示第  $i$  名学生期望获得的分数。

接下来  $n$  行，每行一个长度为  $m$  的字符串  $s_i$ 。其中  $s_{i,j}$  为 1 表示第  $i$  名学生答对了第  $j$  道题，否则为 0。

## 输出格式

对于每个测试用例，输出  $m$  个整数，表示一种排列  $p$ ，使得“惊讶值”最大。如果有多种答案，输出任意一种即可。

## 样例

### 输入 #1

```
3
4 3
5 1 2 2
110
100
101
100
4 4
6 2 0 10
1001
0010
0110
0101
3 6
20 3 15
010110
000101
111111
```

## 输出 #1

```
3 1 2
2 3 4 1
3 1 4 5 2 6
```

## 说明/提示

由于本题需要 spj，所以下发文件的 `.out` 会给出最大“惊讶值”。

### 【样例 2】

见选手目录下的 `ex_test2.in` 和 `ex_test2.out`，该样例满足测试点 4 ~ 5 的限制。

### 【样例 3】

见选手目录下的 `ex_test3.in` 和 `ex_test3.out`，该样例满足测试点 6 ~ 7 的限制。

### 【样例 4】

见选手目录下的 `ex_test4.in` 和 `ex_test4.out`，该样例满足测试点 8 ~ 10 的限制。

### 【样例 5】

见选手目录下的 `ex_test5.in` 和 `ex_test5.out`，该样例满足测试点 11 ~ 20 的限制。

## 数据范围

对于所有测试数据，保证  $t \leq 10^4, n \leq 10, \sum m \leq 10^4, 0 \leq x_i \leq \frac{m(m+1)}{2}, s_{i,j} \in \{0, 1\}$ 。

测试点	$n \leq$	$m \leq$	特殊性质
1 ~ 3	10	7	无
4 ~ 5	1	$10^4$	无
6 ~ 7	10	$10^4$	A
8 ~ 10	10	$10^4$	B
11 ~ 20	10	$10^4$	无

特殊性质 A:  $n \leq m$ ，每个人都恰好答对一道题，并且每道题至多被一个人答对。

特殊性质 B:  $x_i \in \{0, \frac{m(m+1)}{2}\}$ 。

# 围棋 (go)

围棋，是一种开放世界冒险游戏对抗性游戏。

## 题目描述

一些概念：

- 黑白色的石子称为“棋子”，分别称为黑棋和白棋。在本题中分别用 `x` 和 `o` 表示。空地用 `.` 表示。
- 与一颗棋子直接相邻（四连通）的空地叫作这颗棋子的“气”。一块空地可能是多颗棋子的气。
- 两颗同色棋子  $a, b$  在一个连通块中，当且仅当存在一系列与之同色的棋子  $s_1, s_2, \dots, s_k$ ，使得  $s_1 = a, s_k = b$ ，且  $\forall 1 \leq i < k$ ，都有  $s_i$  与  $s_{i+1}$  直接相邻。
- 一块棋（即同色棋子连通块）是活棋，当且仅当这块棋中存在一颗棋子，使得它有至少一口气；反之，则是死棋。**在本题中我们只关心白棋的死活情况。**例如，下面的第一张图中的白棋没有气，所以是死棋；第二张图中白棋的气用 `*` 特别注明；第三张图中，右边的白棋是死棋——即使左边的黑棋也是死棋。

```
.xx.   .*x.   .ox.
xoox   xoo*   oxox
.xx.   .x*.   .ox.
```

给定一个  $n * n$  的棋盘，计算白棋有多少棋子是死的。当然数据并不是来自真实世界，所以棋盘不一定是  $19 \times 19$  的。

但是这个问题太简单了。你要对每颗棋子（包括每一颗黑棋和每一颗白棋）都计算出，将这颗棋子的颜色翻转后（即黑棋变白棋，白棋变黑棋），白棋有多少棋子是死的。注意**翻转是独立的**。

## 输入格式

本题多测。

输入第一行包含一个整数  $T$  表示测试样例数量。

对于每个测试样例：

第一行包含一个整数  $n$ ，表示棋盘的边长。

接下来的  $n$  行，第  $i$  行包含一个字符串  $s_i$ ，且  $|s_i| = n, s_{i,j} \in \{'x', 'o', '.'\}$ ，分别表示第  $i$  行第  $j$  列是黑棋、白棋、空地。

## 输出格式

对于每个测试样例输出一行一个整数  $E \bmod 10^9 + 7$  作为答案。 $E$  的计算方式如下：

- 对所有棋子进行排序，行号（从上到下）为第一关键字，列号（从左到右）为第二关键字。
- $E = \sum_{i=1}^m (10^6 + 7)^{m-i} a_i$ ，其中  $m$  是棋子的总数， $a_i$  是翻转第  $i$  颗棋子后，白棋的死棋子的数量。

请注意**模数和基数是不同的**。

# 样例

## 输入 #1

```
3
2
.o
..
3
.x.
xoo
ox.
2
oo
oo
```

## 输出 #1

```
0
870527216
485539347
```

# 样例解释

对于第二个测试样例，按照 (1, 2), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2) 的顺序翻转颜色，死亡的白棋的个数分别为 1, 0, 1, 2, 0, 0。

对于第三个测试样例，棋盘上的所有棋子，无论是黑棋还是白棋，都是死棋。

# 说明/提示

## 其他样例

见选手目录下的 `ex_go1~7.in` 和 `ex_go1~7.out`，它们分别满足测试点 1 ~ 3, 4 ~ 5, 6 ~ 7, 8 ~ 9, 10, 11 ~ 14, 14 ~ 25 的限制。

# 数据范围

对于所有测试数据，都有  $\sum n \leq 5 \times 10^3$ ，且  $2 \leq n \leq 10^3$ 。

测试点	$\sum n \leq$	特殊性质
1 ~ 3	100	无
4 ~ 5	500	无
6 ~ 7	$5 \times 10^3$	A
8 ~ 9	$5 \times 10^3$	B
10	$5 \times 10^3$	C
11 ~ 14	$5 \times 10^3$	D
15 ~ 25	$5 \times 10^3$	无

特殊性质 A：保证  $s_{i,j} \neq \text{'o'}$ 。

特殊性质 B：保证  $s_{i,j} \neq \text{'x'}$ 。

特殊性质 C：保证  $s_{i,j} \neq \text{'.'}$ 。

特殊性质 D：保证一开始白棋全是死棋。

# 互相抵消 (offset)

## 题目描述

给定长度为  $n$  的序列  $a$ ，进行  $q$  次操作，每次操作形如：

1. 给定  $l, r, x$ ，将区间  $[l, r]$  内的所有元素加  $x$ 。
2. 给定  $l, r$ ，查询  $\sum_{l'=l}^r \sum_{r'=l'}^r ((\sum_{i=l'}^{r'} a_i)^2 + (r-l+2) \times (r'-l') \times a_{l'} \times a_{r'})$ 。

对于所有操作二，输出相应答案模 998244353 的结果。

## 输入格式

本题强制在线。

输入第一行包含三个整数  $type, n, q$ ，表示强制在线参数、序列长度和操作次数。

接下来一行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ 。

接下来  $q$  行，每行先输入一个整数  $op$ ，表示操作类型。若  $op = 1$  则再输入三个整数  $l', r', x'$  表示一次操作一；若  $op = 2$  则再输入两个整数  $l', r'$  表示一次操作二。由于本题强制在线，不妨设上一次操作二的答案为  $lastans$ （初始为 0），对于操作一，需要将  $l', r', x'$  都异或上  $(lastans \times type)$  得到真实的  $l, r, x$ ；对于操作二，需要将  $l', r'$  都异或上  $(lastans \times type)$  得到真正的  $l, r$ 。

## 输出格式

输出包含若干行，对于每一个操作二，输出答案模 998244353 的结果。

## 样例

见选手目录下的 `ex_offset1~6.in` 和 `ex_offset1~6.out`。

# 数据范围

对于所有测试点，保证  
 $type \in \{0, 1\}, 1 \leq n, q \leq 5 \times 10^5, 0 \leq a_i, x < 998244353, op \in \{1, 2\}, 1 \leq l \leq r \leq n$ 。

测试点	$n, q \leq$	特殊性质
1 ~ 2	600	AB
3 ~ 4	$5 \times 10^3$	AB
5 ~ 6	$10^5$	A
7 ~ 8	$5 \times 10^5$	A
9	$10^5$	C
10 ~ 11	$5 \times 10^5$	C
12	$10^5$	D
13 ~ 14	$5 \times 10^5$	D
15 ~ 16	$10^5$	无
17 ~ 18	$3 \times 10^5$	无
19 ~ 20	$5 \times 10^5$	无

- 特殊性质 A：保证  $type = 0$ 。
- 特殊性质 B：保证询问为操作一的概率为  $\frac{3}{4}$ ，操作二的概率为  $\frac{1}{4}$ 。
- 特殊性质 C：保证仅存在至多一个操作二。
- 特殊性质 D：保证不存在操作一。



# 串串 (string)

小 Z 开了一家烧烤店，欢迎大家来品尝！

众所周知，串串上可以串很多食材，比如面包、鸡翅、豆腐。取决于你的口味。

## 题目描述

小 Z 的烧烤店里有  $n$  种食材。这些食材有魔力（它很厉害！），当  $\geq 2$  个相同的食材串在同一个烤串上时，烤串就会被食材强大的魔力击碎，这是小 Z 必须要避免的。我们认为，两个烤串相同，当且仅当它们的食材组成的集合是相同的。也就是说，小 Z 的烧烤店里一共有  $2^n$  种不同的烤串。特别的，单独一根烧烤签也是一种烤串。

有一天，小 L 和小 K 来到小 Z 的烧烤店。他们要点一些烤串，但是有一些特殊的要求。

- 他们不希望点相同的烤串，因为他们会觉得审美疲劳。
- 如果一种食材没有在所点的烤串中出现，他们会觉得非常遗憾；如果一种食材只出现一次，他们就会纠结该让谁吃。所以他们希望每种食材在所点的烤串中至少出现 2 次。

这时他们问小 Z：在  $2^{2^n}$  种点烤串的方法中，有多少符合他们的要求呢？小 Z 被难住了，于是求助于你。小 L 和小 K 并不想为难你，所以他们只需要你回答方案数模一个大质数  $m$  的结果。

## 输入格式

输入一行两个正整数  $n, m$ ，表示食材种类数和模数。

## 输出格式

输出一个整数，表示符合条件的方案数对  $m$  取模的结果。

## 样例

### 输入 #1

2 1000000007

### 输出 #1

2

### 输入 #2

3 1000000009

输出 #2

118

输入 #3

50 111111113

输出 #3

1456748

输入 #4

3000 123456791

输出 #4

16369789

数据范围

对于所有测试数据，保证  $2 \leq n \leq 5000, 10^8 \leq m \leq 10^9 + 9$ ，且  $m$  是质数。

测试点		
1 ~ 2	4	
3 ~ 5	16	
6	100	
7	300	
8	500	= 998244353
9	500	
10	700	= 998244353
11	700	
12	1000	= 998244353
13	1000	
14	3000	= 998244353
15 ~ 17	3000	

测试点	$n \leq$	$m$
18 ~ 20	5000	