

NOI 2025 模拟赛

题目名称	泉轻流花暖	岁云暮矣待月归	慕念萦心间
题目类型	传统型	传统型	传统型
目录名	yumemi	kofuru	kemuru
可执行文件名	yumemi	kofuru	kemuru
输入文件名	yumemi.in	kofuru.in	kemuru.in
输出文件名	yumemi.out	kofuru.out	kemuru.out
每个测试点时限	3.0 s	3.0 s	1.0 s
内存限制	1024 MiB	1024 MiB	1024 MiB

编译选项

对于 C++ 语言	-static -O2 -std=c++14
-----------	------------------------

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. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
- 3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。
- 4. 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。

泉轻流花暖 (yumemi)

【题目描述】

Yumemizuki Mizuki 有一个长为 $2n$ 的括号序列，但这个括号序列不一定是合法括号序列。

现在她可以对某些位置进行修改，具体地，她可以花费 a_i 代价将第 i 个位置的括号取反。她希望花费最小的总代价将括号序列修改为合法括号序列。

现在 Mizuki 会进行 q 次修改，每次会给出 i, v 表示将 a_i 修改为 v ，你需要在所有修改进行前以及每次修改之后输出将括号序列修改为合法括号序列所需的最小代价。

这里修改的影响是永久的，但括号序列不会真的被进行操作。

其中，合法括号序列的定义是：

- 空串是合法括号序列。
- 如果 A 是合法括号序列， (A) 也是合法括号序列。
- 如果 A, B 是合法括号序列， $A + B$ 也是合法括号序列。

所有合法括号序列都可以通过上述操作生成。

【输入格式】

从文件 `yumemi.in` 中读入数据。

第一行两个正整数 n, q 。

接下来一行 $2n$ 个非负整数 a_1, \dots, a_{2n} 。

接下来一行一个长为 $2n$ 的括号串。

接下来 q 行，每行两个数 i, x 表示把 a_i 修改为 x 。

【输出格式】

输出到文件 `yumemi.out` 中。

输出一行 $q + 1$ 个非负整数表示初始和进行完每次修改后的答案。

【样例 1 输入】

```
1 3 5
2 9 2 2 2 2 10
3 ()()((
4 3 7
5 5 10
6 6 5
7 2 2
8 6 2
```

【样例 1 输出】

```
1 14
2 14
3 14
4 9
5 9
6 6
```

【样例 1 解释】

初始的一种最优方案是：修改第 3,4,6 位置的括号，此时括号序列变为 $()()()$ ，总代价为 $2+2+10=14$ 。

【样例 2 输入】

```
1 5 10
2 5 10 6 8 3 6 2 1 16 11
3 (((()))(((((
4 4 9
5 9 9
6 2 11
7 4 20
8 7 9
9 1 5
10 9 16
11 8 4
12 2 18
13 4 20
```

【样例 2 输出】

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

【样例 3,4,5】

见选手目录下的 *yumemi/yumemi3,4,5.in* 与 *yumemi/yumemi3,4,5.ans*。

【数据范围与提示】

对于所有数据, $1 \leq n \leq 10^5, 1 \leq q \leq 5 \times 10^5, 0 \leq a_i, x \leq 10^9$ 。

每个子任务的具体约束如下:

- Subtask 1 (15 分): $n, q \leq 100$ 。
- Subtask 2 (15 分): $n, q \leq 1000$, 且 s 随机生成。
- Subtask 3 (15 分): $n, q \leq 1000$ 。
- Subtask 4 (10 分): $a_i = x = 1$ 。
- Subtask 5 (20 分): $q \leq 10^5$ 。
- Subtask 6 (25 分): 无特殊限制。

岁云暮矣待月归 (kofuru)

【题目描述】

Naganohara Yoimiya 有一个 $n \times n$ 的棋盘，现在她想让 Yumemizuki Mizuki 在上面摆放 m 个棋子。其中，棋盘的每个格子上最多摆放一枚棋子，或者不摆放棋子。

不过，Yoimiya 对最后的摆放方案有一些要求：

- 每行、每列上至少摆放一个棋子。
- 主对角线、副对角线上至少摆放一个棋子。这里主对角线指的是所有坐标形如 (i, i) 的格子，副对角线则是所有坐标形如 $(i, n - i + 1)$ 的格子。
- 此外还有 k 个特殊格子 $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ ，这些格子内不能放置棋子。

请你帮 Mizuki 求出有多少种符合 Yoimiya 要求的摆放方式。

对 $m = 0, 1, 2, \dots, n^2 - k$ 输出答案对 998244353 取模后的结果。

【输入格式】

从文件 **kofuru.in** 中读入数据。

第一行两个整数 n, k 。

接下来 k 行，每行两个正整数 x, y 表示一个限制格子。

【输出格式】

输出到文件 **kofuru.out** 中。

输出一行 $n^2 - k + 1$ 个整数表示 $m = 0, 1, \dots, n^2 - k$ 的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 0 0 0 0 4 10 6 1
```

【样例 1 解释】

以 $m = 4$ 为例：符合要求的方式有：

```
1 110 100 010 010
2 001 001 100 001
3 100 110 101 101
```

其中 1 表示放棋子，0 表示不放。共有 4 种方案。

【样例 2,3,4】

见选手目录下的 *kofuru/kofuru2,3,4.in* 与 *kofuru/kofuru2,3,4.ans*。

样例 2 满足测试点 3 ~ 5 的约束，样例 3 满足测试点 13 ~ 15 的约束，样例 4 满足测试点 21 ~ 25 的约束。

【数据范围与提示】

对于所有数据，保证 $1 \leq n \leq 64, 1 \leq k \leq 7$, k 个特殊格子的位置互不相同。

每个测试点的具体约束如下：

测试点编号	n	k
1 ~ 2	≤ 4	$= 0$
3 ~ 5	≤ 12	
6 ~ 8	≤ 20	
9 ~ 12	≤ 50	
13 ~ 15	≤ 16	≤ 5
16 ~ 20	≤ 32	≤ 6
21 ~ 25	≤ 64	≤ 7

慕念萦心间 (kemuru)

【题目描述】

终于就剩最后一个题啦，云浅决定不再扯那么多题目背景了，因此这道题的题面很简单：

设素数 $p = 998244353$ ；给定矩阵 $A \in M_{n \times n}(\mathbb{F}_p)$ ，问有多少 $B \in M_{n \times n}(\mathbb{F}_p)$ 满足 $AB = BA$ 。

更通俗地讲：给定 $n \times n$ 矩阵 A ，满足 $A_{i,j} \in [0, p-1]$ ；问有多少 $n \times n$ 矩阵 B 满足 $B_{i,j} \in [0, p-1]$ ，且 $AB = BA$ 。这里乘法是矩阵乘法，且加法乘法都在 $\text{mod } p$ 意义下进行。具体地， A, B 两个矩阵的乘积 C 定义为 $C_{i,j} = (\sum_{k=1}^n A_{i,k} B_{k,j}) \text{ mod } p$ 。

可以证明存在正整数 k 使得答案恰好为 p^k 。输出正整数 k 即可。

【输入格式】

从文件 **kemuru.in** 中读入数据。

第一行一个正整数 n 。

接下来 n 行每行 n 个非负整数，表示矩阵 A 的一行。

【输出格式】

输出到文件 **kemuru.out** 中。

输出一行一个非负整数 k ，表示答案是 p^k ，其中 $p = 998244353$ 。

【样例 1 输入】

```
1 3
2 1 2 3
3 4 5 6
4 7 8 9
```

【样例 1 输出】

```
1 3
```

【样例 2 输入】

```
1 3
2 1 1 0
3 0 1 0
4 0 0 1
```

【样例 2 输出】

```
1 5
```

【样例 3,4,5,6,7,8,9】

见选手目录下的 *kemuru/kemuru3,4,5,6,7,8,9.in* 与 *kemuru/kemuru3,4,5,6,7,8,9.ans*。
它们分别满足 Subtask 1,2,3,4,5,6,7 的限制。

【数据范围与提示】

对于所有数据，有 $1 \leq n \leq 500, 0 \leq A_{i,j} < 998244353$ 。

每个子任务的具体约束如下：

- Subtask 1 (20 分)： $n \leq 20$ 。
- Subtask 2 (4 分)： A 是单位矩阵，即所有 $A_{i,i} = 1$ ，其余的 $A_{i,j}$ 均为 0。
- Subtask 3 (6 分)：若 $i \neq j$ ，则 $A_{i,j} = 0$ 。
- Subtask 4 (24 分)： $n \leq 100$ ，且 $A^n = 0$ ，其中 0 表示全 0 矩阵。
- Subtask 5 (10 分)： $n \leq 100$ ，且只有 $i \leq j$ 的 $A_{i,j}$ 可能非零，且如果 $A_{i,j} \neq 0$ ，那么必有 $A_{i,i} = A_{j,j}$ 。
- Subtask 6 (11 分)： $n \leq 100$ 。
- Subtask 7 (25 分)：无特殊限制。

此外，所有 Subtask 中一共只有最多 60 组数据。