

南京大学第 19 届算法设计大赛

试题册

比赛时间：2023 年 5 月 3 日 13:30-17:30



主办单位：南京大学 ACM-ICPC 集训队

试题列表

- A Bobo String Construction
- B Bobo String Count
- C Classical Problem?
- D Election of the King
- E For the Treasury!
- F Merge the Squares!
- G Out of Context
- H Portal 3
- I Square Game
- J We are the Lights

请不要在比赛开始前打开试题册！

A. Bobo String Construction

时间限制: 1 seconds

空间限制: 2048 megabytes

Bobo 在研究编码, 以便向他的好朋友 **oboB** 传送信息。最近他研制出了一种“**Bobo 编码**”, 实际上就是一种二进制编码。但是和直接的二进制编码不同, “**Bobo 编码**”分为三部分: 开头的“起始识别串” t , 然后一段用来传递信息的“信息串” s , 以及结尾的和“起始识别串”相同的“终止识别串” t 。给定一个需要传输的“信息串” s , 以及一个识别串 t , 这个“信息串” s 的**Bobo 编码**即为 $t + s + t$, 其中 $+$ 表示字符串的拼接。传送信息时 **Bobo** 将他要传输的信息串加密成“**Bobo 编码**”逐个传输给 **oboB**, 而 **oboB** 在已接收到的串中只要发现出现了两次识别串, 就认为已经收到了“起始识别串”和“终止识别串”, 从而停止接收信息。注意这里识别串出现的次数可以重叠, 例如如果识别串 $s = 101$, 而 **oboB** 接收到了 10101 , 那么 **oboB** 就会认为识别串已经出现了两次, 从而停止接收信息。

Bobo 还在为他设计的“**Bobo 编码**”洋洋得意, 却丝毫没有意识到他犯了一个很大的错误! 这一天, **Bobo** 要向 **oboB** 传送的信息串是 1101 , 而他们约定的识别串是 010 , 这样这个信息串的“**Bobo 编码**”是 $010 + 1101 + 010 = 0101101010$, 然而 **oboB** 在接收到前 8 个字符 01011010 发现接收到的串里已经出现了两次识别串 010 , 因此错误地把 01011010 当成了“**Bobo 编码**”。

然而即使如此, **Bobo** 觉得他的“**Bobo 编码**”还可以抢救一下。他想知道, 在已知模式串 s 以及要发送的信息串的长度 n 情况下, 能否找到一个合适的信息串使得“**Bobo 编码**”能够被正确解读? 正式地说, 给定一个仅由 0 和 1 构成的字符串 t , 以及一个正整数 n , 你能否找到一个长度为 n 的仅由 0 和 1 构成的字符串 s , 使得 t 在 $t + s + t$ 中出现仅两次(即只在开头和结尾出现)?

输入格式

该题具有多组测试数据。在第一行输入一个正整数 T ($1 \leq T \leq 1000$), 表示测试数据的组数。

对于每组测试数据, 第一行输入一个正整数 n ($1 \leq n \leq 1000$), 表示 **Bobo** 要传输的信息串的长度。

每组测试数据的第二行输入一个仅由 0 和 1 构成的字符串 t ($1 \leq |t| \leq 1000$), 表示 **Bobo** 和 **oboB** 约定的模式串。

输出格式

对于每组测试数据, 如果不存在任何长度为 n 的仅由 0 和 1 构成的字符串 s 使得 t 在 $t + s + t$ 中出现仅两次, 在一行中输出 -1 , 否则输出一个满足条件的字符串 s 。如果存在多个满足条件的字符串 s , 输出其中任意一个都会被视为正确。

样例输入 1

2
6
101
3
0

样例输出 1

001100
111

B. Bobo String Count

时间限制: 1.5 seconds

空间限制: 2048 megabytes

经过了“Bobo 编码”的失败(有关“Bobo 编码”的信息,可以参考问题 *Binary String Construction* 的题目描述), Bobo 痛定思痛决定好好研究字符串, 他现在考虑了一个十分简单的问题:

给定一个仅由 0 和 1 组成的字符串 t 以及一个正整数 n , 对于 $k = 0, 1, 2, \dots, n$, 你需要计算有多少种长度恰好为 n 的仅由 0 和 1 组成的字符串 s 满足

- 字符串 t 在字符串 s 中出现恰好 k 次(注意这里的出现可以重叠, 如 101 在 10101 中出现了两次)。

由于答案可能过大, 你需要将答案对 998244353(一个质数) 取模。

输入格式

第一行输入一个正整数 n ($1 \leq n \leq 2000$)。

第二行输入一个仅由 0 和 1 构成的字符串 t ($1 \leq |t| \leq 2000$)。

输出格式

在一行中输出 $n + 1$ 个数, 依次表示 $k = 0, 1, \dots, n$ 时的答案。

样例输入 1

样例输出 1

10	631 339 53 1 0 0 0 0 0 0 0
1001	

样例输入 2

样例输出 2

10	1 10 45 120 210 252 210 120 45 10 1
0	

样例输入 3

样例输出 3

10	144 235 241 187 116 62 25 11 2 1 0
11	

C. Classical Problem?

时间限制: 3 seconds

空间限制: 2048 megabytes

一天, Bobo 遇到了以下题目:

- 给定两个正整数 n ($n \geq 1$) 和 k ($k \geq 2$), 计算 n 在 k 进制表示下的各位数字之和 $f_k(n)$ 。正式地说, 令序列 $(a_0, a_1, \dots, a_\ell)$ 满足以下条件:

- $a_\ell \neq 0$
- $\forall 0 \leq i < \ell, 0 \leq a_i \leq k - 1$
- $\sum_{i=0}^{\ell-1} a_i k^i = n$

不难证明这样的序列 $(a_0, a_1, \dots, a_\ell)$ 唯一。你需要计算 $f_k(n) = \sum_{i=0}^{\ell-1} a_i$ 。

“这也太简单了, 简直小菜一碟!”, Bobo 飞快地通过了这个题, 紧接着打开了下一题, 题目描述似曾相识:

- 给定两个正整数 n ($n \geq 1$) 和 K ($K \geq 2$), 你需要找到一个 $2 \leq k \leq K$ 使得 n 在 k 进制表示下的各位数字之和最小。你只需要输出这个最小值。正式地说, 你需要计算 $\min_{2 \leq k \leq K} f_k(n)$ 。

Bobo 陷入了沉思, 你能帮帮他吗?

输入格式

该题具有多组测试数据。在第一行输入一个正整数 T ($1 \leq T \leq 50$), 表示测试数据的组数。

对于每组测试数据, 在一行中输入两个正整数 n, K ($1 \leq n \leq 10^{18}, 2 \leq K \leq 10^{18}$), 其具体含义已经在题目描述中给出。

输出格式

对于每组测试数据, 在一行中输出一个数字表示 $\min_{2 \leq k \leq K} f_k(n)$ 。

样例输入 1	样例输出 1
4 15 4 987654321 9 987654321 12345678 1 100	3 15 15 1

样例解释

对于第一组测试数据, 最优方式是选择 $k = 3$, 此时 15 在 3 进制下的表示为 $(120)_3$, 各位数字之和为 $1 + 2 + 0 = 3$, 是能够达到的最小值。其中 $(x)_y$ 表示 x 是一个 y 进制数。

D. Election of the King

时间限制: 1 second

空间限制: 2048 megabytes

在遥远的 Boboland, 每隔五年就会进行一次国王选举。今年恰逢 Boboland 的又一次国王选举。Boboland 的每个城市已经推举出共 n 位国王候选人, 编号分别为 $1, 2, \dots, n$ 。这 n 位候选人拥有各不相同的政治倾向 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$ 表示编号为 i 的候选人的政治倾向, 数字越大代表政治倾向越右, 1 表示极左, 10^9 表示极右)。然后将举行以下候选人内部投票机制来决定最终的国王:

- 投票进行 $n - 1$ 轮, 每轮投票会恰好淘汰一名候选人, 直到最终剩下的唯一一名候选人成为最终的国王。
- 每轮投票规则如下: 每名候选人可以投给任意一名除他以外的候选人。最终得票最多的候选人会被淘汰。如果有多名得票同为最多的候选人, 会淘汰所有得票同为最多的候选人中政治倾向最右的那一位候选人。

你在观察了之前 Boboland 的每届国王选举, 发现每名候选人都秉承着党同伐异的理念, 在每轮投票中都会执行以下策略:

- 在所有剩余的其他候选人中, 投票给和自己政治倾向相差最大的其他候选人 (即 i 号候选人, 如果仍然没被淘汰, 会投票给 $|a_j - a_i|$ 最大的仍然没被淘汰的 j 号候选人)。如果存在多个和自己政治倾向相差最大的其他候选人, 会选择投票给所有和自己政治倾向相差最大的候选人中政治倾向最右的那一位候选人。

现在你想知道, 谁会在今年 Boboland 的选举中成为最终的国王?

输入格式

输入第一行包含一个正整数 n ($1 \leq n \leq 10^6$), 表示候选人的人数。

输入第二行包含 n 个数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9, a_i$ 互不相同), 表示每名候选人的政治倾向。

输出格式

在一行中输出一个数, 表示最终会成为国王的候选人的编号。

样例输入 1

4 5 1 8 10	1
---------------	---

样例输出 1

样例输入 2

4 10 1 9 6	3
---------------	---

样例输入 3

4 3 7 5 1	4
--------------	---

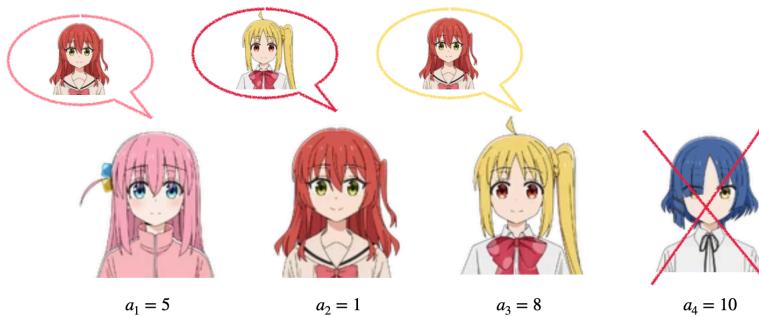
样例输出 3

样例解释

为了帮助理解，我们提供了第一个样例的图示说明。下图给出了第一轮的投票结果，气泡中说明了每个候选人会投票给谁。此时二号和四号候选人平票且同为最多，但是四号候选人政治倾向更右（即 $a_4 > a_2$ ），故淘汰四号候选人。



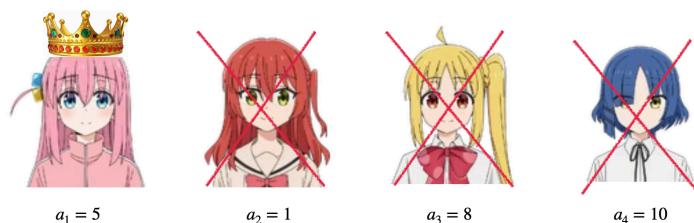
下图给出了第二轮的投票结果，此时二号候选人得票最多，故淘汰四号候选人。



下图给出了第三轮的投票结果，此时一号和三号候选人平票且同为最多，但是三号候选人政治倾向更右（即 $a_3 > a_1$ ），故淘汰三号候选人。



因此最终一号候选人当选国王。



E. For the Treasury!

时间限制: 1 second

空间限制: 2048 megabytes

公元 11 世纪初，有一些被称为维京人的丹麦入侵者在英格兰活动。

Askeladd 是一群维京海盗的领袖，他在这片肥沃的土地上寻找宝藏。在一夜袭击了一个村庄后，他们共收集了 n 个宝藏，按顺序排列，价值分别为 a_1, a_2, \dots, a_n 。**Askeladd** 的团队有一项关于如何分配这些宝藏的协议：作为领袖的 **Askeladd** 将取走前面的 k 个宝藏，即价值为 a_1, a_2, \dots, a_k 的宝藏，而其余海盗将分配其余的宝藏。但是，由于时间太晚，他们决定在第二天早上进行这个分配。

Askeladd 非常聪明且狡猾，他在夜间偷偷摸摸地试图重新安排宝藏，以便他可以获得更多的宝藏总价值。然而，由于某种原因，他只被允许交换两个相邻的宝藏。此外，考虑到被其他海盗发现的风险，每次交换将花费 **Askeladd** c 的价值。**Askeladd** 可以执行任意数量（可能为零）的交换。

Askeladd 想知道他可以获得的最大利润是多少（即他可以获得的所有宝藏总价值减去交换所消耗的总代价）。

输入格式

第一行包含三个整数 n, k 和 c ($0 \leq k \leq n \leq 3 \times 10^5, 0 \leq c \leq 10^9$)，表示宝藏的总数量，**Askeladd** 取走的宝藏数量，以及交换相邻两个宝藏的成本。

接下来一行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，表示宝藏的价值。

输出格式

在一行中输出一个整数，表示 **Askeladd** 能获得的最大利润。

样例输入 1

样例输出 1

3 2 1 1 3 5	6
----------------	---

样例输入 2

样例输出 2

3 2 2 1 3 5	4
----------------	---

样例输入 3

样例输出 3

4 2 1 2 3 5 6	7
------------------	---

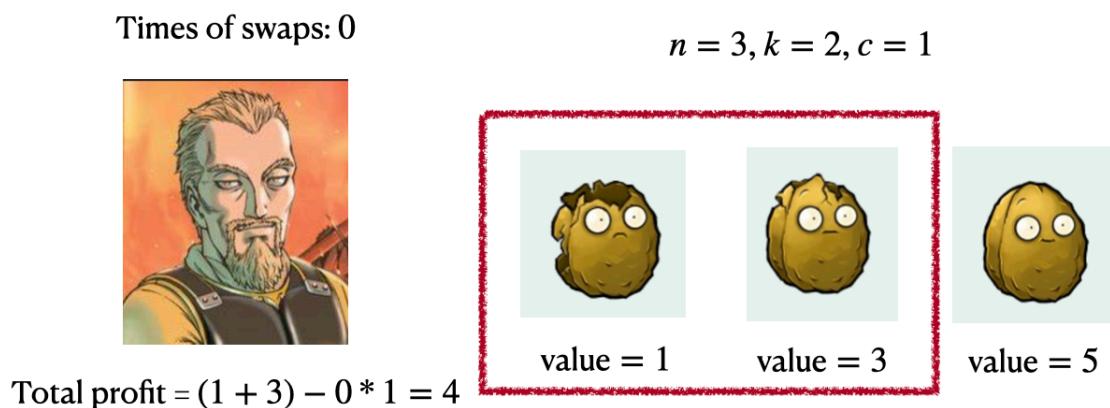
样例输入 4

7 3 2 2 1 3 6 10 9 7	10
-------------------------	----

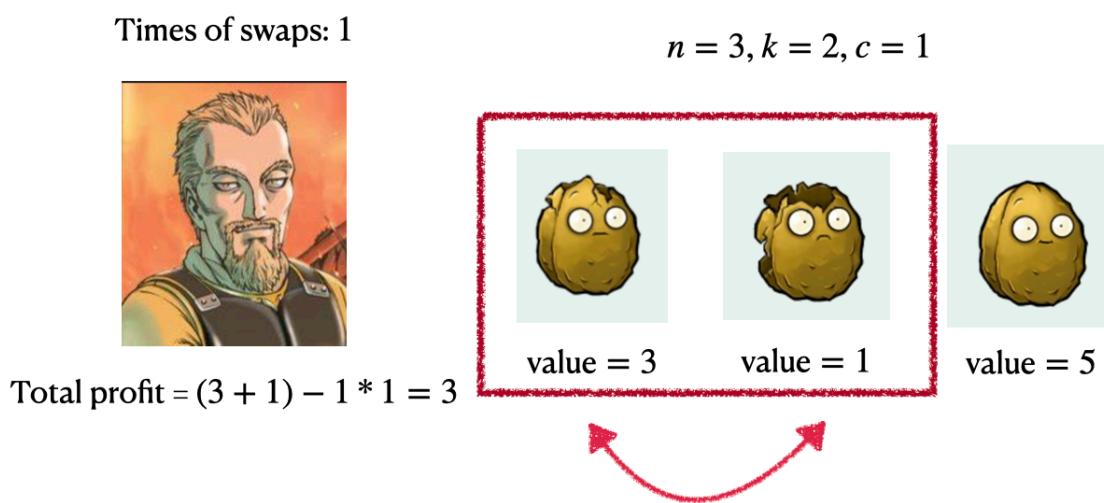
样例输出 4

样例解释

为了帮助理解，我们提供了第一个样例的图示说明。当 Askeladd 不进行任何交换时，他获得的总利润等于 4。



Askeladd 的最佳方法是先交换第一个和第二个宝藏，



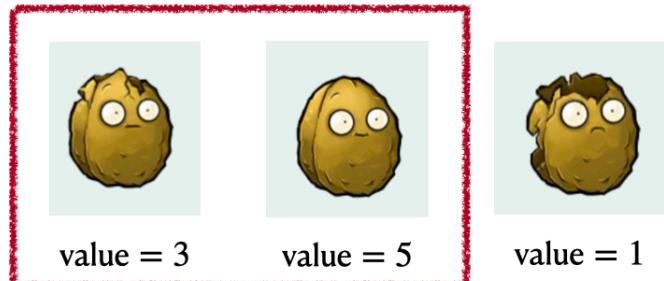
然后交换第二和第三个宝藏，总利润为 6。

Times of swaps: 2

$$n = 3, k = 2, c = 1$$



$$\text{Total profit} = (3 + 5) - 2 * 1 = 6$$

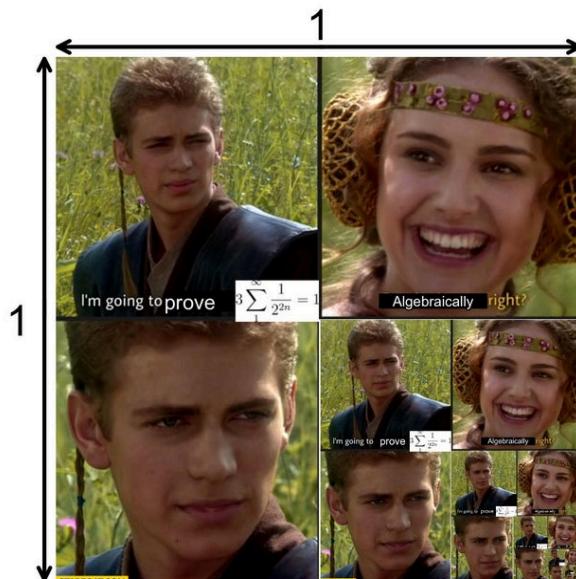


F. Merge the Squares!

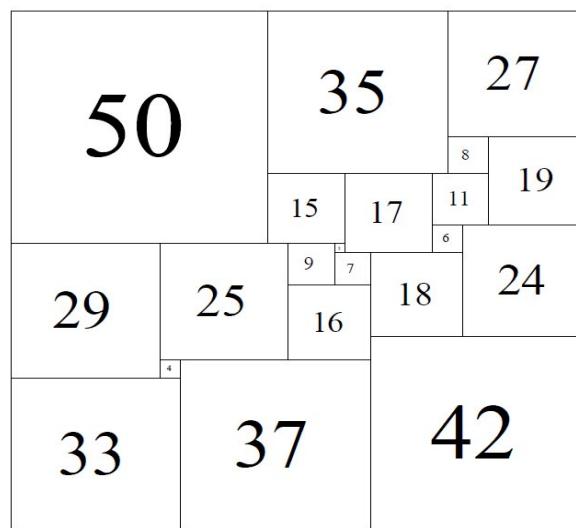
时间限制: 2 seconds

空间限制: 2048 megabytes

就像所有的热爱数学的学生一样, Bobo 很喜欢正方形, 更喜欢大正方形, 最喜欢的是把许多小正方形合并成一个大正方形。“没错, 这就是我在代数卷子上画正方形证明的原因”, Bobo 向代数学的教授解释道。



正因为这样, Bobo 对完美正方形也很感兴趣。完美正方形指在一正方形内切割出大小都相异的小正方形。完美正方形的最小阶数为 21 阶, 由 A. J. W. Duijvestijn 发现。



21 : 112A AJD 1978

然而, 把一些大小相异的小正方形拼成一个大正方形对 Bobo 来说太难了, 他想先做一点简单的, 比如把排成 n 行 n 列的 $n \times n$ 个边长为 1 的小正方形拼成一个边长为 n 的大正方形。

当然, Bobo 可不能只用一次操作就把所有正方拼在一起, 那样太无趣了。他现在有了一个新的要求: 每次合并的正方形数量必须要在 2 到 50 之间 (包含 2 和 50), 并且每次合并完之后的形状仍然要是一个正方形。然后 Bobo 并不知道应该如何操作, 他把这个难题交给了你。

输入格式

在一行中输入一个正整数 n ($1 \leq n \leq 1000$), 表示小正方形的行数和列数。

输出格式

在第一行中输出一个数字 m ($0 \leq m \leq 10^6$) 代表操作次数。接下来 m 行, 每行输出三个数字 x, y, k 表示一次操作。其中 $1 \leq x \leq n$ 和 $1 \leq y \leq n$ 代表这次操作后合并出的大正方形左上角所在的行号以及列号 (行号和列号用 $1 - n$ 表示), $1 \leq k \leq n$ 代表合并出的大正方形的边长。你需要保证你每次操作都满足以下条件:

- $x + k \leq n, y + k \leq n$
- 当前以第 x 行第 y 列的格子为左上角的边长为 k 的正方形内包含的都是完整的正方形, 且正方形的个数在 2 到 50 之间。

你还需要保证在所有操作做完之后, 所有 $n \times n$ 个边长为 1 的小正方形会拼成一个边长为 n 的大正方形。可以看样例解释以获得更多信息。

样例输入 1

1	0
---	---

样例输出 1

样例输入 2

2	1 1 1 2
---	------------

样例输出 2

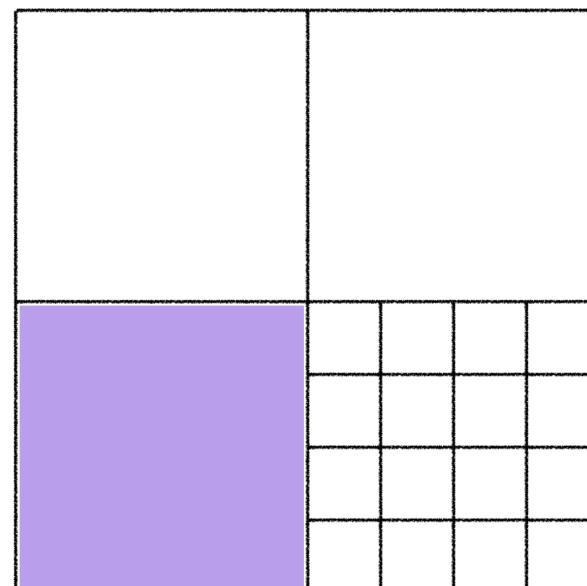
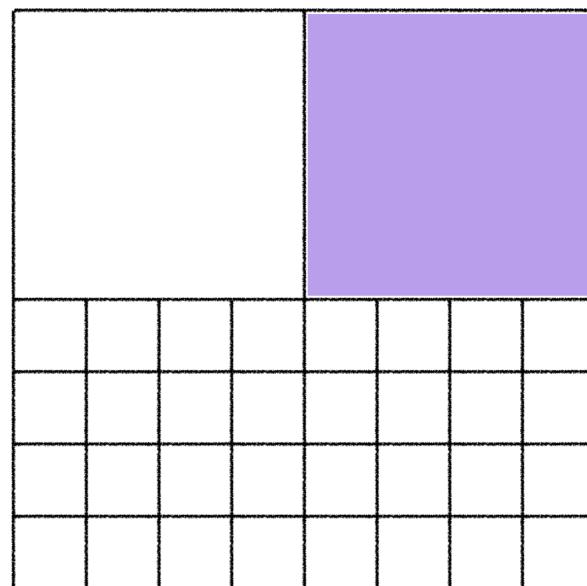
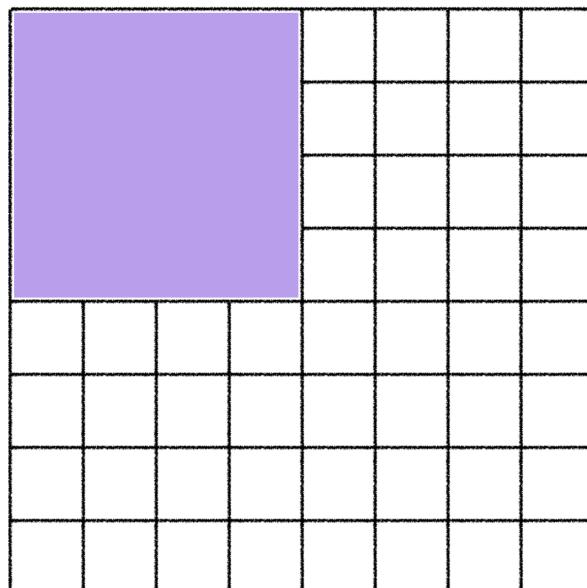
样例输入 3

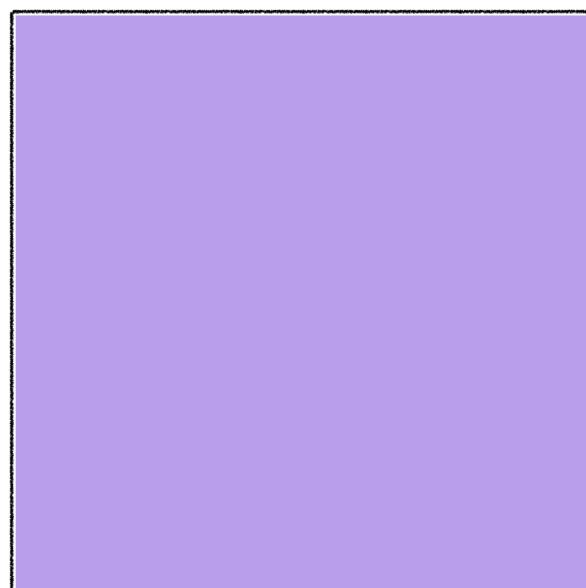
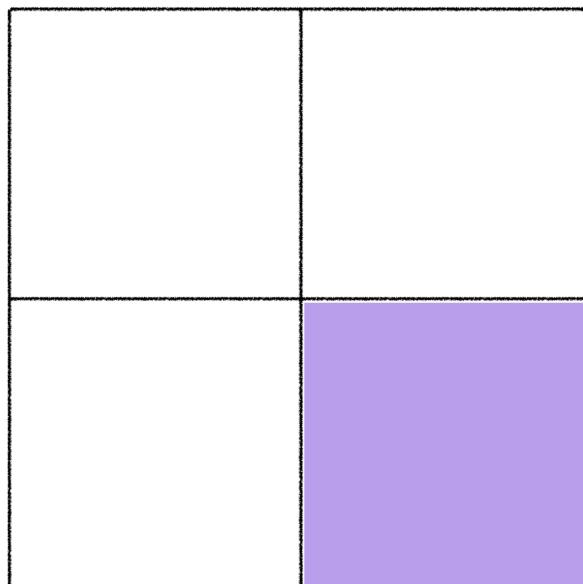
8	5 1 1 4 1 5 4 5 1 4 5 5 4 1 1 8
---	--

样例输出 3

样例解释

为了帮助理解, 我们提供了第三个样例的图示说明。第三个样例解答中的合并过程如以下所示 (紫色代表每次合并的区域)。





G. Out of Context

时间限制: 1 seconds

空间限制: 2048 megabytes

Bobo 作为一名记者，被要求写一篇有关 **Boboland** 的长度为 n 报道。**Bobo** 本想既不吝惜溢美之词，又带点批判的眼光去写这篇报道，但他害怕被有心之人断章取义说这是一篇抨击 **Boboland** 的文章。一篇长度为 n 的报道可以看做是一个长度为 n 的整数序列 a_1, \dots, a_n ，其中 $-m \leq a_i \leq m$ 代表一个赞美度为 a_i 的语句。一旦在报道中出现连续的 k 个语句使得它们总赞美度小于 0 且满足 $k \geq 2$ ，那么这连续的 k 个语句就有可能被有心之人断章取义地截取下来反过来攻击 **Bobo**. **Bobo** 不希望这样的事情发生，他想知道，在不被断章取义的情况下，他能够写出多少篇不同的文章？

正式地说，你需要计算有多少个长度为 n 的整数序列 a_1, a_2, \dots, a_n 满足：

- 对于所有的 $1 \leq i \leq n$, 都有 $-m \leq a_i \leq m$
- 对于所有的 $1 \leq l < r \leq n$, 都有 $\sum_{i=l}^r a_i \geq 0$ 。

由于答案可能过大，你需要将答案对 998244353(一个质数) 取模。

输入格式

在第一行中输入两个正整数 n, m ($1 \leq n, m \leq 5000$)，分别代表报道的长度以及报道的语句的范围。

输出格式

在一行中输出一个正整数，表示答案。

样例输入 1	样例输出 1
3 3	130

样例输入 2	样例输出 2
5 1	64

样例解释

为了帮助理解，我们提供了第一个样例测试用例的图示说明。我们用以下七个 emoji 表情从左到右依次表示文章里的赞美度从 3 到 -3 递减的语句。



以下两个文章就满足不被断章取义的条件，因为任意包含超过一个语句的语句连续段赞美度之和都至少为 0。



以下两个文章就会被断章取义，且可能被断章取义的语句连续段已经用红框标出。



H. Portal 3

时间限制: 3 seconds

空间限制: 2048 megabytes

发生了什么? 你在哪里? 你惊醒过来, 发现自己处于一个奇怪但熟悉的地方。你四处张望, 墙上的标志立刻提醒你, 你现在在“光圈科技丰富学习中心”。

嗨, *Chell*, 很高兴再次见到你。这是 GLaDOS 的电子音。她似乎很兴奋地看到你, 你可能会想知道为什么你又来到这里。事实是, 在一些失败的试验之后, 我终于意识到, 你是唯一伟大的测试对象! 光圈科技需要你! 你被她的突然爆发吓到了。但她随后又用柔和的声音说, 别害怕, 你只需要在一些房间中完成一些测试, 但这次你需要尽快完成。当所有测试完成后, 你可以获得一个蛋糕作为奖励。



在光圈科技丰富学习中心的 GLaDOS

你不知道是否可以相信她, 但你只能尽快完成所有测试。光圈科技丰富学习中心有 n 个不同的房间, 编号为 1 到 n 。由于你不在的这几年里的光圈科技的飞速发展, 现在你可以直接从任何房间 i 到房间 j ($1 \leq i, j \leq n$), 但要花费 a_{ij} 的时间。你可以假设对于所有 $1 \leq i \leq n$, 都有 $a_{i,i} = 0$, 对于所有满足 $i \neq j$ 的 $1 \leq i, j \leq n$, 都有 $a_{i,j} \geq 0$ 。此外, 由于在光圈科技计算机丰富学习中心存在的单向电梯、凝胶、同伴方块和机器人, 可能会出现对于 $i \neq j$ 有 $a_{i,j} \neq a_{j,i}$ 的情况。GLaDOS 交给了你一个每个测试所在房间的列表 v_1, v_2, \dots, v_k ($k \geq 2, 1 \leq v_i \leq n$)。你需要按顺序访问列表上的房间, 并尽快完成所有测试。你现在在房间 v_1 中, 准备随时参加第一次测试。由于你对所有测试都非常熟悉, 因此可以忽略完成每个测试所需的时间, 你只需要尽量减少访问房间的总时间成本, 这听起来很容易。

哦, 我忘了提醒你, 为了让这些测试稍微轻松一点但又不至于过于简单, 你可以看看你身后。你转过身来, 发现盒子里有一个传送门枪, 但盒子上写着“限量使用: 在开始测试前只能使用一次”。经过近距离的观察, 这个传送门枪不是通过射击来创建传送门, 而是通过输入房间号码来创建传送门。旁边放着一份指南, 上面说在尝试完成所有测试之前, 你可以输入两个数字 i, j , 满足 $1 \leq i \leq j \leq n$, 然后将创建一个双向传送门, 使你可以在不用花费时间的情况下直接从一个房间移动到另一个房间, 这意味着 $a_{i,j} = a_{j,i} = 0$ 。你必须在尝试完成所有测试之前选择 i 和 j , 而且不能更改。因此, 你需要仔细选择两个房间以打开传送门, 然后尽快完成所有测试。你现在想知道的问题是, 你最少需要多少时间才能完成所有测试?

输入格式

第一行包含两个整数 n, k ($2 \leq n \leq 500, 2 \leq k \leq 10^6$), 表示光圈科技丰富学习中心中房间数量和房间列表的长度。

接下来有 n 行, 第 i 行 ($1 \leq i \leq n$) 包含 n 个整数 $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($0 \leq a_{i,j} \leq 10^9$), 描述从一个房间直接移动到另一个房间所需的时间。

最后一行包含 k 个整数 v_1, v_2, \dots, v_k ($1 \leq v_i \leq n$), 描述你需要按顺序访问的房间序列。

输出格式

在一行中输出一个整数, 表示完成所有测试所需的最短时间。

样例输入 1

3 4
0 4 2
3 0 6
5 1 0
1 2 3 1

样例输出 1

3

样例输入 2

4 10
0 1000000000 1000000000 1000000000
1000000000 0 1000000000 1000000000
1000000000 1000000000 0 1000000000
1000000000 1000000000 1000000000 0
4 3 2 1 3 2 4 1 2 3

样例输出 2

6000000000

样例输入 3

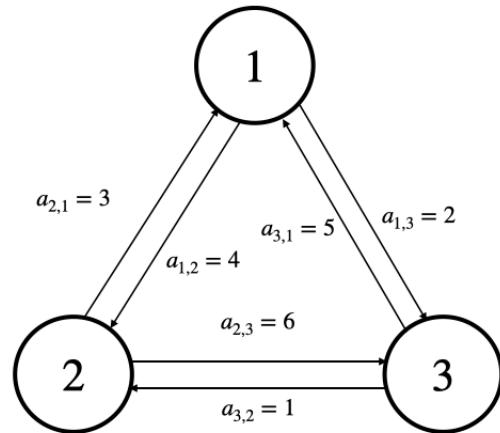
2 2
0 1
2 0
2 1

样例输出 3

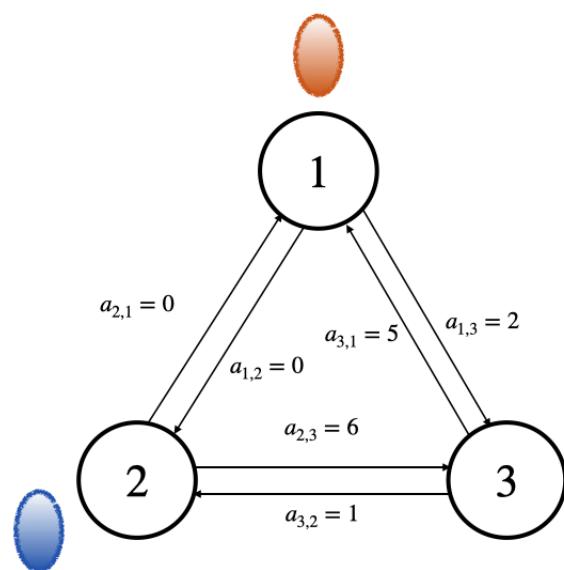
0

样例解释

为了帮助理解, 我们提供了第一个样例的图示说明。表示光圈科技丰富学习中心的图如下所示。

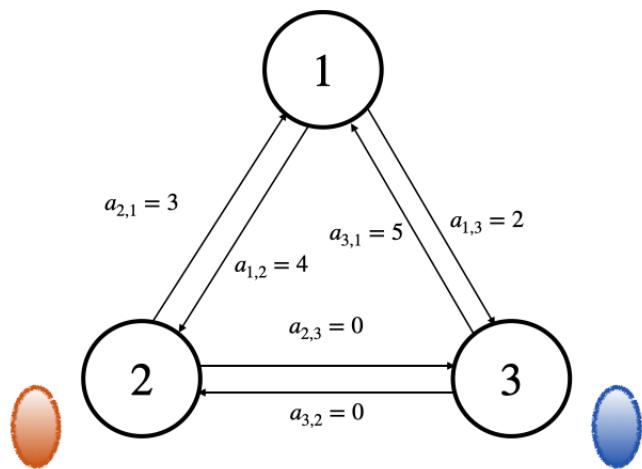


如果我们在房间 1 和 2 之间打开一个传送门，图将变成以下形式：

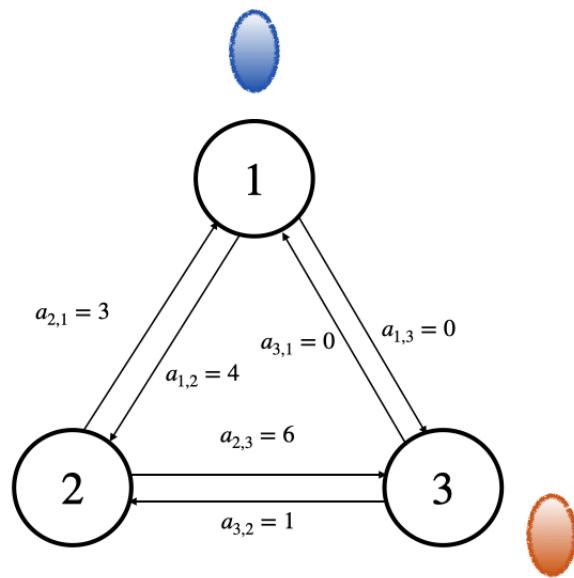


此时，完成所有测试的最佳路线如下：**1 → 2 → 1 → 3 → 2 → 1**，总时间为 $0+0+2+1+0 = 3$ 。这里粗体数字对应于完成测试的房间。

如果我们在房间 2 和 3 之间打开一个传送门，图将变成以下形式：



此时，完成所有测试的最佳路线如下：**1** → 3 → **2** → **3** → 2 → **1**，总时间为 $2+0+0+0+3=5$ 。
如果我们在房间 1 和 3 之间打开一个传送门，图将变成以下形式：



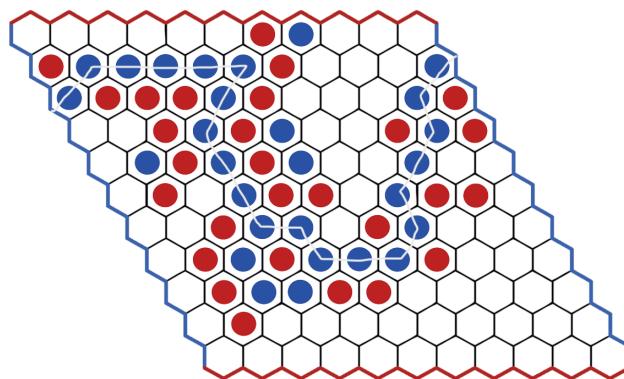
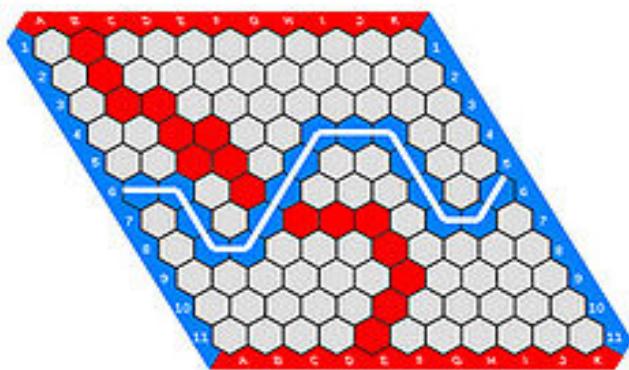
此时，完成所有测试的最佳路线如下：**1** → 3 → **2** → 1 → **3** → **1**，总时间为 $0+1+3+0+0=4$ 。
因此，最优的方案是在房间 1 和 2 之间打开一个传送门，完成所有测试所需的最短时间为 3。

I. Square Game

时间限制: 1 seconds

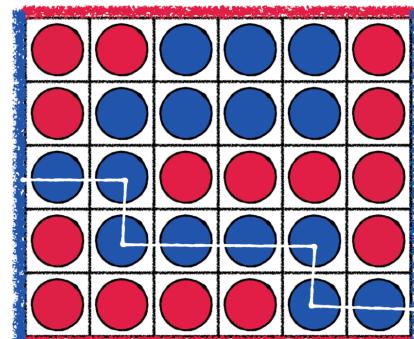
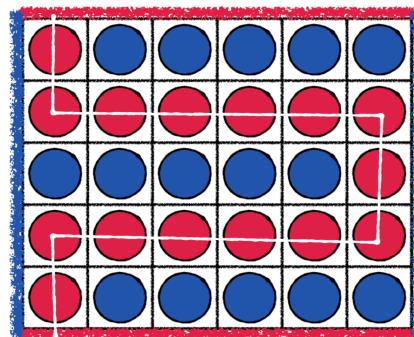
空间限制: 2048 megabytes

你听说过 **Hex** 游戏吗? **Hex** 游戏是一种在六边形棋盘上玩的双人游戏, 游戏规则是红/蓝双方轮流将自己颜色的棋子放置在棋盘的空格子上, 直到红方棋子连接棋盘的上下两边/蓝方棋子连接棋盘的左右两边为止, 达成连接目标的一方获得游戏胜利。连接的条件是存在一条己方棋子颜色的路径使得路径上连续两个格子的小六边形都有至少一条公共边。例如以下两个 **Hex** 棋局都是蓝方获得胜利。

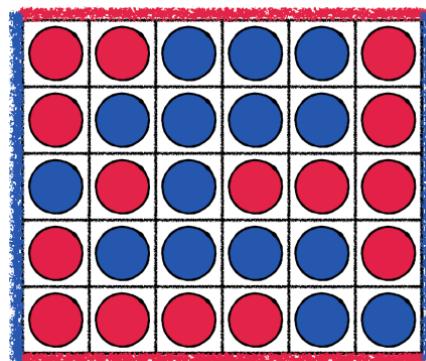
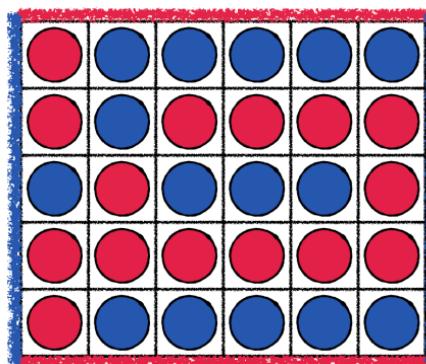


不难证明, **Hex** 游戏一定有且仅有一个获胜方。

Bobo 也想玩 **Hex** 游戏, 但他觉得棋盘里每个格子是六边形的太麻烦了, 于是他发明了如下的 **Square** 游戏! **Square** 游戏的规则和 **Hex** 游戏规则类似, 也是红/蓝双方轮流将自己颜色的棋子放置在棋盘的空格子上, 直到红方棋子连接棋盘的上下两边/蓝方棋子连接棋盘的左右两边为止, 达成连接目标的一方获得游戏胜利。只不过在 **Square** 游戏中, 棋盘变成了由许多小正方形组成的矩形棋盘, 连通的条件仍然是存在一条己方棋子颜色的路径使得路径上连续两个格子的小六边形都有至少一条公共边(即四连通)。如下是两个在 5×6 的棋盘上的 **Square** 游戏棋局, 结果分别是红方胜和蓝方胜。



Bobo 十分兴奋，想去找他的朋友 oboB 分享这个 Square 游戏，但 oboB 听完 Bobo 说完规则后立刻就发现了问题：Bobo 发明的这款 Square 游戏中可能出现平局！例如以下两个在 5×6 的棋盘上的 Square 游戏棋局结果都是平局，因为双方都无棋可下了，而且红方棋子并没有连通棋盘的上下两边，蓝方棋子也没有连通棋盘的左右两边。



“可是…… 平局的棋局应该不多吧？” Bobo 还在嘴硬。为了反驳 Bobo, oboB 随手就画出了一个 n 行 m 列由红色格子和蓝色格子构成的矩阵，“我从里面随机挑出一个子矩阵作为棋盘，都有不小的概率都是一个平局的棋局！” Bobo 看了一下似乎确实是这样，但他突然又想到了

什么，“为了使 **Square** 游戏的规则成立，选出的子矩阵上下必须全是红色的，左右必须全是蓝色的！(对应示例图中矩阵上下和左右红/蓝色的线)”。**oboB** 实在是拿这个嘴硬的 **Bobo** 没办法了，“那你数一数，有多少种这样的棋局？”

正式地说，给定一个 n 行 m 列的矩阵 A ，其中矩阵中的每个格子要么为红色（用 R 表示）要么为蓝色（用 B 表示）。我们用 $A_{i,j}$ 表示矩阵第 i 行第 j 列的格子颜色，你需要计算有多少对四元组 (x_1, x_2, y_1, y_2) 满足以下条件 ((x_1, y_1) 和 (x_2, y_2) 分别是你选取的子矩阵的左上角和右下角的坐标)：

- $2 \leq x_1 \leq x_2 \leq n - 1, 2 \leq y_1 \leq y_2 \leq m - 1$ (选取的必须要是子矩阵)
- $\forall y_1 \leq j \leq y_2, A_{x_1-1,j} = A_{x_2+1,j} = R$ (子矩阵上下必须都是红色的)
- $\forall x_1 \leq i \leq x_2, A_{i,y_1-1} = A_{i,y_2+1} = B$ (子矩阵左右必须都是蓝色的)
- 不存在 $P = ((i_1, j_1), (i_2, j_2), \dots, (i_\ell, j_\ell))$ 满足：(不存在连通子矩阵上下的红色路径)
 - $\ell \geq 1, i_1 = x_1, i_\ell = x_2$
 - $\forall 1 \leq k \leq \ell, x_1 \leq i_k \leq x_2 \wedge y_1 \leq j_k \leq y_2$
 - $\forall 1 \leq k \leq \ell, A_{i_k,j_k} = R$
 - $\forall 1 \leq k \leq \ell - 1, |i_k - i_{k+1}| + |j_k - j_{k+1}| = 1$
- 不存在 $P = ((i_1, j_1), (i_2, j_2), \dots, (i_\ell, j_\ell))$ 满足：(不存在连通子矩阵左右的蓝色路径)
 - $\ell \geq 1, j_1 = y_1, j_\ell = y_2$
 - $\forall 1 \leq k \leq \ell, x_1 \leq i_k \leq x_2 \wedge y_1 \leq j_k \leq y_2$
 - $\forall 1 \leq k \leq \ell, A_{i_k,j_k} = B$
 - $\forall 1 \leq k \leq \ell - 1, |i_k - i_{k+1}| + |j_k - j_{k+1}| = 1$

输入格式

在第一行中输入两个数字 n, m ($1 \leq n, m \leq 300$)，分别表示矩阵的行数和列数。接下来 n 行，第 i ($1 \leq i \leq n$) 行包含一个长度为 m 的仅由 R 和 B 组成的字符串，其中第 i 行中字符串的第 j ($1 \leq j \leq m$) 个字符为 R 代表矩阵的第 i 行第 j 列的格子颜色是红色，为 B 代表矩阵的第 i 行第 j 列的格子颜色是蓝色。

输出格式

在一行中输出一个整数，表示有多少子矩阵构成一个平局的 **Square** 游戏棋局（正式的定义已经在题面中给出）。

样例输入 1

```
6 6
RRRRRB
BRRRBB
BBRBBC
BBBBRB
BBRBBB
BRBBBB
BRRRRR
```

样例输出 1

```
2
```

样例输入 2

5 5
BRRRB
BBRRB
BRBRB
BRRBB
BRRRB

样例输出 2

1

样例输入 3

7 7
BRRRRRB
BRBBBBR
BBRBRBB
BRRRRRB
BBRBRBB
BRBBBBR
BRRRRRB

样例输出 3

7

样例输入 4

3 3
RRR
BRB
RRR

样例输出 4

0

样例输入 5

20 20
 RRRRRRRRRRRRRRRRRRR
 BRRRRRRRRRRRRRRRRRB
 BBBBRRBBBRRRRRRRB
 BRRRBRRBRRBRRRRRB
 BRRRRRRRRRBRRRRRB
 BBBB BBBB BRRRRRB
 BRRRRRRRRRRRRRB
 BRRRRRRRRRRRRRB
 BBBB BBBB BRRRRRB
 BRRRRRRRBRRBRRRB
 BRRRRRRRBRRBRRRB
 BBBB BBBB RRRRRRB
 BRRRRRRRBRRRBRBBB
 BRRRRRRRRRRRBRRB
 BBBB BBBB RRRRRRB
 BRRRRRBRRR RRRRB
 BRRRRRBRRR RRRRB
 BBBB BBBB RRRRRRB
 BRRRRRRRRRRRRRB
 RRRRRRRRRRRRRRRR

样例输出 5

0

样例输入 6

20 20
 RRRRRRRRRRRRRRRRR
 BBBRBRBBRBBBBRBRB
 BBRBRBRBBB RBRB
 BBRBBRBRBBB RBRB
 BBRBBRBRBBRBBRBB
 BBRBRBRBBRBBRBBB
 BBRBBRBRBBRBBRBBB
 BBRBBRBRBBRBBB BBBB
 BBRRBRBBRBBRBBB BBBB
 BBBRBBB RBBRBBB BBBB
 BBBB BBBB RBBB BBBB
 RRRRRRRRRRRRRRRR

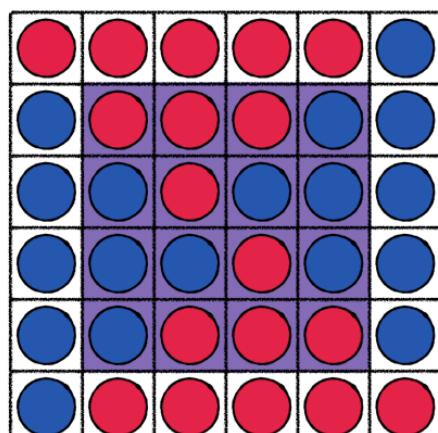
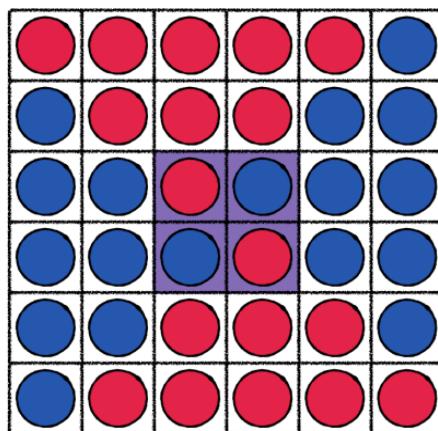
样例输出 6

5

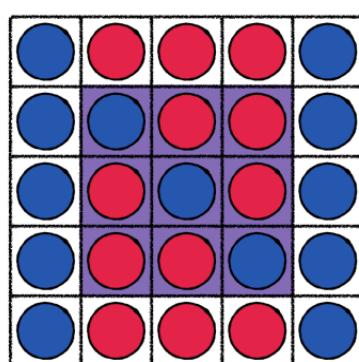
样例解释

注意本题除了这里给出的六组样例外还有一组 $n = m = 25$ 的样例数据，在评测网站中给出，可以自行下载。

为了帮助理解，我们提供了前两个样例的图示说明。第一个样例中的两个合法子矩阵如以下所示。



第二个样例中的一个合法子矩阵如以下所示。



J. We are the Lights

时间限制: 1 second

空间限制: 2048 megabytes

一天 Bobo 正在参加国际灯光展，他注意到了角落里有 $n \times m$ 盏灯，排成 n 行 m 列，初始时所有灯都是灭的。

Bobo 注意到了每行和每列旁边都有开/关两种按钮，这意味着他可以进行以下 4 种操作：

1. row i on: 将第 i 行的所有灯打开。
2. row i off: 将第 i 行的所有灯关闭。
3. column i on: 将第 i 列的所有灯打开。
4. column i off: 将第 i 列的所有灯关闭。

调皮的 Bobo 当然不会放过这个机会。给出 q 个的操作，你需要求出 Bobo 在按顺序执行这 q 次操作后有多少盏灯是打开的状态。

输入格式

第一行输入三个正整数 n, m, q ($1 \leq n, m \leq 1000, 1 \leq q \leq 10000$)，分别表示灯的行数，列数，以及 Bobo 做的操作的个数。

接下来 q 行，每行表示一个操作，操作的输入方式已在题目描述中给出。

输出格式

在一行中输出一个数字，表示按顺序进行 q 次操作后处于打开状态的灯的数量。

样例输入 1

```
2 2 3
row 1 on
column 2 off
row 2 on
```

样例输出 1

```
3
```

样例输入 2

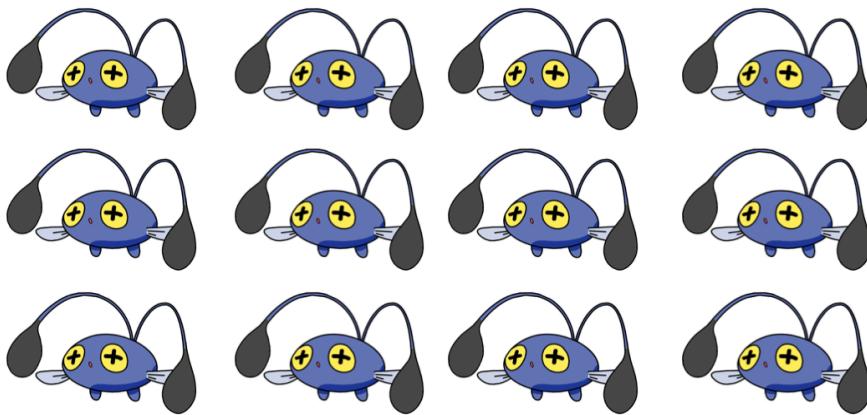
```
3 4 4
row 1 on
column 3 on
row 3 on
column 2 off
```

样例输出 2

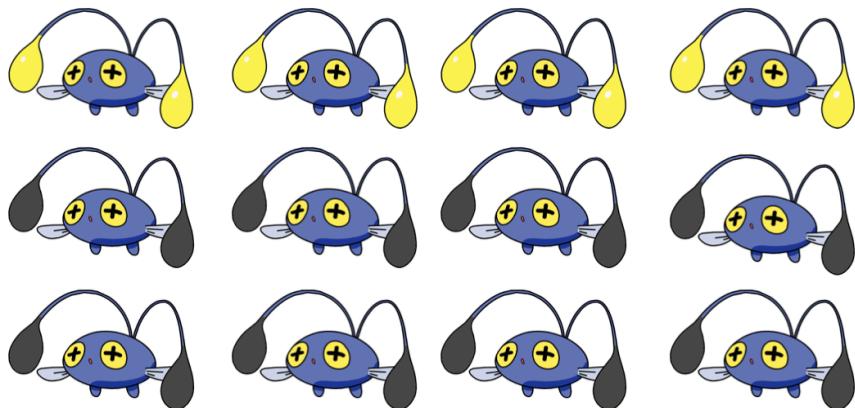
```
7
```

样例解释

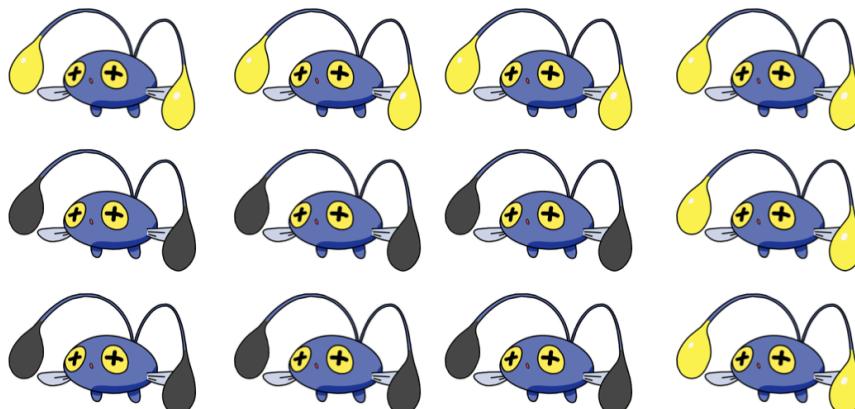
为了帮助理解，我们提供了第二个样例测试用例的图示说明。灯总共有三行，每行四个。初始时所有灯都是关上的，如下图所示。



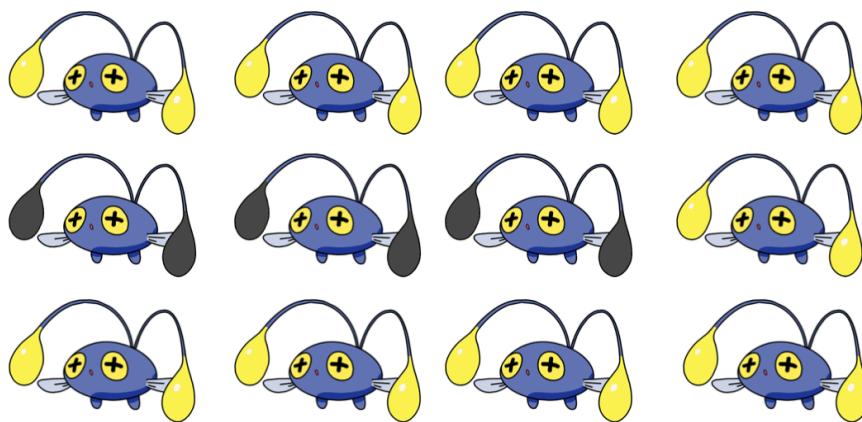
在进行了第一次操作后，灯的开关情况如下图所示。



在进行了第二次操作后，灯的开关情况如下图所示。



在进行了第三次操作后，灯的开关情况如下图所示。



在进行了第四次也就是最后一次操作后，灯的开关情况如下图所示。此时还剩下 7 盏灯亮着。

