

Problem 1-1. 爱水群的小明

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 5 points

小明很喜欢水群，他常常能看到群里的大佬说着各种他听不懂的话，他很羡慕，也很焦虑，他默默想着：“如果我也想像他们一样强就好了。”

请你输出 "我也想像他们一样强"

Example

standard input	standard output
	"我也想像他们一样强"

Problem 1-2. 波动的实力

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 5 points

小李是一个游戏狂热者，但他发现，他每天打游戏的状态会导致自己实力的浮动，没手感、没吃早饭、没热身等等“情况”都会导致小李的实力值波动，所以他选择了 10 天的实力值进行研究，请你告诉他这 10 天里，实力值的最大值、最小值、平均值

Input

一行 10 个整数，分别为 $b_1, b_2, b_3, \dots, b_{10}$ ($-10^9 \leq b_i \leq 10^9$)

Output

一行三个整数 max, min, avg ，分别代表实力值的最大值、最小值、平均值

注意：平均值要输出向 0 取整的值，即直接舍弃小数位的值，例如 $\frac{11}{3} = 3$ ， $-\frac{11}{3} = -3$

Example

standard input	standard output
1 2 3 4 5 6 7 8 9 10	10 1 5

Problem 1-3. 汗流浹背了吧, 老弟

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 10 points

在 1974 年, 第一次东南亚自由搏击, 就得了冠军。1980 年, 打赢了日本重炮手雷龙。接着连续三年, 打败所有日本空手道高手, 赢得全日本自由搏击冠军, 绰号魔鬼筋肉人。

接下来你需要操控我们的魔鬼筋肉人东哥选手和小日子过的不错的日本选手进行决斗, 不过请不要害怕, 毕竟你操控的角色可是摇摆拳法创始人东哥, 而且你还获得了小日子选手的出招表。

形式化来讲, 我们会获得一个仅包含 L 与 R 的字符串出招表, 代表着小日子选手的本次攻击的方向, 你需要操控角色来躲避敌人的所有攻击, 即最后输出你的出招表 (俗话说先礼后兵, 所以我们总是让小日子选手先进行攻击)。

请在躲闪完所有攻击后另起一行输出 汗流浹背了吧, 老弟~。



Input

第一行读入一个整数 n ($1 \leq n \leq 5 * 10^4$)

接下来一行读入一个长度为 n 的字符串 s ($\forall s_i \in \{'L', 'R'\}$)

Output

第一行输出你操控角色的出招表, 接下来一行输出 汗流浹背了吧, 老弟~。

Example

standard input	standard output
5	RRLLR
LLRRL	汗流浹背了吧, 老弟~

Problem 1-4. 小 C 爱答题 (Easy Version)

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 10 points

本题目与 Hard Version 的区别仅为 n, q, a_i 的数据范围不同

小 C 天天上课睡觉，不喜欢学习，热心的你不忍心让小 C 这样堕落下去，依老师上课给小 C 留下了一个问题。但小 C 实在是看不懂，准备摆烂了，你能教教他吗？

具体问题如下：

给你一个长度为 n 的数组 a ，数组下标从 1 开始，你需要回答 q 次询问，每次询问会给你两个整数 l, r ，你的任务是求出 $a_l \sim a_r$ 的区间和 $(a_l + a_{l+1} + \dots + a_r)$ 和区间异或和 $(a_l \oplus a_{l+1} \oplus \dots \oplus a_r)$ ，并输出两个值异或起来的值。

提示：

\oplus 是异或符号，在 C / C++ 中，求 a 与 b 的异或值的方法是 $a \wedge b$ 。

异或运算规则如下：

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

两个整数进行异或运算，指的是它们转成二进制后（位数不同的在前面补 0），每一位相互异或的结果再转成十进制，如二进制下 $(01)_2 \oplus (10)_2 = (11)_2$ ，即 $1 \oplus 2 = 3$ 。

异或是一种重要的运算，它具有如下性质：

归零性： $a \oplus a = 0$ ；

交换律： $a \oplus b = b \oplus a$ ；

结合律： $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ ；

特别的，对于任意一个整数 a ， $a \oplus 0 = a$ 。

Input

第一行输入 2 个整数 n, q ($1 \leq n \leq 1000, 1 \leq q \leq 1000$)

接下来一行输入 n 个整数 $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_i \leq 1000$)

接下来有 q 行，每行读入两个整数 l, r ($1 \leq l \leq r \leq n$)

Output

输出 q 行，每行输出 $a_l \sim a_r$ 的区间和与区间异或和的异或值。

Example

standard input	standard output
5 5	6
1 2 3 4 5	12
1 3	14
2 4	14
3 5	14
1 4	
1 5	

Problem 1-5. 凯撒密码

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 15 points

凯撒密码（英语：Caesar cipher），是一种最简单且最广为人知的加密技术。凯撒密码是一种替换加密技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例如，当偏移量是 3 的时候，所有的字母 A 将被替换成 D，B 变成 E，以此类推。这个加密方法是以罗马共和时期凯撒的名字命名的，据称当年凯撒曾用此方法与其将军们进行联系。

凯撒密码的替换方法是通过排列明文和密文字母表，密文字母表示通过将明文字母表向左或向右移动一个固定数目的位置。例如，当偏移量是左移 3 的时候（解密时的密钥就是 3）：

明文字母表：ABCDEFGHIJKLMNOPQRSTUVWXYZ

密文字母表：DEFGHIJKLMNOPQRSTUVWXYZABC

使用时，加密者查找明文字母表中需要加密的消息中的每一个字母所在位置，并且写下密文字母表中对应的字母。需要解密的人则根据事先已知的密钥反过来操作，得到原来的明文。例如：

明文：THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

密文：WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

简单来说，当偏移量为 k 时，则将原文中的每一个字母用其后的第 k 个字母来替代，字母 Z 后面的字母为 A，以此循环。

如果明文内包含空格，则密文相应位置也为空格，不做改动。

Input

本题为多实例，先输入一行一个正整数 T ($1 \leq T \leq 100$)，表示查询的个数。

对于每次查询，先输入一行两个整数 n, k ($1 \leq n \leq 10^5, -1000 \leq k \leq 1000$)，分别表示字符串的长度和凯撒密码的偏移量。

然后输入一个长度为 n 的字符串 s ，表示需要加密的明文，明文中只包含大写字母及空格，且行首行末不包含空格。

保证所有样例中 n 的和不超过 10^5 。

Output

输出 T 行，每行一个字符串 t ，表示加密后的密文。

Example

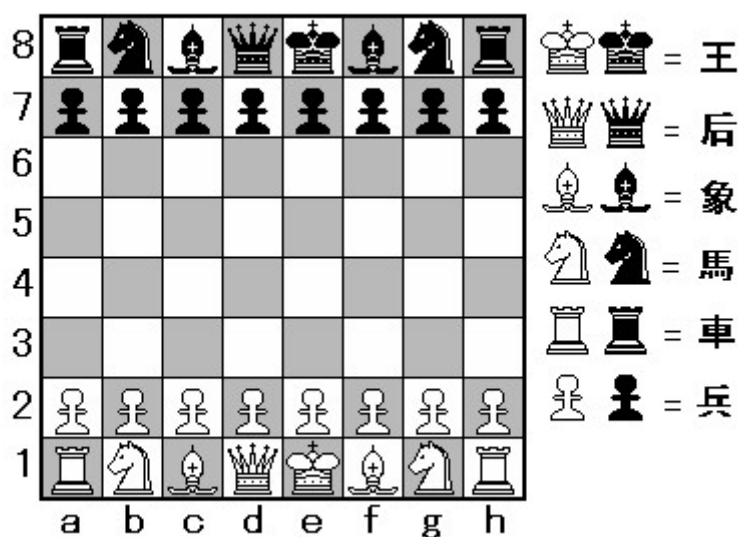
standard input
1
43 3
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
standard output
WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

Problem 1-6. 国际象棋

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes
Score: 15 points

国际象棋（英语：Chess），又叫西洋棋，是一种盛行于国际范围的二人对弈的战术棋盘游戏，是目前世界上最流行的游戏之一，中文为区别于中国象棋而冠以“国际”二字。棋盘为正方形，纵横各八格，共六十四方格，黑白（深色与浅色）相间；棋子分黑白（深浅）两色，共三十二枚，每方各有一王、一后、双车、双象、双马和八兵。不同棋子走法不同，以把对方的王“将死”为胜。

国际象棋由黑白两种棋子组成。无论是白方视野还是黑方视野，棋盘的最右下角的一格一定要是白色，白后一定要放在白格上，黑后一定要放在黑格上。白王一定在 e1 格，黑王一定在 e8 格。



Alice 和 Bob 想玩国际象棋了，但是他们还没有一个棋盘，作为热心的好同学，请你帮他们画一个边长为 n 的正方形棋盘吧。

Input

一行一个正整数 n ($1 \leq n \leq 500$)

Output

一个边长为 n 的正方形棋盘，每个小格由 2×4 的字符组成，其中 . 代表白格，* 代表黑格，# 代表边框（如样例所示）

Example

standard input	standard output
4	##### #...##### #...##### ##### ##### ##### ##### ##### ##### ##### ##### ##### ##### ##### ##### #####

Note

对于 java 与 python 的提交，请使用较快的输出方式，以免因为输出量过大而超时

Problem 1-7. CTF

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 20 points

夺旗（英语：Capture the flag，简称 CTF）在计算机安全中是一种活动，当中会将“旗子”秘密地埋藏于有目的的易受攻击的程序或网站。参赛者从其他参赛者或主办方偷去旗子。

非常崇拜探姬的小学妹最近迷上了 CTF，学长给了她一个压缩包，让她找到其中的 flag 并提交。

她今天终于破解了压缩包的密码，但解压后她却发现，里面竟然有足足 100 个 txt 文件！她使用 Ctrl+F 大法搜索过后，发现了这些 txt 文件里面有着相同的规律：

每个 txt 文件中都藏着一个 flag，这些 flag 都有固定的格式，即 `flag + 一个两位十进制数 + 真正的 flag 内容`，其中这个两位十进制数代表真正的 flag 内容的长度。

例如，在 `flag03ctf` 中，flag 后的两位十进制数 03 表示真正的 flag 内容的长度为 3 个字符，此时真正的 flag 内容就是 `ctf`，所以你需要提交 `flag{ctf}`。

小学妹苦于自己能力不足，于是她拜托你帮她找出文件中的 flag，你能帮帮她吗？

Input

先输入一行一个正整数 n ($1 \leq n \leq 10^5$)，表示接下来字符串的长度。

然后输入一个长度为 n 的字符串 s ，表示某个 txt 文件中的文本内容。

不保证给定的字符串 s 中只包含一个 flag 串，但你只需要输出你找到的第一个 flag 即可。

保证字符串 s 中不含空格，且第一个 flag 一定合法。

Output

输出一行一个字符串 t ，表示破译出来的 flag，格式为 `flag{真正的 flag 内容}`（如样例所示）

Example

standard input	standard output
9 flag03ctf	flag{ctf}
25 tjjjctfctflag05hellozzuli	flag{hello}

Problem 1-8. 数独

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 20 points

数独是一种数学逻辑游戏，游戏由 9×9 个格子组成，玩家需要根据格子提供的数字推理出其他格子的数字。游戏设计者会提供最少 17 个数字使得解答谜题只有一个答案。这种游戏只需要逻辑思维能力，与数字运算无关，所以数学不好的人也很适合。虽然玩法简单，但提供的数字却千变万化，所以不少教育者认为数独是锻炼脑筋的好方法。

规则：

游戏由 9 个 3×3 个的九宫格组成，包含 9 行、9 列和 9 个宫。

每一行、每一列的数字均须包含 1 ~ 9，不能缺少，也不能重复。

每一宫（粗黑线围起来的区域， 3×3 的九宫格）的数字均须包含 1 ~ 9，不能缺少，也不能重复。

如下图为一个典型的数独谜题以及该谜题的解：

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

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

咖波很喜欢做数独！它刚刚做完了好多个数独，但是咖波不会验证自己做的对不对，所以它拜托你写一个程序来帮它判断一下它解对了没有。



Input

先输入一行一个正整数 T ($1 \leq T \leq 200$), 表示查询的次数。

每次查询中, 输入一个 9×9 的矩阵 A , 表示咖波的作答。矩阵中的每个元素用 $A_{i,j}$ 来表示, 其中 $A_{i,j}$ 是矩阵中第 i 行第 j 列的元素, $\forall A_{i,j} \in \{1, 2, \dots, 9\}$

Output

对于每次查询, 如果咖波的作答正确, 则输出 Yes, 否则输出 No。

你可以以任何大小写输出答案中的每个字母。例如, yEs, yes, Yes 和 YES 都将视为肯定的回答。

Example

standard input	standard output
1 5 3 4 6 7 8 9 1 2 6 7 2 1 9 5 3 4 8 1 9 8 3 4 2 5 6 7 8 5 9 7 6 1 4 2 3 4 2 6 8 5 3 7 9 1 7 1 3 9 2 4 8 5 6 9 6 1 5 3 7 2 8 4 2 8 7 4 1 9 6 3 5 3 4 5 2 8 6 1 7 9	Yes

Problem 2-1. 我不吃牛肉

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 25 points

这一天忙碌了一上午的曹同学来到了铁牛面馆来吃面，店长李铁牛师傅十分的热情好客，当场给曹同学推荐了自己的拿手好戏：**铁牛牛肉面**。

饥肠辘辘的曹同学十分期待李师傅的手艺，当场表示要尝一碗铁牛牛肉面，不过不能放葱花，李师傅答应了下来，立刻前往后厨下面。不一会儿，香喷喷的牛肉面便被端了上来，曹同学看到面条上满满的牛肉，而且还没有自己最讨厌的葱花，满脸愉悦，当场说道：“**我爱吃牛肉!**”。

对于一碗香喷喷的铁牛牛肉面，其由以下材料构成：牛肉 (*Beaf*)，葱花 (*Green onion*)，面条 (*Noodles*)，水 (*Water*)，调料 (*Seasoning*)。

铁牛牛肉面之所以十分有名，那便是因为李师傅拥有着祖传的秘方，形式上秘方是一串全为大写字母的字符串，字符串上的字母取自以上五个材料的首个大写字母，即可代表着指定的材料，而一道正宗的铁牛牛肉面必须由长度的连续的秘方字符串的子串组成。

美味只是铁牛牛肉面远近闻名的优点之一，用料足也是一大被客人赞不绝口的优点，因此一道正宗的铁牛牛肉面需要在满足客人要求（即不存在客人忌口的材料）的情况下尽可能的用料充实，具体来说选取的字串必须是所有满足客人要求的字串中最长的。只有这样做出来的铁牛牛肉面才是真正的好吃，实惠不愧于父老乡亲的赞誉。

聪明如你，能找到最长的满足以上条件的秘方字符串的子串长度吗？

Input

第一行输入一个整数 n ($1 \leq n \leq 10^6$)
第二行读入一个长度为 n 的字符串 s ($\forall s_i \in \{'B', 'G', 'N', 'W', 'S'\}$)
第三行读入一个整数 k ，代表客人的忌口的数量. ($0 \leq k \leq 5$)
接下来 k 行，每行读入一个字符串表示客人忌口的材料。

Output

输出一个整数，表示最长的合法字符串子串长度。

Example

standard input	standard output
10 BGWSBBBBBB 1 Green onion	8

Problem 2-2. 小 C 爱答题 (Hard Version)

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 25 points

本题目与 Easy Version 的区别仅为 n, q, a_i 的数据范围不同

经过你的帮助，小 C 成功答对了老师给出的问题，受到老师表扬的他开心极了，爱上了学习。这不，熬夜刷题的小 C 遇到了那道题的 Hard 版本，早已打通任督二脉的他直接把这道题目秒了，现在他把这道交给你，聪明如你能不能解决这道问题呢？

具体问题如下：

给你一个长度为 n 的数组 a ，数组下标从 1 开始，你需要回答 q 次询问，每次询问会给你两个整数 l, r ，你的任务是求出 $a_l \sim a_r$ 的区间和 $(a_l + a_{l+1} + \dots + a_r)$ 和区间异或和 $(a_l \oplus a_{l+1} \oplus \dots \oplus a_r)$ ，并输出两个值异或起来的值。

提示：

\oplus 是异或符号，在 C / C++ 中，求 a 与 b 的异或值的方法是 $a \wedge b$ 。

异或运算规则如下：

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

两个整数进行异或运算，指的是它们转成二进制后（位数不同的在前面补 0），每一位相互异或的结果再转成十进制，如二进制下 $(01)_2 \oplus (10)_2 = (11)_2$ ，即 $1 \oplus 2 = 3$ 。

异或是一种重要的运算，它具有如下性质：

归零性： $a \oplus a = 0$ ；

交换律： $a \oplus b = b \oplus a$ ；

结合律： $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ ；

特别的，对于任意一个整数 a ， $a \oplus 0 = a$ 。

Input

第一行输入 2 个整数 n, q ($1 \leq n \leq 10^6, 1 \leq q \leq 10^6$)

接下来一行输入 n 个整数 $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_i \leq 10^{10}$)

接下来有 q 行，每行读入两个整数 l, r ($1 \leq l \leq r \leq n$)

Output

输出 q 行，每行输出 $a_l \sim a_r$ 的区间和与区间异或和的异或值。

Example

standard input	standard output
5 5	6
1 2 3 4 5	12
1 3	14
2 4	14
3 5	14
1 4	
1 5	

Problem 2-3. 公平的分配

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 25 points

小李喜欢吃糖果，某天他吃到了他认为口感很好的糖果，他想要给他的朋友送上一些，但小李是学校的明星人物，他的朋友有太多太多，他不知道怎么分配糖果，于是选择随机发放糖果，最后一共有 n 个人分配到了糖果，每个人的糖果数分别为 $a_1, a_2, a_3, \dots, a_n$ 。

但这引起了大家的不满，因为这样分配的不公平的，分到糖果少的人会感到伤心，具体来说，一但有两个人的糖果数的差值大于 1，即存在两个人的糖果数为 x, y 满足 $|x - y| > 1$ 时，就会有同学产生不满。

于是小李请你帮他调解朋友之间的不满，你每次可以选择两名同学，让其中一名同学给另外一名同学一颗糖果，这被称为一次操作。

你现在要做的就是最小化操作次数，然后告诉小李你最少需要多少次，就可以让同学之间没有不满，达到公平的分配。

可以证明，至少有一种方案使得分配公平。

Input

第一行一个正整数 n ，表示分到糖果的人数 ($n \leq 5000$)

第二行 n 个正整数 $a_1, a_2, a_3, \dots, a_n$ ，表示每个人分到糖果的数量 ($\forall a_i \leq 10^7$)

Output

一个正整数，代表达到公平分配的最小化的操作次数

Example

standard input	standard output
5 3 13 9 4 7	7

Note

第二个人给第一个人 4 颗糖果 $a_1 \rightarrow 7, a_2 \rightarrow 9$ ，操作数 +4

第二个人给第四个人 2 颗糖果 $a_2 \rightarrow 7, a_4 \rightarrow 6$ ，操作数 +2

第三个人给第四个人 1 颗糖果 $a_3 \rightarrow 8, a_4 \rightarrow 7$ ，操作数 +1

最终每人分别得到的糖果数为 $[7, 7, 8, 7, 7]$

Problem 2-4. 幸运女神的眷顾

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 25 points

小李和小明最近喜欢上了抽奖，但他们运气一直不太好，这让他们很懊恼。

最终他们决定去找气运之子帮忙，气运之子表示自己可以请幸运女神帮忙，但需要小李和小明配合完成一个条件。具体来说，小李和小明会分别获得一个可选择整数域 $[x, y]$ ，这两个整数域可以不同，他们分别可以在自己的整数域内选择一个数字，当且仅当他们俩人选择的数字和为气运之子的运气值 X 时，气运之子才愿意帮忙。

这对聪明且默契的小李和小明来说，只要有办法达到这个条件，他们就能够做到。但他们不知道他们有多少种选择的方法能获得获得幸运女神的眷顾，请你来告诉他。

Input

第一行一个正整数 T ，代表你需要回答 T 个不同的问题 ($T \leq 100$)

对于每一个问题，输入三行数据

第一行一个正整数 X ，代表气运之子的运气值 ($X \leq 2 * 10^9$)

第二行两个整数，空格隔开，代表小李的整数域端点 x_1, y_1 ($1 \leq x_1 \leq y_1 \leq 10^9$)

第三行两个整数，空格隔开，代表小明的整数域端点 x_2, y_2 ($1 \leq x_2 \leq y_2 \leq 10^9$)

Output

对于每一个问题，输出一行数据

一行一个整数 cnt ，代表他们有 cnt 种选择的方法能获得获得幸运女神的眷顾

Example

standard input	standard output
2	3
5	0
2 5	
1 4	
8	
1 2	
3 4	

Note

第一个问题，共有 3 种选法，分别是 (2, 3), (3, 2), (4, 1)。

第二个问题，无论怎么选，都无法满足

Problem 3-1. 坏苹果

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes
Score: 30 points

空梦闲暇时间用 Kotlin 基于 JNI 写了一个 JVM 端的操作 Windows 控制台的库，通过这个库我们便可以在 JVM 中操作控制台。

为了展示库的功能，空梦决定使用这个库编写一个应用程序——在控制台播放 “Bad Apple”。



空梦很快就写好了最基本的代码，但是非常遗憾的是，由于 JNI 存在一些性能损失，空梦逐帧绘制时出现了明显的卡顿，这段程序很显然是需要进行优化的。

空梦想出了两种优化方式：

1. 多线程优化
2. 画布重用

多线程优化顾名思义就是多个线程同时绘制一帧或多帧，以此更加充分的利用 CPU 的各个核心的性能；画布重用就是绘制当前帧时不重置画布，而是在上一帧（如果使用了多级缓存则需要在前 $N - 1$ 帧）的基础上进行绘制，仅重绘两帧之间不同的像素。

这两种优化方法的思路都是可行的，前者尝试通过多线程技术优化程序性能，后者则尝试通过改变程序逻辑来优化性能。

经过一番抉择，空梦决定使用“画布重用”来进行优化，但是由于空梦太笨，没有搞定如何计算两帧之间的不同的区域，并推导出一个最优的重绘方案，现在他来寻求参加天梯赛的各位佬的帮助，希望佬们能够帮助他解决这个问题。

空梦已经为控制台建好了坐标系，坐标系原点位于控制台的左上角， X 轴向右为正， Y 轴向下为正，原点的坐标为 $(0, 0)$ ；同时他也已经对图像进行了预处理，将 “Bad Apple” 的每一帧都转换为了灰度图像（RGB 图像具有三个通道，灰度图像仅有一个通道），并规定了区分亮色与暗色像素的分界线，大于等于 128 的像素认为是亮色，小于 128 的像素认为是暗色。

需要注意的是，控制台在绘制颜色时只能横向绘制，不能纵向绘制（即每次填充的矩形的高度只能为 1），绘制区域超过当前行的范围时会**自动换行**。

假设我们现在有一个背景为亮色、长宽为 3×3 的画布：

222, 255, 160

128, 200, 255

233, 255, 151

现在我们要在 (0, 1) 的位置填充一个宽 4 的暗色矩形，填充后的结果为：

222, 255, 160

000, 000, 000

000, 255, 151

现在请你写一个程序，比较两帧的画面的差异，在保证填充区域最小的情况下，计算出绘制操作数量最少的方案。

Input

第一行输入两个正整数 n, m ($1 \leq n, m \leq 500$)，其中 n 表示画布的宽， m 表示画布的高。

接下来 m 行每行 n 个非负整数，表示上一帧当前行的各个像素的灰度值 $A_{i,j}$ ，两两之间使用逗号间隔（行末没有逗号）。

接下来 m 行每行 n 个非负整数，表示要绘制的这一帧的当前行的各个像素的灰度值 $B_{i,j}$ ，两两之间使用逗号间隔（行末没有逗号）。

注：输入每个灰度值 $A_{i,j}, B_{i,j}$ 均为三位整数，有前缀 0, $\forall A_{i,j}, B_{i,j} \in \{000, 001, 002, \dots, 255\}$ 。保证最后一行数据行末包含换行符。

Output

先输出一行一个正整数 k ，表示最少的操作数。

然后输出 k 行，每行表示一个绘制操作，一个绘制操作的格式如下：

$x, y, color, width$

其中 x, y 表示这个绘制操作起始的坐标点， $color$ 表示要绘制的颜色 (BLACK 或 WHITE, 其中 BLACK 代表暗色，WHITE 代表亮色)， $width$ 表示绘制的宽度。

注意：逗号后包含一个空格，行末没有逗号。

输出时按绘制操作的起始坐标进行排序，逐行扫描（从左到右，从上到下）先扫描到的点先输出。

Example

standard input	standard output
3 3 222,255,160 128,200,255 233,255,151 222,255,160 000,000,000 000,255,151	1 0, 1, BLACK, 4
5 5 001,002,003,004,005 201,202,203,204,205 125,126,127,128,129 128,127,126,125,124 000,111,222,233,255 255,233,222,111,000 128,127,126,125,124 129,128,127,126,125 201,202,203,204,205 005,004,003,002,001	6 0, 0, WHITE, 3 1, 1, BLACK, 4 0, 2, WHITE, 2 3, 2, BLACK, 2 1, 3, WHITE, 4 2, 4, BLACK, 3

Note

在样例输入 1 中，需要修改的坐标是 (0, 1), (1, 1), (2, 1), (0, 2)，均为亮色改为暗色，所以我们在 (0,1) 填充一个宽度为 4 的暗色矩形即可

Problem 3-2. 兄弟们，我起床了，一日之计在于晨呐

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 30 points

兄弟们，我起床了，一日之计在于晨呐 —— *Caiji* 17:37

众所周知，*Caiji* 每天都是早上 7 点 30 准时起床，从来没有睡过懒觉，(*Caiji*: 没错，我从来不睡懒觉)，那这就很奇怪了捏 $\backslash(* - \nabla - *) /$

为什么 *Caiji* 会在这个时间在群中发出这条信息呢？显然，*Caiji* 一定是陷入了时间诅咒（可怕的时间诅咒会让被诅咒者的时间流逝速度和现实的时间流逝速度不同!!!）

善良如你，发现了 *Caiji* 深陷时间诅咒之中怎能袖手旁观？那么，让我们一起开启拯救 *Caiji* 大行动吧！

聪明如你，自然从来不打无准备的仗，必然是有必胜的把握，当然这一切的自信来源于你的强大的能力；能够改变时间的流逝速度!!! 而身为施加邪恶诅咒的大坏蛋当然也不是无能之辈，它也有能力改变时间的流逝速度（不过严格弱于你的能力）。

高手如你，自然有高手的风度，你从来都是用最小的力气赢得胜利，也从来只会在每次敌人出手后出手，这样才能展现出高手的闲庭信步。

形式化来讲：

你将会获得长度为 n 的正整数数组 a ，数组的第 i 项 a_i 代表着 *Caiji* 第 $i-1$ 秒到第 i 秒这段时间下现实中会度过多少秒。

显然， $\sum_{i=1}^n a_i =$ 实际上 *Caiji* 度过的在现实层面下的时间。

而与此同时，在 *Caiji* 角度下 *Caiji* 度过的时间是 n 秒。

之后你会得到一个正整数 k ，代表着大坏蛋能够使某个 a_i 变成 $a_i + k$ 。

接下来你会获得一个整数 q ，表示大坏蛋会出手 q 次。

之后 q 行，每行会得到两个整数 l, r 。

对于每对 l, r ，大坏蛋会使 a_l, a_{l+1}, \dots, a_r 都加上 k ，即对于某个 a_i 经由大坏蛋的操作后会变成 $a_i + k$ 。为了应对大坏蛋的使坏，你会在整局选择一个正整数 m ，使 a_l, a_{l+1}, \dots, a_r 都除以 m 并向下取整，即对于某个 a_i 经由你的操作后会变成 $\lfloor \frac{a_i}{m} \rfloor$ 。对于每对 l, r 你都会在大坏蛋的每次使坏后紧接着出手一次。

请输出满足 $\sum_{i=1}^n a_i \leq n$ 的最小的 m 。

Input

第一行输入一个正整数 n ($1 \leq n \leq 86400$)

第二行输入长度为 n 的整数数组 a ($0 \leq a_i \leq 5000$)

接下来一行输入 2 个正整数 k, q ($1 \leq k \leq 5000, 1 \leq q \leq 100$)

Output

输出一行一个整数，表示最小的 m 。

Example

standard input	standard output
4 1 2 3 4 1 2 1 3 2 4	3

Problem 3-3. 神明的选择

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Score: 30 points

神明遇到了挑战，他现在需要强化他的军队，神明的军队的士兵人数为 n ，战力值分别为 $a_1, a_2, a_3, \dots, a_n$ 。

神明现在有 k 次加强士兵的机会。具体来说，神明每次会等概率的选择一个士兵，然后让他的战力值翻倍，可以重复选择某个士兵。

神明并不在意强化后的士兵的战力值，他只想知道在用完所有强化机会后，士兵组成的军队的战力值的所有情况数，即强化后不同的 a 数组种数。

当且仅当这两个数组中至少有一个数不同时，我们称两个数组是不同的，即当且仅当存在 i 使得 $p_i \neq q_i$ 时，数组 p 与 q 不同。

由于这个结果可能很大，请你把最后的答案对于 998244353 取模

Input

第一行两个整数 n, k ，分别代表士兵的人数和加强士兵的次数 ($1 \leq n \leq 2000, 0 \leq k \leq 2000$)

第二行 n 个正整数，分别是 $a_1, a_2, a_3, \dots, a_n$ ，代表每个士兵的战力值 ($a_i \leq 2000$)

Output

一行一个整数 ans ，代表强化后不同的 a 数组个数对于 998244353 取模的值

Example

standard input	standard output
3 2 1 2 3	6

Note

第一种情况：两次的选择为 (1, 1)，得到 [4, 2, 3]

第二种情况：两次的选择为 (1, 2)，得到 [2, 4, 3]

第三种情况：两次的选择为 (1, 3)，得到 [2, 1, 6]

第四种情况：两次的选择为 (2, 2)，得到 [1, 8, 3]

第五种情况：两次的选择为 (2, 3)，得到 [1, 4, 6]

第六种情况：两次的选择为 (3, 3)，得到 [1, 2, 12]