

# The 21st Tencent Cup Sichuan University Programming Contest



**Tencent 腾讯**

Chengdu, China

May 20th, 2023

## Problem A. 计算异或和

Input file: standard input  
Output file: standard output  
Time limit: 1.5 seconds  
Memory limit: 512 megabytes

给定两个正整数  $n, m$ , 对于所有满足以下两个条件的有序  $m$  元组  $\{a_1, a_2, \dots, a_m\}$ :

1.  $a_1 + a_2 + \dots + a_m = n$
2. 任意  $1 \leq i \leq m$ ,  $a_i$  为非负整数

计算  $a_1 \oplus a_2 \oplus \dots \oplus a_m$  的和。

其中, 两个有序  $m$  元组  $\{a_1, a_2, \dots, a_m\}$  与  $\{b_1, b_2, \dots, b_m\}$  不同, 当且仅当存在  $(1 \leq i \leq m)$ , 使得  $a_i \neq b_i$   
输出答案对  $10^9 + 7$  取模后的结果

### Input

输入两个正整数  $n, m (1 \leq n \leq 10^{15}, 1 \leq m \leq 500)$

### Output

输出一个整数, 表示答案对  $10^9 + 7$  取模后的结果

### Examples

standard input	standard output
4 2	12
20 10	51871212

### Note

样例解释:

对于第一个样例, 所有符合条件的二元组为:  $(0, 4), (1, 3), (2, 2), (3, 1), (4, 0)$ , 对应的  $a_1 \oplus a_2$  分别为 4, 2, 0, 2, 4, 总和为 12。

## Problem B. 最大价值

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       512 megabytes

你是蔬菜大学 XCPC 集训队的一名队员。

蔬菜大学的 XCPC 集训室可以抽象成  $n$  个点  $m$  条边的无向图。门所在的点为  $S$ ，而你的座位位于  $T$ 。

一天，你带着价值为  $k$  的外卖走进了集训室。你发现集训室的每条边上都有一名饥肠辘辘的队员在游荡。具体来说，边  $i$  有边权  $w_i$ ，表示对应队员的饱食度。如果你走过这条边时候手上外卖价值不高于  $w_i$ ，那么这名队员不会理你。否则，他会开始食用你的外卖，直至你手中外卖价值恰好剩余  $w_i$  时候才会放你走。

你不能将一部分外卖暂存至某个点，因为暂存的部分会在你离开该点后离奇地消失。你想知道，从  $S$  出发并到达  $T$  时你手中最多剩余多少价值的外卖。

### Input

第一行五个整数  $n, m, S, T, k$ ，含义如题面所示，保证  $S$  和  $T$  不同。

接下来  $m$  行每行三个整数  $u, v, w$  描述每条边，分别表示两个端点和边权。

$2 \leq n \leq 10^6, 1 \leq m \leq 10^6, 0 \leq k \leq 10^9, 0 \leq w \leq 10^9$

### Output

一个整数，代表最多能带到  $T$  点的外卖价值。如果  $S$  不可达  $T$ ，则认为是 0

### Example

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

## Problem C. 二维数组变换

Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 512 megabytes

给定一个  $n \times n$  的二维矩阵  $A$ ，你需要对左上角为  $(x_0, y_0)$ ，右下角为  $(x_1, y_1)$  的子矩阵进行下面五种操作

- 1  $x_0 y_0 x_1 y_1$ : 顺时针旋转九十度
- 2  $x_0 y_0 x_1 y_1$ : 左右对称翻转
- 3  $x_0 y_0 x_1 y_1$ : 上下对称翻转
- 4  $x_0 y_0 x_1 y_1$ : 延左上至右下对角线翻转
- 5  $x_0 y_0 x_1 y_1$ : 延右上至左下对角线翻转

现给定  $q$  次操作，你需要依次完成，并输出最终结果。

注：保证子矩阵为正方形，如果对操作含义不是很清楚，可结合样例理解

### Input

第一行包含 2 个整数  $n$  和  $q$

接下来  $n$  行，每行  $n$  个整数，描述二维数组  $A$

最后  $q$  行，每行五个整数，代表一种操作，含义如题面所示

$1 \leq n \leq 500, 1 \leq q \leq 100, x_0 \leq x_1, y_0 \leq y_1, 0 \leq A_{i,j} \leq 255$

### Output

输出  $n$  行，每行包含  $n$  个数

### Examples

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

## Problem D. 随机数生成器

Input file: standard input  
Output file: standard output  
Time limit: 15 seconds  
Memory limit: 512 megabytes

有一种随机数生成器工作原理如下：

它存储了若干个数据：一个正整数  $k$ ，以及  $n$  个非负整数构成的数组  $a_1, a_2, \dots, a_n$ 。

每次生成随机数时候，它先反复地在  $[1, n]$  上等概率地随机生成整数，重复  $k$  次，得到序列  $t_1, \dots, t_k$ ；随后计算  $l = \min t_i, r = \max t_i$ ，并返回  $\sum_{i=l}^r a_i$  作为结果。

你喜欢很大的数。你心中有一个非负整数  $v$ ，如果随机数生成器返回的数大于等于  $v$ ，你就会很高兴。

经过一番友好交流，生成器的所有者允许你对内部数据进行修改。具体来说，你可以从数组  $a$  的最前面删去连续零个或若干个数，再从最后面删去连续零个或若干个数，但不能将整个数组全部删完。随后将剩下的部分作为新的  $a$ ，其长度作为新的  $n$ 。

你需要寻找修改方案，使得返回的数大于等于  $v$  的概率尽量大，输出这个最大概率。

### Input

第一行输入正整数  $T$ ，表示数据组数。

对于每组数据，第一行输入三个整数  $n, k, v$ ，第二行输入  $n$  个非负整数表示  $a_1, \dots, a_n$ 。保证  $1 \leq n \leq 3 \times 10^5, 1 \leq k \leq 5, 0 \leq a_i, v \leq 10^9$ 。

保证  $T$  组数据的  $\sum n \leq 3 \times 10^5$ 。

### Output

输出一个浮点数，表示最大概率。你需要保证绝对误差在  $10^{-8}$  以内。

### Example

standard input	standard output
5	1.000000000000
3 1 0	1.000000000000
1 1 1	1.000000000000
3 1 1	0.656250000000
1 1 1	0.967454037008
3 2 1	
1 1 1	
8 2 3	
1 1 1 1 1 1 1 1	
9 5 6	
0 4 1 2 3 1 0 0 0	

## Problem E. 传感器对齐

Input file: standard input  
Output file: standard output  
Time limit: 3 seconds  
Memory limit: 512 megabytes

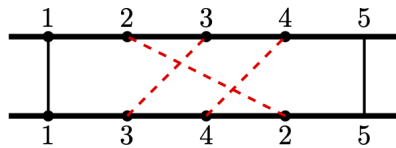
传感器是一种将物理量转变为数字信号的电子元件。虽然传感器能比较准确地测试出某物理量，但很可能出现随机误差。研究员小南在某次实验中，用两个不同的传感器同时对同一个变量进行了  $n$  个单位时间的观测，得到了一些实验数据。两传感器的读数并不完全相同，但大致趋势是统一的。所以小南希望尽可能地将两段读数关联起来，看看能不能得到更准确的结果。

为此，小南制定了如下方案。假设有传感器  $A$  和  $B$ ，它们每间隔单位时间采样得到一个整数，设在第  $i$  时间， $A$ 、 $B$  两传感器的读数分别为  $x_A^{(i)}$  和  $x_B^{(i)}$ 。

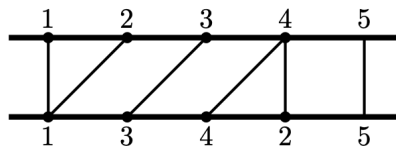
由于不想浪费数据，所以小南要求每个  $A$  传感器的数据要对应一个或多个  $B$  传感器的数据， $B$  传感器的每个数据也要对应一个或多个  $A$  传感器的数据。

另外，因为数据有时序性，所以，如果  $x_A^{(i)}$  与  $x_B^{(j)}$  匹配，则对任意的  $a > i, b < j$ ， $x_A^{(a)}$  与  $x_B^{(b)}$  不能匹配；对称地，对任意的  $c < i, d > j$ ， $x_A^{(c)}$  与  $x_B^{(d)}$  也不能匹配。

以下反例中，虚线部分为错误连线，因为产生了交叉。



该样例的正确连线如下



定义建立了匹配的一组读数的距离为读数之差的绝对值，即  $\text{dist}(i, j) = |x_A^{(i)} - x_B^{(j)}|$ 。距离越小，说明关联程度越高。

小南希望能在满足条件的情况下，最小化整个匹配的距离之和，并得到这个距离之和。

### Input

第一行一个整数  $T$ ，表示测试用例组数。

对于每一组测试样例：

- 第一行一个整数  $n$ ，表示每个传感器观测得到数值的数量
- 第二行  $n$  个整数  $x_A^{(1)}, x_A^{(2)}, \dots, x_A^{(n)}$ ，为  $A$  传感器得到的数据
- 第三行  $n$  个整数  $x_B^{(1)}, x_B^{(2)}, \dots, x_B^{(n)}$ ，为  $B$  传感器得到的数据

数据范围：  $1 \leq T \leq 200$ ，  $1 \leq n \leq 1000$ ，  $-100 \leq x \leq 100$ 。

注意：输入数据中存在纯随机的数据，这种情况不需要特殊处理，只需要按照题面中的要求输出结果，小南看到一个大得不得了数字会考虑换一对传感器。

## Output

对于每个测试用例，输出一行，该行仅有一个整数，为最小的距离。

## Example

standard input	standard output
3	3
5	6
1 2 3 4 5	22
1 3 4 2 5	
5	
1 2 -2 1 9	
3 2 -4 1 7	
7	
1 2 -2 1 9 1 1	
3 2 -4 1 7 9 9	

## Problem F. 珂朵莉树

Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 512 megabytes

珂朵莉树 (Chtholly Tree), 又名老司机树 (Old Driver Tree, ODT), 可以用来 AC 一类特定的维护序列的题目。珂朵莉树基于 C++ 的 `std::set`, 其核心思想是把序列中相邻且相同的元素合并成一个结点保存在 set 里面, 即用 set 中单个结点表示原序列中一整段值相同的子区间。

比如, 序列 (3,3,6,5,2,2,2,2) 的 ODT 有 4 个结点, 分别表示  $[1,2]$  中所有元素为 3,  $[3,3]$  中所有元素为 6,  $[4,4]$  中所有元素为 5,  $[5,8]$  中所有元素为 2。

如果一道维护序列的题目包含操作 "将下标  $[l,r]$  内元素的值修改为  $x$ ", 且测试数据完全随机生成, 则该序列的 ODT 的结点数量期望为  $O(\log n)$ , 于是就可以在这个 set 上暴力处理所有修改和查询的操作。

CBR 正在研究珂朵莉树。他决定从最简单的情况开始: 考虑一个包含  $n$  个元素的序列, 在这个序列上不断执行区间  $[l,r]$  随机、值  $x$  不重复的修改操作, 那么在足够多次操作后, 该序列的 ODT 的结点数量的期望是多少?

CBR 知道这个期望值是  $O(\log n)$  的, 并且他通过大量实验, 观察到  $n$  较大时该期望值接近  $2\ln n$ 。但 CBR 不会算这个期望的准确值, 所以他找你帮忙算一算。

### Input

第一行是一个正整数  $T$  ( $T \leq 1000$ ), 代表测试数据的组数;

接下来的  $T$  行, 每行包含一个正整数  $n$  ( $n \leq 10^{18}$ ), 意义如上所述。

### Output

对于每组数据, 输出一行, 代表 ODT 结点数量的期望值。你的答案的 **绝对误差** 需要小于  $10^{-3}$ 。

### Example

standard input	standard output
5	1.0000000000000000
1	1.6666666666666667
2	2.2000000000000002
3	2.6428571428571428
4	3.0202020202020203
5	

### Note

区间  $[l,r]$  随机: 所有区间  $[l,r]$  的概率相等



## Problem G. 我的车呢？

Input file:           standard input  
Output file:          standard output  
Time limit:           1 second  
Memory limit:        512 megabytes

这是一道交互题。

小H正在停车场寻找他的车。但是不巧的是他是个星际玩家，无法直接看见自己的车，即使车就在他附近。他只能通过手中的车钥匙来判断车的位置。

具体来说，我们将停车场看成一个二维平面，小H的位置表示为一个点  $(x, y)$ ，车的位置也表示为一个点  $(x_1, y_1)$ 。如果小H与车的距离在车钥匙的侦测范围  $r$  内，车钥匙会闪烁提示；否则不会有任何提示。

小H现在位于  $(0, 0)$ ，他的车钥匙正在闪烁。你可以让小H最多移动 300 次，以判断车的位置。

上述  $r, x, y, x_1, y_1$  均为浮点数，且保证  $-10000 < x_1, y_1 < 10000$ ， $0 < r < 10^7$ 。

### Interaction Protocol

首先，交互程序会输出一行，包含一个浮点数  $r$ ，表示车钥匙的探测范围。

你可以输出询问，形式为 “? x y”（不含引号），其中  $x$  和  $y$  是范围无限制的任意浮点数，表示将小H的位置移动到  $(x, y)$ 。此时如果车钥匙会闪烁，则交互系统输出1；否则输出0。

你也可以输出回答，形式为 “! x y”（不含引号），其中  $x$  和  $y$  是浮点数。如果  $(x, y)$  与车的位置的距离的绝对值不超过  $10^{-6}$ ，则视为回答正确。否则会导致 **Wrong Answer**。

你最多可以进行 300 次询问，但只能进行一次回答。

在进行一次询问或回答后，请立即输出换行，并使用下列语句刷新屏幕缓冲区，否则可能会导致超时错误：

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `sys.stdout.flush()` in Python;

### Example

standard input	standard output
5	? 1.0 6.0
1	? 1.0 -4.0
1	! 1.0 1.0

## Problem H. 账号已注销，我想账号已注销了

Input file:           standard input  
Output file:          standard output  
Time limit:           3 seconds  
Memory limit:        512 megabytes

风神从哪儿来  
受伤的狗尾巴摇晃的更热烈  
想你了Gu5enPai  
紧握着你的手把把敌人给锁裂  
我们的队 就像泥头车  
车头看透彻了  
慢 慢 队友都走了  
下场比赛 冠军梦醒了  
先打白金吧  
看 淡 弹幕独轮车  
诶 像一直在被优化  
没队要的是我  
主播不会吹喇叭  
没法把你们给送走  
开播成都四点  
窗外日不在  
睁开眼  
想看清也（青野）看不清了  
喝到起不来  
至少酬勤我还记得  
找一个地方盲扎  
飞起半空中  
是谁又说像他（想他）  
什么风都吹不散呀  
纯净流开启  
不晓得哪时碎的甲（那是谁的家）

### 题目描述

似去的卡神试图通过注销账号来召回天上的 *aono*，他找到了队中最会祷告的狙神来做法招魂。

目前狙神手中拥有  $n$  柱香，依次编号  $1, 2, \dots, n$ ，对应地有高度  $a_1, a_2, \dots, a_n$ 。现在狙神要从这  $n$  柱香中选出至少一柱，将他们按原顺序构成一个新的序列，并丢掉其他未选中的香。考虑到卡神的身高，为了让招魂成功，狙神应当使相邻两柱香的高度差  $\geq k$ 。特殊地，如果只剩一支香，那么招魂总会成功。

狙神希望知道有几种合法的选取方法。两种选取方法彼此不同，当且仅当存在某个位置  $i$  恰好只在一个方法中被选取。

毫无疑问面对如此巨大的数据狙神的大脑很快就过载了，作为铁血 *ftj*，出题人希望你能帮助狙神算出这个数字模  $10^9 + 7$  是多少。

## Input

输入数据有两行:

第一行两个整数  $n, k$  表示有几柱香和高度差。

第二行  $n$  个整数,  $a_i$  表示第  $i$  柱香的高度。

$1 \leq n \leq 500000, 0 \leq a_i \leq 100000, 0 \leq k \leq 100$

## Output

输出一个整数表示组合数, 答案要求模  $10^9 + 7$ 。

## Example

standard input	standard output
4 2 1 2 3 4	7

## Note

对于给出样例, 合法的选取方法是:  $\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_1, a_3\}, \{a_1, a_4\}, \{a_2, a_4\}$

## Problem I. 这真的是签到题

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       512 megabytes

给定一个长度为  $n$  的数列  $a_1, a_2, a_3, \dots, a_n$ , 寻找最大的  $a_i$ , 满足其所有的质因子都在该数列中

### Input

第一行包含一个整数  $n$

第二行包含  $n$  个整数  $a_1, a_2, a_3, \dots, a_n$

$1 \leq n \leq 10^5, 2 \leq a_i \leq 10^6$

### Output

输出一个整数  $a_i$ ,  $a_i$  为最大的且所有质因子在该数列中的数, 如果不存在则输出  $-1$

### Examples

standard input	standard output
10 7 9 9 11 4 11 3 3 8 7	11
2 6 6	-1

## Problem J. BLG冲冲冲！

Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 512 megabytes

今天是2023年5月2日，MSI（Mid-Season Invitational，英雄联盟季中邀请赛）开始啦！今年的MSI分为入围赛和淘汰赛两个阶段，其中入围赛的赛制规则和赛程安排如下：



图1

由上图可以看出，八支战队将一同争夺三个晋级名额，能够晋级的三支战队分别为：A组G3胜者、B组G3胜者、败者组决赛胜者。小B同学是BLG战队（位于上图中A组）的十年老粉，他通过自己无敌的游戏理解预测出了这八支战队两两之间交战的胜负概率，现在他希望基于以上预测计算出BLG战队从A组G3胜出从而获得晋级名额的概率。

### Input

多组数据输入。

第一行有一个整数  $t(1 \leq t \leq 1000)$  表示数据组数。

每组数据包含一个  $8 \times 8$  的矩阵，表示小B同学的预测。矩阵的每个元素都是不小于 0 不大于 100 的整数，第  $i$  行第  $j$  列的元素  $a_{i,j}$  表示若编号为  $i$  的战队与编号为  $j$  的战队交战，编号为  $i$  的战队获胜的概率为  $\frac{a_{i,j}}{100}$ 。保证  $a_{i,j} + a_{j,i} = 100 (1 \leq i, j \leq 8)$ 。每支战队的编号如下：PSG编号为1，DFM编号为2，G2编号为3，LLL编号为4，BLG编号为5，R7编号为6，GG编号为7，GAM编号为8。

## Output

对于每组数据输出一个浮点数表示答案，与标准答案的绝对误差或相对误差不超过 $10^{-6}$ 的结果会被视为正确。换言之，假设你的答案是 $a$ ，标准答案是 $b$ ，则如果 $|a - b| \leq 10^{-6}$ 或 $\frac{|a-b|}{\max(|a|,|b|)} \leq 10^{-6}$ 就可以通过本题。

## Example

standard input	standard output
1 50 50 50 50 0 50 50 50 50 50 50 50 0 50 50 50 50 50 50 50 0 50 50 50 50 50 50 50 0 50 50 50 100 100 100 100 50 100 100 100 50 50 50 50 0 50 50 50 50 50 50 50 0 50 50 50 50 50 50 50 0 50 50 50	1

## Problem K. 字符串游戏

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       512 megabytes

小H和小W在玩一个字符串游戏。

具体来说小H有  $n$  种字符串，小W有  $m$  种字符串，每一种都有无限个。小H可以从这  $n$  种无限个字符串中取出任意个来拼成一个长串（不能为空串）。小W可以从这  $m$  种无限个字符串中取出任意个来拼成一个长串（不能为空串）。如果他们能让这两个串完全相同则游戏成功。

他们想要知道游戏是否能成功，如果能成功，拼成的字符串最短是多长。你能帮帮他们吗？

### Input

第一行两个整数  $n, m (1 \leq n, m \leq 5000)$ 。

接下来  $n$  行每行，每行一个字符串，表示小H手中的字符串。（保证  $n$  个串长度总和  $\leq 5000$ ，字符是小写字母，没有空串）。

接下来  $m$  行每行，每行一个字符串，表示小W手中的字符串。（保证  $m$  个串长度总和  $\leq 5000$ ，字符是小写字母，没有空串）。

### Output

一行一个整数，如果不能成功输出  $-1$ ，如果能成功输出长串的最短长度。

### Examples

standard input	standard output
2 3 ab bc abc abb cabbc	8
2 3 abc bcd abcd a bcde	-1

## Problem L. MEX问题

Input file: standard input  
Output file: standard output  
Time limit: 10 seconds  
Memory limit: 512 megabytes

我们称一个整数序列  $x_1, x_2, \dots, x_k$  为**美丽的**，当且仅当对于所有  $i (1 \leq i \leq k)$ ，都有  $x_i - \text{MEX}(x_1, x_2, \dots, x_i) \leq 1$  成立，其中  $\text{MEX}(x_1, x_2, \dots, x_k)$  表示最小的不在集合  $x_1, \dots, x_k$  中出现的非负整数。例如， $\text{MEX}(1, 0, 1, 3) = 2$ ， $\text{MEX}(2, 1, 5) = 0$ 。

给定一个长度为  $n$  的非负整数序列  $a$ ，计算它有多少个非空子序列是**美丽的**，答案对 998244353 取模。其中序列  $x$  是序列  $y$  的子序列，当且仅当从  $y$  中删除若干（可以为 0）个元素后能得到  $x$ 。

### Input

多组数据输入。

第一行有一个整数  $t (1 \leq t \leq 10^5)$  表示数据组数。

每组数据的第一行有一个整数  $n (1 \leq n \leq 5 \cdot 10^5)$ ，第二行有  $n$  个整数  $a_1, a_2, \dots, a_n (0 \leq a_i \leq n)$ 。

所有数据的  $n$  的总和不超过  $5 \cdot 10^5$ 。

### Output

对于每组数据，输出一个整数表示答案。

### Example

standard input	standard output
4	5
3	3
0 2 1	31
2	7
1 0	
5	
0 0 0 0 0	
4	
0 1 2 3	



## Problem M. 数字模拟

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       512 megabytes

每个数字都可以用5\*5的字符串矩阵来表示，其中矩阵只包含星号“\*”和表示空白的字符“.”。0-9依次表示如下（为了美观，略去字符“.”）：

```
* * * * *
*       *
*       *
*       *
* * * * *
```

```

*
*
*
*
*
```

```
* * * * *
*
* * * * *
*
* * * * *
```

```
* * * * *
*
* * * * *
*
* * * * *
```

```
*       *
*       *
* * * * *
*
*
```

```
* * * * *
*
* * * * *
*
* * * * *
```

```
* * * * *
*
* * * * *
*       *
* * * * *
```

```
*****
      *
      *
      *
      *
      *

*****
*      *
*****
*      *
*****

*****
*      *
*****
      *
*****
```

现在给出一个5\*18的字符串矩阵，代表三个数字，其中每个数字后面紧跟着一列空格用于分隔开数字。  
你需要输出这三个数字，三个数字之间不需要空格。

Input

一个5\*18的字符串矩阵

Output

三个0-9的数字

Example

standard input	standard output
*****  ....* ***** *...*  ....*  ....* *...*  ....* ***** *...*  ....* *.... *****  ....* *****	012

## Problem N. 解密

Input file: standard input  
Output file: standard output  
Time limit: 1.5 seconds  
Memory limit: 512 megabytes

CBR发明了一种独特的加密算法，并把它应用在了他的网站里。为了检验这套算法是否真的安全，你决定尝试破解它。

虽然CBR的前端写得混乱不堪还没有注释，但经过几十分钟的不懈努力，你终于看明白了加密算法的流程。关键步骤概括如下：

被加密的明文是  $m$  个二维向量  $T_0, T_1, \dots, T_{m-1}$ ，加密密钥是  $n$  个  $2 \times 2$  的矩阵  $A_0, A_1, \dots, A_{n-1}$ 。加密时，首先生成加密密钥  $A_i = b_i G^{a_i}$ ，其中  $a_i$  和  $b_i$  是任意两个整数， $G = \begin{pmatrix} 1 & 3 \\ -4 & -5 \end{pmatrix}$ ，然后对于所有  $k = 0, \dots, m-1$ ，计算

$$S_k = \sum_{i+j=k} A_i T_j$$

所得  $S_0, S_1, \dots, S_{m-1}$  即为密文。

同时，你发现解密过程和加密过程很类似。解密密钥是  $m$  个  $2 \times 2$  的矩阵  $B_0, B_1, \dots, B_{m-1}$ 。解密时，对于所有  $k = 0, \dots, m-1$ ，计算

$$T_k = \sum_{i+j=k} B_i S_j$$

所得  $T_0, T_1, \dots, T_{m-1}$  即为原始明文。

所有运算在模素数  $p = 931135487$  下进行。

现在，你有一些加密密钥，你要算出它们对应的解密密钥。因为CBR没有在前端实现解密算法，所以你决定自己动手。

### Input

第一行是一个正整数  $T$  ( $T \leq 5$ )，代表测试数据的组数；

接下来是  $T$  组数据，每组数据中：

第一行包含两个正整数  $n, m$  ( $n, m \leq 10^5$ )，意义如上所述；

接下来  $n$  行，第  $i+1$  行包含 4 个  $[0, p)$  内的整数  $A_{i,1,1}, A_{i,1,2}, A_{i,2,1}, A_{i,2,2}$ ，代表  $A_i$  的 4 个元素。数据保证有解。

### Output

对于每组数据：输出  $m$  行，第  $i+1$  行包含 4 个  $[0, p)$  内的整数  $B_{i,1,1}, B_{i,1,2}, B_{i,2,1}, B_{i,2,2}$ ，代表  $B_i$  的 4 个元素。

### Example

standard input	standard output
2	1 0 0 1
2 2	931135486 931135484 4 5
1 0 0 1	330942419 845502507 734934298 502208379
1 3 931135483 931135482	811063786 267667172 884624420 275729442
2 3	751187425 100246669 487094766 550694087
765802924 214928128 334186154 335946668	
459485228 792738722 184529020 736278758	

## Problem O. 捕鱼达人！

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       512 megabytes

小H是一位捕鱼达人，他经常驾驶船只出海捕鱼。

海上不仅有鱼，还有令人作呕的海洋垃圾。我们可以将海洋视为一个二维平面，小H的船在位置  $(0,0)$  上。海上有  $n$  个物品，每个物品都有一个坐标和一个价值。第  $i$  个物品的价值可记作整数  $v_i$ ，如果它是鱼，则  $v_i \geq 0$ ；如果是海洋垃圾，则  $v_i < 0$ 。

小H有一张神奇的网。这个网可以视为 **边界** 过小船位置的一个圆。小H可以任意调整网的大小，甚至可以将网无限扩大，退化成一个半平面。现在，小H想知道他一次抛网所能获得的所有物品（包含圆边界上的物品）权值和最大是多少？

### Input

第一行一个数 整数  $n(0 < n \leq 1000)$  表示物品的总个数。

第二行  $n$  个整数，第  $i$  个整数  $v_i(-10^7 \leq v_i \leq 10^7)$  表示第  $i$  个物品的权值。

接下来  $n$  行，每行两个整数  $x, y(-10^4 \leq x, y \leq 10^4)$  表示第  $i$  个物品的坐标（保证物品坐标两两不同且不为原点）。

### Output

输出一行一个整数，表示最大权值和。

### Examples

standard input	standard output
5 -1 -3 3 5 -2 -2 2 3 -1 -3 2 -1 4 5 -3	7
3 10 -8 -8 0 10 5 5 -5 5	2

### Note

第一个样例的一种可能方案如下：

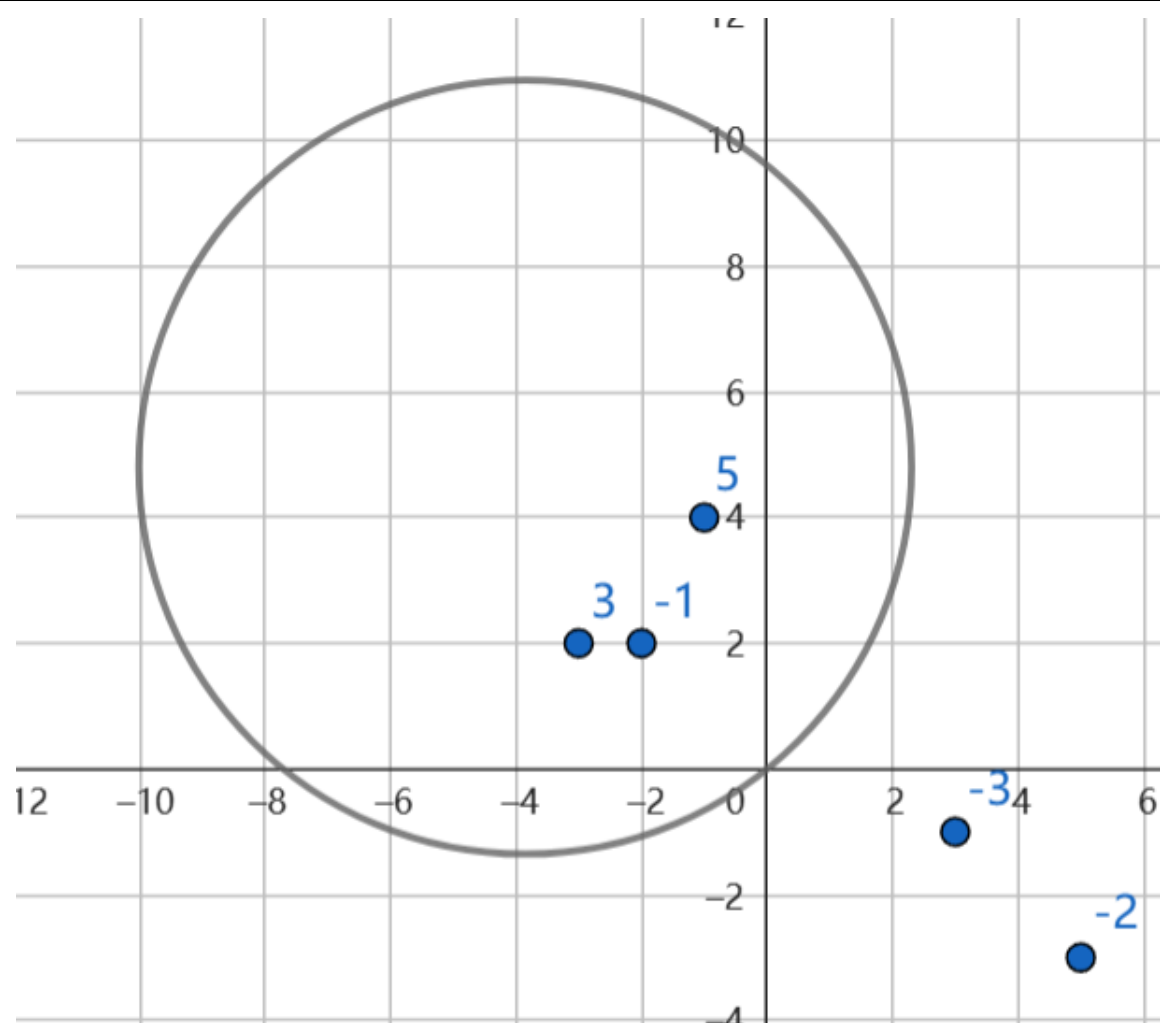


图1