

Problem A. 两人成伴，三人不欢

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

让我们复习一些图论的知识点。本题中讨论的图均为无向图。

- 连通图：如果图中任意不同的两点之间均存在一条路径，则称这张图为连通图。
- 桥：对于连通图中的一条边，如果删除它后图将不连通，则称这条边为桥。
- 边双连通：如果连通图中不存在桥，则称这张图为边双连通图。
- 三元环：称互不相同的三个点 (u, v, w) 为图中的一个三元环，当且仅当存在边 (u, v) , (v, w) 和 (u, w) 。

现在给定一张有 n 个点， m 条边的无向无环图（也就是一个森林），你需要添加一些边，使其成为边双连通图，且没有三元环。输出最少需要添加多少边，并给出一组加边方案。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据组数。
对于每组数据，第一行两个整数 n 和 m ($1 \leq n \leq 10^5, 0 \leq m \leq n - 1$)，分别表示图中节点和边的个数。
接下来 m 行，每行两个整数 u 和 v ($1 \leq u, v \leq n$)，表示一条连接 u 和 v 的无向边。保证图中无重边且无自环，同时保证任意两节点间最多只有一条路径。
保证所有数据的 n 之和不超过 10^5 。

Output

对于每组数据，第一行输出一个整数 k ，表示最少需要添加的边的条数。接下来 k 行，每行输出两个整数 u 和 v ($1 \leq u, v \leq n$)，表示添加一条连接 u 和 v 的边。你需要确保添加边后，图中仍无重边且无自环。
如果有多种 k 最小的方案，输出任意一种即可。如无方案，输出一行 -1 。

Example

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

Problem B. 恶魔的面积

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

给定一个半径为 r 的圆，圆上按逆时针顺序排布着互不相同的五个点 A_i ($i = 1, 2, \dots, 5$)，依次连接 $A_1 - A_3 - A_5 - A_2 - A_4 - A_1$ ，便可以得到一个五角星，我们称上述连接顺序为**五角星顺序**。如果稍加调整排布位置，便可以得到一个逆五芒星，这便是恶魔的象征。

然而恶魔觉得这个图案太死板了，它觉得这个五角星的每个顶点都应该是可以活动的，这样才便于召唤恶魔。所以，恶魔会令点 A_i 在某一个圆的圆周上活动。形式化地，考虑如下五个点集：

$$S_i = \{(x, y) \mid x = r \cos \theta, y = r \sin \theta, \theta_{s_i} \leq \theta \leq \theta_{t_i}\}, i = 1, 2, \dots, 5$$

其中 θ_{s_i} , θ_{t_i} 分别是点 A_i 可以活动的极角范围的左右端点。从 S_i 中任意选取 A_i (保证 S_i 之间没有交集且是逆时针排布)，按五角星顺序连接即可得到一个五角星。对于二维平面上的一个点 P ，如果存在 S_i 中的点 A_i ，且 P 在这五个点对应五角星内，则称其为**恶魔封印点**。请问由所有恶魔封印点构成的点集的面积是多少？五个点对应的五角星，是指把它们按**五角星顺序**连接形成的五角星。

恶魔有 T 个这样的问题要问你，如果你不能尽快回答上来的话，它会诅咒你下次抽塔罗牌时一定会抽出恶魔牌。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据组数。

每组数据有三行，第一行有一个整数 r ($1 \leq r \leq 10^3$)，第二行有五个整数，第 i 个整数表示 θ_{s_i} ($0 \leq \theta_{s_i} \leq 359$)，第三行有五个整数，第 i 个整数表示 θ_{t_i} ($0 \leq \theta_{t_i} \leq 359$)，上述数据中的角度都是**角度制形式**，代表了题目所述五个点集。

保证点集之间没有交集且是逆时针排布，即 $\theta_{s_i} \leq \theta_{t_i}$ ($i = 1, 2, \dots, 5$)，且 $\theta_{t_i} < \theta_{s_{i+1}}$ ($i = 1, 2, \dots, 4$)。

Output

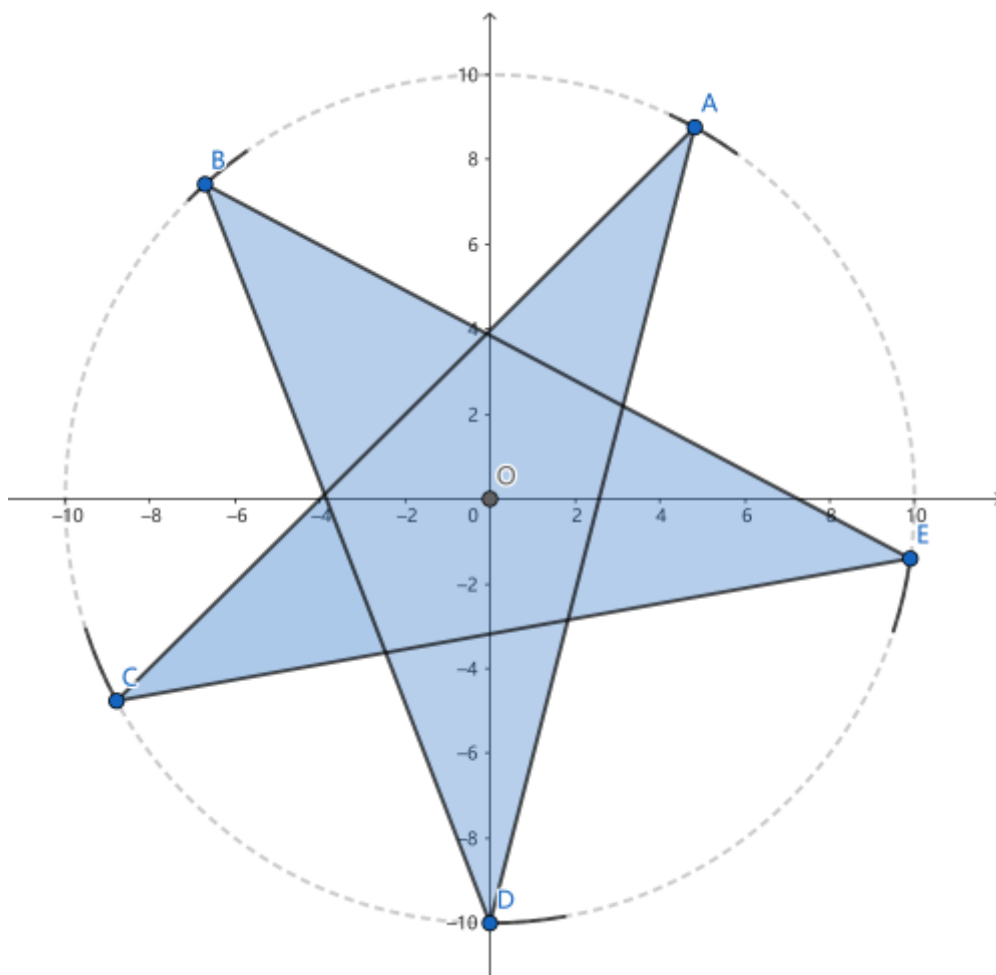
对于每组数据，输出一行一个实数，表示恶魔封印点 P 可能出现的所有位置的面积。只有当你的答案与正确答案之间的相对误差或绝对误差不超过 10^{-6} 时，你的答案才会被认为是正确的。

Example

standard input	standard output
2	112.256994145
10	168.007261527
54 126 198 270 342	
54 126 198 270 342	
10	
54 126 198 270 342	
64 136 208 280 352	

Note

第一个样例给出了半径为 10 的圆的内接正五角星的面积。下图显示了第二个样例可能的五角星，其中五段实线圆弧是样例二的五个点集。



请注意读入大量浮点数可能消耗大量时间。

Problem C. 无线电测向运动

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

这是一道交互题。

无线电测向运动，也称为无线电定向寻找或狐狸狩猎，是一项结合无线电技术与户外导航的运动。参与者使用特制的接收器寻找隐藏的无线电发射器，这项运动考验了方向感和无线电操作技能。

某大学开设了无线电测向运动这一门体育课。期末考试的时候，老师会将同学们带到当地某旅游景点。景点地图可以看作是一个 n 个点的环（ n 为奇数），顺时针依次编号为 $1, 2, \dots, n$ 。老师事先在两个不同的点各埋了一个发射器，然后给了你一个无线电接收器。

这个接收器比较特殊，每次接收信号需要手动按一次接收器上的按钮，然后接收器会告诉你这两个发射器距离你当前所在的点的最短距离之和（距离定义为经过的边数）。期末考核通过的标准是：在使用不超过 40 次接收器的情况下，找到这两个发射器的位置。

现在，请你写一个交互程序，从而能顺利通过期末考试。

Interaction Protocol

首先从标准输入中读取一个正整数 T （ $1 \leq T \leq 10^3$ ），表示测试数据组数。

对于每一组数据，首先从标准输入中读取一个正整数 n （ $3 \leq n \leq 10^9$ ，且 n 是奇数），表示环的点数。然后开始交互过程。每一次交互，你可以向标准输出中输出以下两种格式的数据：

- **? x**: 表示你在编号为 x 的点使用一次接收器，之后你的程序需要从标准输入中读取一个整数 $dist$ ，表示两个发射器到点 x 的最短距离之和。你需要保证 $1 \leq x \leq n$ 。
 - 对于每一组数据，**? x** 操作应不超过 40 次。如果你询问的次数超过了限制，交互器将立即结束，你会接收到「答案错误」的评测结果。
- **! x y**: 表示你的答案，即这两个发射器分别所在的点（顺序无所谓）分别在 x 和 y 。此时：
 - 如果你的输出格式无误且答案正确，那么交互器将立即进入到下一组数据；
 - 否则，如果你的输出格式有误或者答案不正确，交互器将立即结束，你会接收到「答案错误」的评测结果。

注意：每次输出都需要输出换行并且刷新缓冲区，否则你可能会得到意想不到的除了答案正确之外的结果。为了刷新缓冲区，你可以：

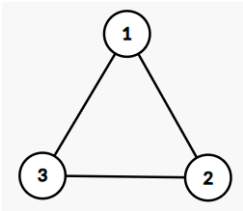
- 对于 C 或 C++，使用 `fflush(stdout)` 或 `cout.flush()`。
- 对于 Java，使用 `System.out.flush()`。
- 对于 Python，使用 `stdout.flush()`。

Example

standard input	standard output
2	
3	
1	? 1
1	? 2
2	? 3
5	! 1 2
4	? 5
3	? 1
	! 2 3

Note

第一组样例如下图，隐藏点为 1 和 2：



- 询问 ? 1，对应的最短距离之和为 $0 + 1 = 1$ ；
- 询问 ? 2，对应的最短距离之和为 $1 + 0 = 1$ ；
- 询问 ? 3，对应的最短距离之和为 $1 + 1 = 2$ ；
- 最后，输出 ! 1 2，答案正确。

Problem D. 都市摩天楼

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

最近你翻出了曾经用过的诺基亚手机，并且成功开机了！这时你想起自己还没通关过都市摩天楼（*City Bloxx*）这款游戏，所以你打算在今天通关。

游戏可以抽象为：给定一个 n 行 n 列的网格，从上到下将行编号为 1 到 n ，从左到右将列编号为 1 到 n ，初始每个单元格中的值均为 -1 。你需要给单元格赋值，每轮你可以选择一个单元格，考虑与其四连通（上下左右）的单元格（不包括你选择的单元格）中的值组成的集合 S ，则你可以将这个单元格重新赋值为 $[0, \text{mex}(S)]$ 中的一个整数，一个格子可以多次进行赋值操作。请构造一种赋值方案，使得最后值为 3 的单元格数不少于 $\lfloor \frac{2(n-1)^2}{3} \rfloor$ 个。

对于一个集合 S ， $\text{mex}(S)$ 表示集合中最小没有出现过的自然数，比如 $\text{mex}(\{-1\}) = 0$ ， $\text{mex}(\{1, 3\}) = 0$ ， $\text{mex}(\{0, 1, 2\}) = 3$ 。

Input

第一行一个整数 T ($1 \leq T \leq 5$)，表示测试数据组数。

对于每组数据，一行一个整数 n ($3 \leq n \leq 100$)，表示网格大小。

Output

对于每组数据，首先输出一行一个整数 k ($1 \leq k \leq 10^5$)，表示操作步数。

接下来输出 k 行，每行输出三个整数 x, y, w ($1 \leq x, y \leq n, 0 \leq w \leq 3$)，表示给第 x 行第 y 列的单元格赋值为 w 。

本题中你不需要最小化操作步数，只需给出满足条件的方案即可。

Example

standard input	standard output
2	(Please refer to the Contest material)
3	
5	

Problem E. 序列操作

Input file: **standard input**
Output file: **standard output**
Time limit: 6 seconds
Memory limit: 1024 megabytes

给定一个长度为 n 的整数序列 a_1, a_2, \dots, a_n 。对于这个序列的某个区间 a_l, \dots, a_r ，一次 *Reduce* 操作会将这个区间的最大值除以 2（向下取整），如果有多个最大值选择下标最小的一个。现有 q 次询问，每次询问给定三个整数 l, r, k ，询问对 a_l, \dots, a_r 区间执行 k 次 *Reduce* 操作之后的区间最大值。询问之间相互独立，也就是说每次询问都是从最初给定的序列开始操作。

Input

第一行两个整数 n, q ($1 \leq n, q \leq 10^5$)，表示序列长度和询问次数。
第二行 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^5$)。
接下来 q 行每行三个整数 l, r, k ($1 \leq l \leq r \leq n, 0 \leq k \leq 10^9$)，表示一次询问。

Output

对于每次询问输出一行一个整数，表示从初始序列开始操作后的区间最大值。

Examples

standard input	standard output
3 2	2
2 0 2	2
2 3 0	
1 3 0	
6 6	1
9 5 0 3 6 7	0
1 4 7	7
3 3 233	0
6 6 0	0
3 4 4	9
4 5 15	
1 1 0	

Problem F. 下载速率显示器

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

你现在正在从某网盘下载东西，但该网盘的下载速度十分堪忧。一直盯着下载速率显示器会让你稍微舒服点。

下载速率显示器按如下方式工作：在第一到第 $(k - 1)$ 秒，下载速率显示器会一直显示 loading，但从第 k 秒开始，第 i 秒会显示第 $(i - k + 1)$ 秒到第 i 秒的平均下载速率。但由于显示屏太小了，因此当平均下载速率大于等于 1024KiBps 时，下载速率显示器会显示单位为 MiBps 下的结果。

你的下载任务会持续 n 秒，给定平均速率的计算区间 k ，你想知道第 k 秒到第 n 秒显示屏显示的内容是什么。

注：1MiBps = 1024KiBps。

Input

第一行输入两个正整数 n, k ($2 \leq n, k \leq 10^5, n > k$)。
接下来一行 n 个正整数 a_i ($1 \leq a_i \leq 10^5$)，第 i 个正整数表示第 i 秒下载了 a_i KiB 的数据。

Output

输出 $n - k + 1$ 行，第 i 行输出从第 i 秒到第 $(i + k - 1)$ 秒显示屏显示的内容。每行输出一个实数和一个字符串，中间用一个空格隔开。首先输出平均下载速率，然后输出单位，为 KiBps 或 MiBps。具体格式可参考样例输出。只有当你输出的平均下载速率与正确答案之间的相对误差或绝对误差不超过 10^{-4} 时，你输出的平均下载速率才会被认为是正确的。

Example

standard input	standard output
5 2	512.500000 KiBps
1 1024 2048 3 5	1.500000 MiBps
	1.001465 MiBps
	4.000000 KiBps

Note

请注意读入大量浮点数可能消耗大量时间。

Problem G. 下载时间显示器

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

你现在正在从某网盘下载东西，但该网盘的下载速度十分堪忧。你觉得只显示下载速率不够缓解你的焦虑，所以你开始计算下载时间。

你所使用的网络带宽为 $B\text{MiBps}$ ，现在你有两个大文件要下载。第一个文件从 t_1 秒时开始下载，大小为 $a_1\text{MiB}$ ，第二个文件从 t_2 秒时开始下载，大小为 $a_2\text{MiB}$ 。由于你使用了最先进的拥塞控制技术，任意时刻只有一个文件在下载时，下载速率会恒定保持为 $B\text{MiBps}$ ，而当有两个文件在下载时，两个文件的下载速率会恒定保持为 $\frac{B}{2}\text{MiBps}$ 。当有文件开始或结束下载时，所有文件的下载速率会瞬间调整为目标值。

现在你想知道两个文件分别需要多长时间下载完成。

Input

第一行一个整数 T ($1 \leq T \leq 10^5$)，表示测试数据组数。

对于每组数据，一行五个整数 B, t_1, a_1, t_2, a_2 ($1 \leq B, a_1, a_2 \leq 10^5$, $0 \leq t_1, t_2 \leq 10^5$, $t_1 \leq t_2$)，意义如题目描述。

Output

对于每组数据输出一行两个实数，分别表示第一个和第二个文件下载完成的耗时。只有当你的答案与正确答案之间的相对误差或绝对误差不超过 10^{-6} 时，你的答案才会被认为是正确的。

Example

standard input	standard output
3	2.000000000 0.800000000
5 0 10 10 4	2.666666667 0.666666667
3 2 7 2 1	3.666666667 3.666666667
3 2 7 3 7	

Note

请注意读入大量浮点数可能消耗大量时间。

Problem H. 完蛋，我被房产包围了

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

很多人有一些钱，期望获得一些房，而你恰恰相反，你有很多房产，但是没多少钱。好在你可以比较精准的预言未来的买房狂潮，以及你有三个「聪明」的助理，你可以将房子储存在他们手里，由他们帮你卖房，获得利润。

你的预言包括两种，按照时间顺序排布。以下展开解释两种预言：

- 一次储存机会，你需要指定一个助理，并把一套价值为 a_i 的房子储存在他手里；
- 一次买房狂潮，三个助理中手里有房储存着的那些助理将分别选择一套房子，把它卖出去，并给你送来相应利润（注意利润和房子原价值很可能不一样）。

以下是你的三个「聪明」的助理：

	小红	小绿	小蓝
卖房时从手里选择出售的房子	最早存入的房子	价值最高的房子	价值最低的房子
卖出价值为 a_i 的房后给你的利润	$a_i - 1$	$a_i - \lceil \frac{a_i}{10} \rceil$	a_i

给定按时间排布的长度为 n 的预言序列，请为其中所有第一类预言选择合适的助理，使得从第二类预言中获得的总利润最大，并输出这个总利润。请注意，预言结束后所有没能卖出的房子都会被清空，不会带来利润。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据组数。

每组测试数据第一行有一个整数 n ($1 \leq n \leq 200$) 表示预言的个数，随后 n 行，每行描述了一条预言，按时间顺序给出，格式如下：

- 1 a_i ($1 \leq a_i \leq 10^6$)，表示第一类预言；
- 2，表示第二类预言。

输入保证 $\sum n \leq 10^4$ 。

Output

对于每组测试数据输出一行一个整数，表示你能获得的最大利润。

Example

standard input	standard output
2	102
7	103
1 40	
2	
1 30	
1 20	
1 15	
1 1	
2	
7	
1 40	
1 30	
1 20	
1 15	
1 1	
2	
2	

Note

对于第一组数据，先把价值 40 的房存给小蓝，然后到来了一次买房狂潮，助理小蓝通过卖房，给你送回了 40 的利润。之后，你再把价值 30 的房存给小蓝，把价值 20 的房存给小红，把价值 15 的房存给小绿，把价值为 1 的房存给小红。在最后一次买房狂潮中，助理小红选择最早存入的房，价值 20，通过卖房给你送回 19 的利润，小蓝给你 30 利润，小绿给你 13 利润，总利润即为 102。可以证明这是获得最多利润的分配方法。

对于第二组数据，一种获得最多利润的分配方法为，40 给小红，30 给小红，20 给小蓝，15 给小蓝，1 给小蓝。

Problem I. 整数反应

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 1024 megabytes

现有一列 n 个整数，从左到右编号为 1 到 n ，这些整数有 0 和 1 两种颜色，每个整数有且只有一种颜色。这些整数按编号从 1 到 n 的顺序依次进入到一个整数多重集合 S_1 里。

每当一个新整数 x 进入 S_1 时，你必须选择一个在 S_1 中的且与 x 颜色不同的整数 y 与 x 产生反应，然后 x 和 y 消失，将反应产物 $x + y$ 插入另一个集合 S_2 中。如果不存在这样的 y ，则不会发生反应，只将 x 放入 S_1 中。

给定这列整数与每个整数的颜色，求当最后一个元素处理完成后， S_2 中最小元素的最大值可能是多少？

Input

第一行一个整数 n ($2 \leq n \leq 10^5$)，表示整数个数。

第二行 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^8$)，表示这列整数。

第三行 n 个整数 c_1, c_2, \dots, c_n ($c_i \in \{0, 1\}$)，其中 c_i 表示第 i 个整数的颜色。

保证至少存在一个 i 满足 $c_i = 0$ ，至少存在一个 j 满足 $c_j = 1$ 。

Output

输出一行一个整数，表示答案。

Examples

standard input	standard output
4 1 3 2 4 0 0 1 1	5
6 1 3 4 2 5 6 0 1 0 1 0 1	4
7 3 3 4 4 5 3 1 0 0 1 1 1 0 0	7

Problem J. 骨牌覆盖

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 1024 megabytes

现有一个 n 行 m 列的网格，行按从上到下的顺序从 1 到 n 编号，列按从左到右的顺序从 1 到 m 编号。第 i 行第 j 列的单元格权值为 $w_{i,j}$ 。

现在你可以用任意多个（包括零个） $1 \times k$ 的骨牌覆盖这个网格，骨牌可以旋转， k 可以取任意正整数，但不能超出网格边界，每个骨牌选取的 k 可以不同。覆盖时需保证这些骨牌不重叠，且没有边相邻。求覆盖的格子权值和最大为多少。

Input

第一行两个整数 n, m ($1 \leq n, m \leq 18$)，表示网格大小。

接下来 n 行，第 i 行 m 个整数 $w_{i,1}, w_{i,2}, \dots, w_{i,m}$ ($|w_{i,j}| \leq 10^6$)，表示单元格的权值。

Output

输出一行一个整数，表示覆盖的格子权值和的最大值。

Examples

standard input	standard output
2 2 2 1 1 1	3
4 4 -1 2 1 -2 3 2 0 -2 -3 -2 0 3 2 3 0 1	15

Problem K. 删数游戏

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Alice 和 Bob 在玩一个删数游戏。
一开始有 n 个整数 a_1, a_2, \dots, a_n , Alice 和 Bob 轮流删数, Alice 先进行操作。每个人可以删去一个最大的数 x , 并选择任意一个小于 x 的非负整数 y , 新增 $1, 2, \dots, y$ 这些数各一个。特别的, 可以选择 $y = 0$, 此时将不会新增数。这意味着当删去 1 时, 不能新增数。删去最后一个数的人胜。
两人都采用最优策略, 请判断 Alice 和 Bob 谁胜。

Input

第一行一个正整数 n ($1 \leq n \leq 10^3$), 表示一开始数的个数。
第二行 n 个正整数 $a_1, a_2 \dots a_n$ ($1 \leq a_i \leq 10^9$)。

Output

输出一行, 如果 Alice 胜, 输出 **Alice**, 否则输出 **Bob**。

Examples

standard input	standard output
2 3 3	Bob
2 2 3	Alice