

2026 南晓开放原子社大学生程序设计新年赛

决赛 赛册



试题列表

A	恭喜发财，……
B	疯狂动物城
C	沈阳的回忆
D	增关之治
E	烟花秀
F	森林冰火人
G	GCD VS XOR
H	好想整顿一下这片大地
I	真的是 $A + B$
J	坤坤的模数
K	马年福印
L	魔术

本试题册共 12 题，19 页。

如果您的试题册缺少页面，请立即通知志愿者。

开赛前请不要打开赛册，否则可能会被判为作弊。

本次比赛由南晓开放原子社 ICPC 竞赛集训队命题
联系方式: IT@openatom.njxzu.cn

承办方



南晓开放原子社

南晓开放原子社

命题方



Nanjing Xiaozhuang University

ICPC International Collegiate Programming Contest Team

南晓开放原子社 ICPC 竞赛集训队

Problem A: 恭喜发财,

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 128 MB

恭喜发财, 红包拿来!



春节将至, 周天子和 **8bits** 在整理年终红包。

桌上摆放着若干堆红包, 共有 n 堆。第 i 堆红包从上到下依次叠放, 共有 k_i 个红包, 每个红包都有一个确定的金额。不同堆的红包数量可能不同, 金额也各不相同。

两人轮流拿红包, 周天子先手, 并且双方都采用最优策略。

1. 周天子每次必须选择任意一个还未取完的堆, 并从该堆的堆顶取走 1 个红包;
2. **8bits** 每次必须选择任意一个还未取完的堆, 并从该堆的堆底取走 1 个红包。

被取走的红包金额将立即计入对应玩家的总收益。当所有红包都被取完时, 游戏结束。

请你帮周天子算算他最多能获得多少收益。

Input

第一行包含一个整数 n ($1 \leq n \leq 10^5$), 表示红包堆的数量。

接下来给出这 n 堆红包的信息。对于第 i 堆红包，第一行包含一个整数 k_i ($1 \leq k_i$)，表示该堆红包的数量；第二行包含 k_i 个整数，表示该堆红包从堆顶到堆底的金额顺序。

保证所有红包堆中红包总数满足 $\sum k_i \leq 5 \times 10^6$ 。每个红包的金额均为整数，且其绝对值不超过 10^9 。

Output

输出一个整数，表示周天子的最终受益。

Examples

Standard Input	Standard Output
3 3 5 1 4 2 7 3 3 6 6 6	24

Note

骗你的，你只是 play 的一环。

Problem B: 疯狂动物城

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB

在这座城市里，正义也需要付出精确计算的成本。



米尔顿

你是林雪獭家族的族长米尔顿，你发现盖瑞的祖宗艾格尼丝持有“气候墙”的专利，不对，是“非法”持有毒液以及危险的武器“牙齿”，你决定派出特种部队逮捕他。

爬行动物的家园可以抽象为一个无向图，包含 n 个节点和 m 条边，节点编号为 $1 \sim n$ 。

1. 你从节点 1 出发；
2. 艾格尼丝位于某个节点（记为目标节点）；
3. 图中有 k 个节点设立了火力点。

为了安全，通过火力点前必须花钱收买该火力，且每个火力点收买价格各不相同。若一条路径经过某个火力点所在的节点，则必须支付该节点对应的费用。

事已密成，你最多收买 10 个火力点。你需要设计一条路线，从节点 1 出发，花费最少的钱前往艾格尼丝所住的节点。

Input

第一行，三个整数 $n, m, k (1 \leq n, m, k \leq 1000)$ ，分别表示节点数、边数以及火力点数量。

接下来 m 行，每行两个整数 $u, v (1 \leq u, v \leq n)$ ，表示节点 u 与节点 v 之间存在一条无向边。

接下来 k 行，每行两个整数 $x, y (1 \leq x \leq n, 1 \leq y \leq 10)$ ，表示节点 x 是一个火力点，收买该火力点需要花费 y 。

最后一行，一个整数 $t (1 \leq t \leq n)$ ，表示艾格尼丝所在的目标节点。

Output

输出一个整数，表示在最多收买 10 个火力点的条件下，到达目标节点所需的最小花费。若无法在限制条件下到达目标节点，输出 -1。

Examples

Standard Input	Standard Output
4 3 0 1 2 2 3 3 4 4	0
5 5 2 1 2 2 5 1 3 3 4 4 5 2 10 3 1 5	1
12 11 11 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11 12 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 1 11 1 12 1 12	-1

Note

样例 1 说明：没有火力点，直接到达目标节点，无需花费。

样例 2 说明：存在多条路径，选择经过火力点费用更低的路线。

样例 3 说明：必须经过超过 10 个火力点，无法满足限制条件。

Make Lynxley Great Again

Problem C: 沈阳的回忆

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB

我努力过无数次，但我知道，机会只会出现在其中的一两次。
——《飞驰人生 2》



我努力过无数次，但我知道，机会只会出现在其中的一两次。

给定一个包含 n 个节点的有向图，节点编号为 $1 \sim n$ 。

图中所有边的长度均为 1，并满足以下条件：

1. 图中固定存在一条长度为 n 的有向环： $1 \rightarrow 2, 2 \rightarrow 3, \dots, (n-1) \rightarrow n, n \rightarrow 1$
2. 另外还存在 m 条额外的有向边，第 i 条额外边为 $(u_i \rightarrow v_i)$ ，其长度同样为 1。

你和“机会”分别从节点 s 与节点 t 出发，出发时间均为 0。你们以速度 1 沿有向边移动，经过一条边恰好消耗 1 个单位时间。

当到达某个节点时，你们会 **立刻**从该节点的所有出边中 **随机选择**一条继续前进，在节点处不发生停留，也不计任何停留时间。

若存在某一时刻，你和“机会”同时出现在同一个节点，则认为你 **邂逅了机会**。

然而，“机会”并不一定会被邂逅。在已知图的结构以及起点 s, t 的情况下，请判断：**你是否仍然有可能在某个时刻邂逅“机会”**？

Input

第一行，两个整数 $n, m (5 \leq n \leq 10^9, 0 \leq m \leq 10^5)$ ，分别表示节点数量与额外有向边的数量。

接下来 m 行，每行两个整数 $u, v (1 \leq u, v \leq n)$ ，表示一条额外的有向边 $u \rightarrow v$ 。

最后一行两个整数 $s, t (1 \leq s, t \leq n)$ ，分别表示你与机会的起始节点。

Output

若仍有可能邂逅，输出 **yes**，否则输出 **no**。

Examples

Standard Input	Standard Output
5 1 1 3 1 5	yes
10 3 1 4 3 6 8 3 1 2	no

Note

样例 1 说明：存在额外边使得两人的时间差可以被抵消，因此存在相遇的可能。

样例 2 说明：无论如何选择路径，两人的到达时间差始终无法消除，无法相遇。

虽然是否能遇到属于自己的机会尚不可知，
但唯有全力以赴，才能在邂逅机会之时将其牢牢抓住。

Problem D: 增关之治

输入文件: standard input
输出文件: standard output
时间限制: 1.5 S
内存限制: 1024 MB



懂王

大美利坚不联合王国现任皇帝——懂王提出了一种关税调整机制。

设有 n 个国家，用一个整数数组 a_1, a_2, \dots, a_n 表示各国当前的关税指标。政府设定了一个固定的关税增量 k ，并允许在多轮政策执行中，对每个国家独立决定是否加征该关税。

在一次关税调整中：对于每个国家 i ，可以选择不加征关税，或加征一个单位的关税 k ；每个国家的选择相互独立，但所有调整在同一轮制裁中同时生效。

形式化地，选择一个数组 $b_1, b_2, \dots, b_n (b_i \in \{0, k\})$ ，然后同时对所有数组元素执行： $a_i = a_i + b_i (1 \leq i \leq n)$ 。

你的任务是在最多只能执行 k 轮上述关税调整的条件下，构造一种可持续化竭泽而渔的结果，即使得最终结果满足 $\gcd(a_1, a_2, \dots, a_n) > 1$ 。

Input

本题包含多组测试数据。第一行为测试组数 $t (1 \leq t \leq 1000)$ 。每组测试数据描述如下。

每组测试数据的第一行包含两个整数 n 和 $k (1 \leq n \leq 10^5, 1 \leq k \leq 10^9)$ ，分别表示数组长度和给定常数。

第二行为 n 个整数 $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9)$ ，表示数组 a 的各个元素。

保证所有测试数据中 n 之和不超过 10^5 。

Output

对于每组测试数据，输出一行 n 个整数，表示经过操作后的最终数组。

输出的各元素要求都在 1 到 $10^9 + k^2$ 之间。如果有多种合法输出，只需输出其中一种即可。

Examples

Standard Input	Standard Output
8	8 10 10
3 3	7 14 21 14
2 7 1	2 2 4 4
4 5	9 6 9 12 9
2 9 16 14	77 99
4 1	10000000000000000001
1 2 3 4	1000000000
5 2	25 15 5
5 6 7 8 9	
2 10	
7 9	
1 1000000000	
1	
1 371	
1000000000	
3 6	
1 3 5	

Note

注意，你无需最小化操作次数，可以证明始终存在合法解。

在第一个测试用例中，输出 $[8, 10, 10]$ 是合法的，因为 $\gcd(8, 10, 10) = 2 > 1$ ，并且可以通过不超过 3 次操作将 $[2, 7, 1]$ 变为 $[8, 10, 10]$ 。

一种可能的操作过程如下表所示：

操作次数	b	操作后的 a
1	$[3, 0, 3]$	$[5, 7, 4]$
2	$[0, 0, 3]$	$[5, 7, 7]$
3	$[3, 3, 3]$	$[8, 10, 10]$

其他诸如 $[2, 10, 4]$ ， $[8, 16, 4]$ ， $[5, 10, 10]$ 也都是合法输出。

在第二个测试用例中，输出 $[7, 14, 21, 14]$ 是合法的，因为 $\gcd(7, 14, 21, 14) = 7 > 1$ ，从 $[2, 9, 16, 14]$ 出发，选择 $b = [5, 5, 5, 0]$ ，即可到达 $[7, 14, 21, 14]$ ，且操作次数不超过 5 次。

Problem E: 烟花秀

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB



烟花

春节期间，城市举行大型烟花表演。指挥系统在二维平面上设置了一台烟花发射装置，其位置记为点 $A(x_a, y_a)$ 。

烟花一旦点燃，只会沿着固定方向 $\vec{v} = (d_x, d_y)$ ($\vec{v} \neq (0, 0)$) 直线飞行，且不会回头。

因此，烟花的飞行轨迹可以抽象为一条射线 $R = \{A + t\vec{v} \mid t \geq 0\}$ 。

夜空中有一盏无人机，用来拍摄烟花，其当前位置记为点 $P(x_p, y_p)$ 。无人机可以在空中进行平移移动（不允许旋转），一次移动的代价等于其移动路径的长度。

为了拍到理想画面，无人机需要移动到某个位置，使得其恰好位于烟花的飞行轨迹 R 上。

请你计算：无人机至少需要移动多远，才能到达射线 R 上的某一点。

Input

第一行包含一个整数 Q ($1 \leq Q \leq 2 \times 10^5$)，表示询问次数。

接下来 Q 行，每行包含六个整数 $x_a, y_a, d_x, d_y, x_p, y_p$ ，分别表示探测射线起点 $A(x_a, y_a)$ ，探测射线的方向向量 $\vec{v} = (d_x, d_y)$ ，以及点 $P(x_p, y_p)$ 的坐标。保证 $(d_x, d_y) \neq (0, 0)$, $(x_a, y_a, d_x, d_y, x_p, y_p) \leq 10^9 + 1000$ 。

Output

对于每个询问，输出一个实数，表示所需的最小平移距离。

若你的答案与标准答案的绝对误差或相对误差不超过 10^{-4} ，则认为你的答案是正确的。

Examples

Standard Input	Standard Output
3	1.000000
0 0 1 0 0 1	0.707107
0 0 1 1 1 0	2.000000
1 1 1 0 -1 1	

Note

在第一个询问中，探测射线沿 x 轴正方向发出，点 P 位于 $(0, 1)$ ，到射线的最小距离为 1；

在第二个询问中，点到射线的垂足位于射线方向上；

第三个询问中，垂足位于射线反方向，因此最近点为射线的起点。

Problem F: 森林冰火人

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB



森林冰火人

森林正在进行总统选举，现在有两个候选人——“火人”和“冰人”。森林崇尚民主共和，有着和灯塔相似的选举制度。森林中共有 n 个州，第 i 个州有 a_i 张选举人票和 b_i 个选民。当获得一个州的一半以上选民的选票时，便获得了这个州的所有选举人票。当获得一半以上的选举人票时，便赢得了大选。

现在假设每个州的选民人数以及总的选举人票数都是奇数，且每个选民都会投出自己的票。请你帮“火人”算算，他最少需要获得多少个选民的票才能赢得大选。

Input

第一行一个整数 $n(1 \leq n \leq 10^3)$ ，表示州的数量。

接下来 n 行，每行两个整数 a_i, b_i ， $a_i(1 \leq a_i \leq 20, 1 \leq b_i \leq 10^9)$ 表示第 i 个州的选举人票数， b_i 表示第 i 个州的选民人数。

保证所有 b_i 为奇数，且 $\sum a_i$ 为奇数。

Output

输出一个整数，表示“火人”为赢得大选至少需要获得的选民票数最小值。

Examples

Standard Input	Standard Output
3 5 9 3 7 1 1	5
5 7 21 4 9 4 9 3 7 1 1	14

Problem G: GCD VS XOR

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB

G 题不能不考 gcd, 正如西方不能失去耶路撒冷。

——出题人

你有一个正整数 x 。

你需要找到另一个正整数 y , 使得 y 严格小于 x , 并且满足方程 $\gcd(x, y) = x \oplus y$ 。

Input

第一行, 包含一个整数 $t (1 \leq t \leq 10^4)$ 表示测试数据的数量。

每个测试用例只有一行, 包含一个整数 $x (1 \leq x \leq 10^{18})$ 。

Output

对于每个测试用例, 输出一个正整数 y , 如果你找到了一个可行的答案, 否则输出 -1 。

Examples

Standard Input	Standard Output
2	10
15	-1
1	

Note

$\gcd(x, y)$ 表示 x 和 y 的最大公约数, 例如 $\gcd(4, 6) = 2$ 。

按位异或是一种二进制操作, 它接受两个等长的位模式, 并对每对相应位执行逻辑异或操作。如果两个位中只有一个为 1, 则每个位置的结果为 1; 但如果两个位都为 0 或都为 1, 则结果为 0。

最短的题目, 最深的思考!

Problem H: 好想整顿一下这片大地

输入文件: standard input
 输出文件: standard output
 时间限制: 1 S
 内存限制: 1024 MB

走在这片大地上, 每个人都有每个人的抱负和野心, 说真的, 正值过年时分, 我们常常出门压马路, 周围全是不熟的亲戚怎么办呢, 东张西望一下?



那么你一定注意到脚下的砖块, 每一块砖的形状都是相同的 1×2 , 大小固定, 如果现在给你一块空地皮 $n \times m$, 你希望自己有能掌控这片大地, 所以你需要找到用这些砖头填满这片大地的方案数, 也就是说, 请你把这片 $n \times m$ 的大地分割成若干个 1×2 的长方形, 求方案数。

例如 $n = 2, m = 3$, 共有三种方案。

Input

输入包含多个测试用例。

每个测试用例由两个整数构成: 大矩形的高度 n 和宽度 $m, 1 \leq n, m \leq 11$ 。

当 $n = m = 0$ 时输入结束。

Output

对于每个测试用例, 输出使用大小为 2×1 的小矩形填充给定大矩形的不同方式的数量。

假设给定的大矩形是定向的, 即对称的铺砌方式应多次计算。

Examples

Standard Input	Standard Output
1 2	1
1 3	0
1 4	1
2 2	2
2 3	3
2 4	5
2 11	144
4 11	51205
0 0	

Problem I: 真的是 A + B

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB

本题为交互题。

评测程序中隐藏了两个整数 a 和 b 。你的任务是通过交互获得它们，并输出它们的和。

Input

第一行，一个整数 n 代表有 n 轮交互。

接下来每一行，两个整数 a, b ($-10^{15} \leq a, b \leq 10^{15}$)。

Output

每次输出交互输入的 $a + b$ 的值。

Examples

Standard Input	Standard Output
2	3
1 2	7
3 4	

Note

交互题参考: <https://oi-wiki.org/contest/interaction/>

每次打印查询后，不要忘记输出换行符并刷新输出。否则，你将得到超时。

- 在 C++ 中使用 `fflush(stdout)` 或 `cout.flush()`;
- 在 Python 中使用 `sys.stdout.flush()`;
- 其他语言请参阅文档。

Problem J: 坤坤的模数

输入文件: standard input
 输出文件: standard output
 时间限制: 1 S
 内存限制: 1024 MB

你居然帮助坤坤解决了倍数问题！

新的一年坤坤更深入的学习数学题，现在他有一个新的问题来请教你！

如果你做出来，坤坤将给你一个大红包（真的）！

现在有一个正整数 n ，选择所有不大于 n 的正整数 i 分别让 n 取余^[1] i ，将得到的结果从大到小降序排序，得到的新序列为 a 。

例如当 $n = 10$ 时：

$$\left\{ \begin{array}{l} 10 \bmod 1 = 0 \\ 10 \bmod 2 = 0 \\ 10 \bmod 3 = 1 \\ 10 \bmod 4 = 2 \\ 10 \bmod 5 = 0 \\ 10 \bmod 6 = 4 \\ 10 \bmod 7 = 3 \\ 10 \bmod 8 = 2 \\ 10 \bmod 9 = 1 \\ 10 \bmod 10 = 0 \end{array} \right. \rightarrow \{0, 0, 1, 2, 0, 4, 3, 2, 1, 0\}$$

↓sort

$$a = \{4, 3, 2, 2, 1, 1, 0, 0, 0, 0\}$$

坤坤想要知道该余数序列降序排序后，前 k 项的和，请你帮助他求出。

[1] 模运算，又称取余运算，常用 $a \bmod b = c$ 来表示 a 除以 b 的余数是 c 。

Input

在一行上输入两个正整数 n, k ($1 \leq k \leq n \leq 10^9$)。

Output

输出一个整数，代表余数序列降序排序后前 k 项的和。

Examples

Standard Input	Standard Output
10 5	12

Note

样例的处理过程已经在题干中展示，答案为 $4 + 3 + 2 + 2 + 1 = 12$ 。

Problem K: 马年福印

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB

马年将至，人们会在一面很长的墙上依次贴上写有字符的“福字”。每一张福字只包含一个字符，不同位置的福字彼此独立，同一种字符的福字可以出现多次。

这些福字从左到右排列，形成一个字符串 S 。

给定另一字符串 T ，表示你想要拼出的新年祝福语。

你可以从 S 中选择一个连续区间 $[l, r]$ ($1 \leq l \leq r \leq |S|$)，并仅使用该区间内的福字来拼接字符串。拼接时字符的顺序可以任意调整，但每一张福字只能使用一次。

请你计算，共有多少个连续区间 $[l, r]$ ，使得区间 $S[l \dots r]$ 中的字符数量足够拼出字符串 T ，即：对任意字符 c ，字符 c 在 $S[l \dots r]$ 中出现的次数，不少于其在 T 中出现的次数。

Input

第一行包含一个字符串 S ($1 \leq |S| \leq 2 \times 10^5$)，字符串均仅由小写英文字母组成。

第二行包含一个字符串 T ($1 \leq |T| \leq 5000$)，字符串均仅由小写英文字母组成。

Output

输出一个整数，表示满足条件的连续区间数量。

Examples

Standard Input	Standard Output
abcba bc	8

Note

区间为 $[1, 3], [1, 4], [1, 5], [2, 3], [2, 4], [2, 5], [3, 4], [3, 5]$ 。

Problem L: 魔术

输入文件: standard input
输出文件: standard output
时间限制: 1 S
内存限制: 1024 MB



魔术扑克

魔术师 Albert_Li 被邀请去马年某地春晚表演，他想设计一个魔术，规则如下。

请你帮他看看能否成功策划这个魔术。

魔术师 Albert_Li 有一副由 n 张卡牌组成的牌堆 s ，每张卡牌上都写有一个整数，表示该卡牌的数值。

Albert_Li 可以从牌堆中选取两张卡牌，设它们的数值分别为 x 和 y 。如果 $|x - y| = 1$ ，那么他可以执行以下两种魔术操作之一：

- 将这两张卡牌的数值分别变为 $x - 1$ 和 $y - 1$ ，然后放回牌堆；
- 将这两张卡牌的数值分别变为 $x + 1$ 和 $y + 1$ ，然后放回牌堆。

你也有一副包含 n 张卡牌的牌堆 t 。你想知道，Albert_Li 是否可以通过执行任意次（可能为 0 次）上述魔术操作，使得他的牌堆与您的牌堆完全相同。

请注意，牌堆中可以存在数值相同的卡牌，并且这些卡牌之间没有任何区别。

Input

第一行，包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。

每个测试用例的第一行，包含一个整数 n ($1 \leq n \leq 10^5$)，表示牌堆中卡牌的数量；

第二行，包含 n 个整数 s_1, s_2, \dots, s_n ($1 \leq s_i \leq 10^9$)，表示 Albert_Li 的牌堆中各卡牌的数值；

第三行，包含 n 个整数 t_1, t_2, \dots, t_n ($1 \leq t_i \leq 10^9$)，表示你的牌堆中各卡牌的数值。

保证所有测试用例中 n 的总和不超过 10^5 。

Output

对于每个测试用例，如果 Albert_Li 可以通过若干次魔术操作使他的牌堆与您的牌堆完全相同，则输出 Yes；否则输出 No。

Examples

Standard Input	Standard Output
3	Yes
3	Yes
1 10 100	No
100 10 1	
2	
1 2	
99 100	
3	
1 2 3	
1 1 6	

Note

在第一个样例中，两个牌堆本身就已经相同，因此输出 Yes。

在第二个样例中，对数值为 1 和 2 的两张卡牌连续执行第二种魔术操作 98 次后，可以得到目标牌堆，因此输出 Yes。

在第三个样例中，无法将 Albert_Li 的牌堆变换为目标牌堆，因此输出 No。