

AIM Tech Round 5(Div. 1 + Div. 2)

I. 题目链接

- [A. Find Square](#)
- [B. Unnatural Conditions](#)
- [C. Rectangles](#)
- [D. Order book](#)
- [E. Restore Array](#)
- [F. Make Symmetrical](#)
- [G. Guess the number](#)
- [H. Make Square](#)

II. 简要翻译

A. Find Square

在一个 $n \times m$ 的方格矩阵内，有且仅有一个边长为奇数的正方形，满足该正方形内的所有方格为黑色 (**B**)，除此之外，其余所有方格均为白色 (**W**)。你的任务是找出该正方形的中心点所在的方格位置并输出其坐标。

$$1 \leq n, m \leq 115。$$

B. Unnatural Conditions

令 $s(x)$ 为 x 的各个数位上的数字之和。给定 n, m ，你的任务是找到两个正整数 a, b 并输出，其中 a, b 满足：

- $s(a) \geq n$
- $s(b) \geq n$
- $s(a + b) \leq m$

要求输出的 a, b 不得包含前导零，且单个数的长度不能超过 2230。

$$1 \leq n, m \leq 1129。$$

C. Rectangles

给定平面直角坐标系中的 n 个矩形，你的任务是找到一个点 (x, y) ，使得该点至少在 $n - 1$ 个矩形的内部（包含边界），输出该点的坐标。保证存在合法解。

$$2 \leq n \leq 132674，给定矩形的顶点坐标的绝对值不超过 10^9 。$$

D. Order book

一个初始为空的数集。该数集中仅包含两类数： A 类数与 B 类数，且满足在任意时刻，数集中所有 A 类数的最小值（若存在）严格大于所有 B 类数的最大值（若存在）。现在按顺序有 n 次操作，操作共有如下两类：

- **ADD p**：向数集中插入一个数字 p ，但数字 p 的类型不确定（即数字 p 可能为 A 类数，也可能为 B 类数）。
- **ACCEPT p**：取走了数集中的数字 p ，且数字 p 满足为当前数集中所有 A 类数的最小值或所有 B 类数的最大值。

已知向数集中插入的所有数字均不相同。求这 n 次操作共有多少种可能的情况。两种情况被视为不同当且仅当存在一次插入操作，插入的数字 p 的类型不相同。输出可能情况的总数对 $10^9 + 7$ 取模后的结果。

$1 \leq n \leq 363304, 1 \leq p \leq 308983066。$

E. Restore Array

有两个长度为 n 的数列 a, b ，下标编号为 $1 \sim n$ ，已知数列 b 满足：

$$b_i = \begin{cases} a_i \bmod a_{i+1}, & i < n \\ a_n \bmod a_1, & i = n \end{cases}$$

给定数列 b ，求是否存在满足条件的数列 a ，若存在，则在第一行输出 YES，并在第二行输出数列 a ；若不存在，输出 NO。

$2 \leq n \leq 140582, 0 \leq b_i \leq 187126。$

F. Make Symmetrical

一个初始为空的整点集合。你需要实现如下三种操作：

- 1 x y：向集合中插入点 (x, y) ，保证在这之前集合中不存在该点。
- 2 x y：从集合中删除点 (x, y) ，保证在这之前集合中存在该点。
- 3 x y：求至少还需要添加多少点，使得每个点都能找到一个与之关于过原点和点 (x, y) 的直线对称的点。特别地，若一个点在该条直线上，那么它的对称点为它本身。

共有 q 次操作，对于每个操作 3，输出一行一个整数，表示答案。

$1 \leq q \leq 2 \times 10^5, 1 \leq x, y \leq 112904$ ，保证操作 3 的数目不会超过 10^5 ，操作 1 与操作 2 的数目和不会超过 10^5 。

G. Guess the number

这是一道交互题。

你需要完成一个猜数游戏，你的目的是猜测一个数字 x ，其中 x 满足 $x \in [1, M], M = 10004205361450474$ 。

你有最多 5 次询问机会，每次询问你需要输出一个正整数 k ，同时输出一个长度为 k 的严格递增的序列 t （下标编号为 $0 \sim k - 1$ ），满足 $1 \leq k \leq 10^4$ 且 $k \leq x$ 。每次询问得到的反馈为一个整数 i ，对应的意义如下：

- 若 $i = -2$ ，表明你在该次询问中的输出非法，或者 5 次机会用尽，游戏失败，你应立即结束程序。
- 否则，若 $i = -1$ ，表明你给出的序列中包含了数字 x ，游戏获胜，你应立即结束程序。
- 否则，数字 x 满足 $t_{i-1} < x < t_i$ 。特别地，若 $i = 0$ ，表示 $x < t_0$ ；若 $i = k$ ，表示 $x > t_{k-1}$ 。

注意每次输出之后刷新缓冲区（C++ 语言需执行 fflush(stdout) 语句或 cout.flush() 语句）。

H. Make Square

我们称数列 b_1, b_2, \dots, b_m 是“优秀的”，当且仅当存在 $i < j (1 \leq i, j \leq m)$ 满足 $b_i \times b_j$ 是完全平方数。

给定长度为 n 的数列 a ，下标编号为 $1 \sim n$ ，共有 q 次询问，每次给定 l, r ，求使得数列 $a_l, a_{l+1}, \dots, a_{r-1}, a_r$ 成为“优秀的”至少需要多少次操作。一次操作定义为将数列中的任意一个数乘或除以一个质数（若除以一个质数，需要保证能够整除）。对于每次询问，输出一行一个整数，表示答案。

$2 \leq n \leq 194598, 1 \leq q \leq 1049658, 1 \leq a_i \leq 5032107, 1 \leq l < r \leq n。$