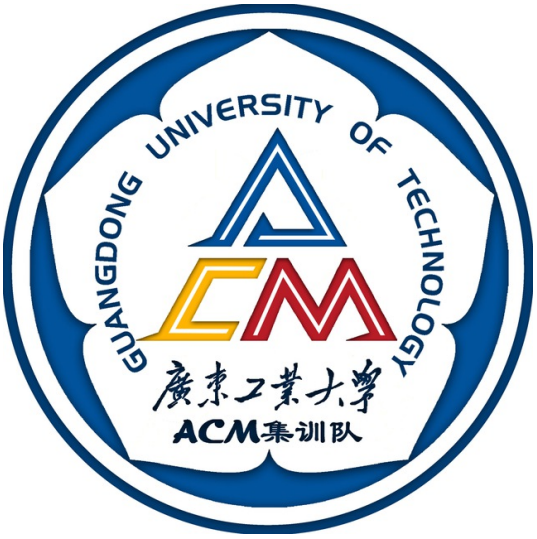
2025 年广东工业大学 ACM 新生程序设计竞赛（初赛）

广东工业大学 ACM 集训队

2025 年 10 月 23 日



试题列表

A	最大子串匹配删除操作
B	我不是 AI
C	逃出生天
D	师出同门
E	稳定区间
F	【模板】Pollard-Rho
G	互异排列—扩展
H	互异排列—生成
I	2-冲突数对
J	收藏宝石

共 10 题

Problem A. 最大子串匹配删除操作

输入文件: standard input
输出文件: standard output

给定两个字符串 S 和 T ，每回合，你可以执行以下两种操作之一，直到 T 变为空字符串：

1. 删除操作：删除字符串 T 开头的第一个字符。
2. 匹配删除操作：如果在 S 中存在一个子串 $S[i,j]$ 与当前的 T 完全相同，则可以删除 S 的前缀 $S[1,j]$（即保留 $S[j+1,|S|]$），同时删除字符串 T 开头的第一个字符。

你的目标是计算 **最多** 可以执行多少次匹配删除操作。

输入格式

第一行和第二行分别输入仅由小写英文字母构成的字符串 S 和 T（$1 \le |S|, |T| \le 2 \times 10^5$）。

输出格式

输出一个整数，表示最多可以执行匹配删除操作的次数。

样例

standard input	standard output
abcb aab	2
ddddddd dddd	3
lovelloovvee love	3
hahahahahaha haha	4
acmer medal	0
treewithsegmentcalledsegmenttree segment	3

提示说明

在第一个测试样例中，按如下顺序操可以执行最多次数的匹配删除操作：

- T=“aab”，S=“abcb”，执行删除操作。
- T=“ab”，S=“abcb”，执行匹配删除操作，匹配 $S[1,2]$。
- T=“b”，S=“cb”，执行匹配删除操作，匹配 $S[2,2]$。

然后字符串 S 和字符串 T 均为空，无法再执行任何操作。

Problem B. 我不是 AI

输入文件: standard input
输出文件: standard output

现在 AI 发展得越来越厉害，为了杜绝 AI 代替选手参赛的情况，请通过此题。

输入格式

输入一行字符串组成的句子，包含大写或小写英文字母组成的字符串。

保证每个字符串间只由一个空格分隔，且行末没有多余的空格。

输出格式

- 当输入 **严格等于** `Are you AI` 时（大小写完全匹配），输出一行 `I am not AI`
- 其他所有情况都输出一行 `I do not know`

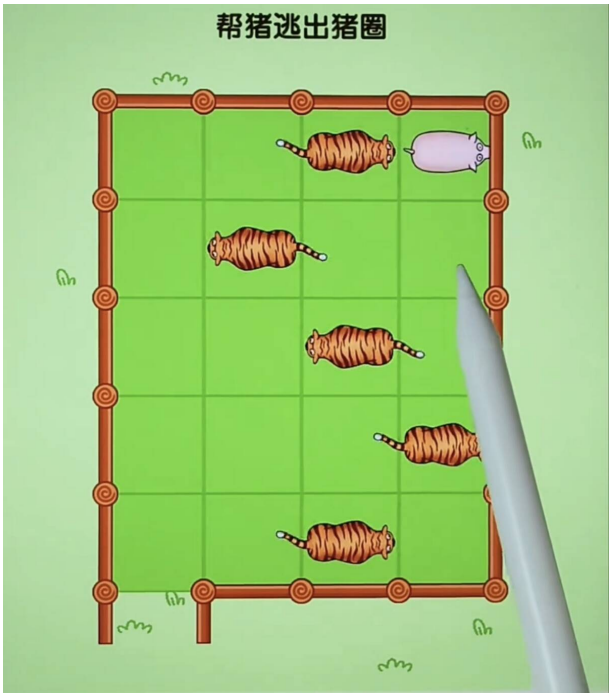
样例

standard input	standard output
Are you AI	I am not AI
are you ai	I do not know
How are you	I do not know
AI you Are	I do not know

Problem C. 逃出生天

输入文件: standard input
输出文件: standard output

在一个 n 行 m 列的网格猪圈中，一只可怜的小猪被困在了右上角的位置 $(1, m)$。它的目标是到达左下角的出口 $(n, 1)$，但网格中充满了危险，每一行都有一只凶猛的老虎在巡逻！



每时刻，**小猪先执行移动，老虎后执行移动**，移动方式如下:

- 小猪移动时，可以选择 **移动到相邻格子或停留在原地** 。也就是说，如果小猪在当前时刻前处在 (x, y)，则下一时刻小猪可以移动到网格范围内的 $(x-1, y), (x+1, y), (x, y-1), (x, y+1), (x, y)$ 其中之一。
- 老虎移动时，只会尝试沿着当前方向移动：如果老虎前进方向的下一个格子在网格范围内，则朝当前方向前进一格；否则老虎将先把移动方向反向，然后再朝当前方向前进一格。

在任意时刻的小猪或老虎移动前或移动后，需要保证小猪不能移动到网格外，且不能与任何老虎处在同一格子中（包括移动到出口的时刻）。

保证初始时小猪与所有老虎不在同一格子，请你帮小猪判断它是否可以达成目标。

输入格式

第一行包含两个整数 n, m（$2 \le n, m \le 100$），表示网格的行数和列数。

接下来 n 行，第 i 行包含一个整数 c_i（$1 \le c_i \le m$）和一个字符 d_i（$d_i \in \{L, R\}$）。

- c_i 表示第 i 行老虎的初始列位置。
- d_i 表示老虎的初始移动方向，‘L’ 表示向左，‘R’ 表示向右。

保证 $c_1 \neq m$ 。

输出格式

如果小猪能够到达出口，输出 “YES”，否则输出 “NO” 。

你可以以任意大小写输出 “YES” 和 “NO”（例如，字符串 “yEs”、“yes”、“Yes” 和 “YES” 将被识别为肯定的回答）。

样例

standard input	standard output
2 2 1 R 1 R	NO
2 5 3 L 2 R	YES
5 4 3 R 2 L 3 L 4 R 3 R	NO

提示说明

对于第一个测试样例，小猪无法到达出口：

- 在第一个时刻，小猪只能移动到 $(2,2)$ 或停留在 $(1,2)$ 。
- 无论它如何选择，老虎移动后都将与小猪处在同一格子中。

对于第二个测试样例，小猪可以按以下次序逐步移动到 $(2,1)$ ：

- 小猪移动到 $(1,4)$ ，然后两只老虎分别移动到 $(1,2)$ 和 $(2,3)$ 。
- 小猪移动到 $(1,3)$ ，然后两只老虎分别移动到 $(1,1)$ 和 $(2,4)$ 。

- 小猪移动到 $(2,3)$，然后两只老虎分别移动到 $(1,2)$ 和 $(2,5)$。
- 小猪移动到 $(2,2)$，然后两只老虎分别移动到 $(1,3)$ 和 $(2,4)$。
- 小猪移动到 $(2,1)$，然后两只老虎分别移动到 $(1,4)$ 和 $(2,3)$。

Problem D. 师出同门

输入文件: standard input
输出文件: standard output

在修仙界，玄真国与幻月国长期对峙，近日终于爆发大战。

两国的军队均由来自 n 个不同宗派的修士组成，玄真国军队中来自第 i 种宗派的修士有 a_i 名，而幻月国军队中来自第 i 种宗派的修士有 b_i 名。

每个宗派的修士都掌握独特的功法，若两国派出同一宗派的修士对阵，则因师出同门，双方不会相互攻击；否则，双方将各祭出杀招，导致双方修士均陨落一名。

具体来说，当两国交战时，每次战斗的规则如下：

- 玄真国从还有修士的宗派中选择一个（设宗派为 i ），幻月国也从还有修士的宗派中选择一个（设宗派为 j ）。
- 如果 $i = j$（即同一宗派），则双方修士不会互相攻击。。
- 如果 $i \neq j$，则双方各损失一名修士（即 a_i 和 b_j 各减少 1 ）。

我们称 $i \neq j$ 的战斗为“有效战斗” 。

现在的问题是：是否存在一系列战斗，使得在战斗结束后，**玄真国或幻月国中至少有一方** 的修士全部阵亡（即所有 $a_i = 0$ 或所有 $b_i = 0$ ）?

如果存在这样的战斗序列，请输出 **第一次有效战斗** 中双方选择的宗派编号 i 和 j（$i \neq j$）。如果存在多组解，输出任意一组即可。

输入格式

第一行输入一个整数 t（$1 \leq t \leq 10^4$），表示测试用例数量。

对于每个测试用例：

- 第一行一个整数 n $(1 \leq n \leq 10^5)$ ，表示宗派数量。
- 第二行 n 个整数，第 i 个表示 a_i $(1 \leq a_i \leq 10^9)$ 。
- 第三行 n 个整数，第 i 个表示 b_i $(1 \leq b_i \leq 10^9)$ 。

数据保证所有测试的 n 总和不超过 10^5 。

输出格式

对于每个测试用例：

- 若无法达成目标，输出一行 "NO"。
- 若可以达成目标，输出一行 "YES"，并在下一行输出两个整数 i 和 j（$1 \le i,j \le n$ 且 $i \ne j$），表示第一次操作选择的下标。

样例

standard input	standard output
3 1 2 4 3 1 1 2 3 2 1 4 1 1 1 3 1 1 1 1	NO YES 1 2 YES 2 1

提示说明

在第一个样例中，只有一种宗派，无论如何安排，双方均不会相互攻击。

在第二个样例中，按顺序执行以下战斗后，玄真国全军覆没：

- $i = 1, j = 2$
- $i = 2, j = 1$
- $i = 3, j = 1$
- $i = 3, j = 1$

因此第一次战斗可以是 $i = 1, j = 2$ 。

Problem E. 稳定区间

输入文件: standard input
输出文件: standard output

给定一个长度为 n 的序列 a 和 q 个询问。

每个询问给出区间 $[l,r]$，你需要判断该区间是否存在一个 **非空** 子序列，其元素总和是 3 的倍数。

具体来说，你需要找到一个长度 $m \geq 1$ 的下标序列 p ，对所有 p_i 满足 $l \leq p_1 < p_2 < \ldots < p_m \leq r$ ，且 $\sum_{i=1}^{m} a_{p_i}$ 是 3 的倍数。

若存在这样的下标序列，则称区间 $[l,r]$ 是稳定的。

输入格式

第一行 2 个整数 n 和 q（$1 \leq n,q \leq 10^5$）

第二行 n 个整数 $a_1,a_2,\ldots,a_n$（$1 \leq a_i \leq 10^9$）

接下来 q 行每行 2 个整数 l,r（$1 \leq l \leq r \leq n$）

输出格式

对于每个询问，输出一行：

- 如果区间是稳定的则输出 “Yes”。
- 否则输出 “No”。

你可以以任意大小写输出 “Yes” 和 “No”（例如，字符串 “yEs”、“yes”、“Yes” 和 “YES” 将被识别为肯定的回答）。

样例

standard input	standard output
5 4 1 4 5 3 6 1 2 2 3 4 4 1 5	No Yes Yes Yes

Problem F. 【模板】Pollard-Rho

输入文件: standard input
输出文件: standard output

这是一道交互题

给定一个正整数 n（$2 \le n \le 2^{60}-1$），输出它最大的质因子。

这个问题太难了，所以你有 64 次尝试的机会，对于每次尝试：

- 你需要输出一行一个整数 m（$0 \le m \le 2^{63}-1$）。
- 交互器会考虑 n 最大的质因子 p，计算 $d = p \oplus m$ 。
- 随后交互器会输出 d 的二进制表示中数位 1 的个数。

在这里，$x \oplus y$ 代表整数 x 和 y 的按位异或运算。

显然，当交互器输出 0 时，说明你输出了正确答案，此时你应停止尝试。

输入格式

一行一个整数 n，保证 $2 \le n \le 2^{60}-1$ 。

交互协议

程序交互应从读取整数 n 开始。

在进行尝试时，请输出一行一个整数 m（$0 \le m \le 2^{63}-1$）。

然后，你需要读入交互器的输出的一个整数，表示 d 的二进制表示中数位 1 的个数。

注意：接收到交互器的输出 0 或尝试次数达到限制后，你的程序应立即终止。

此外，在每次输出完毕后，你都需要立即 **刷新缓冲区**！你可以使用如下语句来刷新缓冲区：

- 对于 C/C++：`fflush(stdout)`
- 对于 C++：`std::cout << std::flush`
- 对于 Java：`System.out.flush()`
- 对于 Python：`stdout.flush()`

对于 C++ 语言，在输出换行时如果你使用 `std::endl` 而不是 '`\n`'，也可以自动刷新缓冲区。

样例

standard input	standard output
21 0	 7
1234567654321 5 0	 1145 4649

提示说明

显然，你只需要编写一个高效率的质因数分解算法即可通过此题。

Problem G. 互异排列—扩展

输入文件: standard input
输出文件: standard output

给定一个 1 到 n 的排列 p，小 H 需要将数字 $n+1$ 插入到 p 的某个位置（包括开头和结尾），得到一个新的排列 q（长度为 $n+1$）。他希望新排列 q 的所有前缀和对 $(n+1)$ 取模的结果 **互不相同** 。

形式化地，定义前缀和数组 S，其中 $S_k = \left(\sum_{j=1}^{k} q_j\right) \bmod (n+1)$，对于 $k = 1, 2, \ldots, n+1$。要求 $S_1, S_2, \ldots, S_{n+1}$ 这 $n+1$ 个值互不相同。

请你帮助小 H 判断是否存在这样的插入位置，如果存在，输出任意一个可行的排列 q，否则输出 -1 。

输入格式

第一行一个整数 T（$1 \le T \le 1000$），表示测试用例数量。

每个测试用例包含两行：

- 第一行一个整数 n（$1 \le n \le 10^6$）
- 第二行 n 个整数 $p_1, p_2, \ldots, p_n$，表示原排列 p

保证所有测试用例的 n 之和不超过 10^6。

输出格式

对于每个测试用例：

- 如果存在满足条件的插入位置，输出一行 $n+1$ 个整数表示排列 q（多个答案输出任意一个）。
- 否则输出一行 -1。

样例

standard input	standard output
2 1 1 2 2 1	2 1 -1

Problem H. 互异排列—生成

输入文件: standard input
输出文件: standard output

在“互异排列—扩展” 问题中，小 H 需要在给定一个排列 p 中找到一个位置插入数字 $n+1$ 得到排列 q，使得前缀和对 $(n+1)$ 取模的结果 **互不相同**。但并不是所有排列 p 都有解。

小 H 想知道，对于给定的 n，是否存在一个 1 到 n 的排列 p，使得 “互异排列—扩展” 问题有解（即 p 存在一个插入位置满足条件）。

请你帮助小 H 判断是否存在这样的排列 p，如果存在，输出任意一个可行的排列 p，否则输出 -1。

输入格式

第一行一个整数 T（$1 \le T \le 1000$），表示测试用例数量。

每个测试用例包含一行一个整数 n（$1 \le n \le 10^6$）。

保证所有测试用例的 n 之和不超过 10^6。

输出格式

对于每个测试用例：

- 如果存在满足条件的排列 p，输出一行 n 个整数表示排列 p（多个答案输出任意一个）。
- 否则输出一行 -1。

样例

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

Problem I. 2-冲突数对

输入文件: standard input
输出文件: standard output

给定一个大小为 n 的数组 $[a_1, a_2, \ldots, a_n]$ 。

定义满足以下条件的有序下标对 (i, j) 称为 “2- 冲突数对” ：

- $1 \leq i < j \leq n$
- $a_i \neq a_j$

冲突数对 (i_1, j_1) 和 (i_2, j_2) 不同，当且仅当 $i_1 \neq i_2$ 或 $j_1 \neq j_2$ 。

你需要求出有多少种不同的 “2-冲突数对” 。

输入格式

第一行一个整数 n（$1 \leq n \leq 10^5$），表示序列长度。

第二行 n 个整数 $a_1, a_2, \ldots, a_n$（$1 \leq a_i \leq 10^5$），表示序列中的元素。

输出格式

输出一行一个整数，表示有多少种不同的 “2-冲突数对” 。

样例

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

Problem J. 收藏宝石

输入文件: standard input
输出文件: standard output

小 L 是一名魔法师，他收藏了 n 个宝石。为了方便展示，他用 m 条绳索将这些宝石与房梁相互连接。其中，第 i 个宝石可以视为第 i 个节点，而房梁是一个编号为 0 的特殊节点。

当小 L 需要使用某个宝石时，他会选择一个宝石，并 **切断所有与该宝石相连的绳索** 。

若此操作恰好使得 **仅有一个宝石** 失去与房梁的连接（即被取下），则称这次操作是 **合法的** ，且该宝石被取下。

每个宝石都有一个自己的价值 a_i。小 L 想知道， **以当前的绳索连接方式，是否存在一种取下顺序** ，能使他每次都能合法取下 **当前剩余宝石中价值最大的宝石之一** 。

输入格式

第一行包含三个整数 n, m, k $(1 \leq n, m, k \leq 10^5)$，分别表示宝石数量和绳索数量，以及宝石价值的上界。

第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq k)$，其中 a_i 表示第 i 个宝石的价值。

接下来 m 行，每行包含两个整数 x, y $(0 \leq x, y \leq n)$，表示第 x 个节点与第 y 个节点之间有一条绳索连接。特殊地，第 0 号节点表示 **房梁** 而不是宝石。

保证所有宝石均与房梁连通，保证任意两个节点之间至多只有一条绳索连接，没有绳索连接同一个节点。

输出格式

如果符合要求的合法取下顺序，输出 “YES”，否则输出 “NO” 。

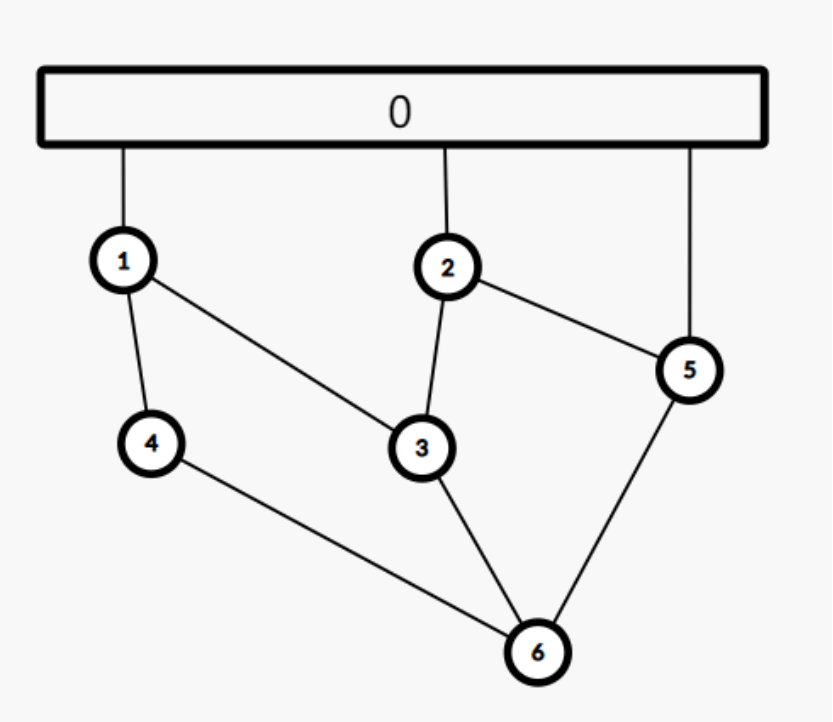
你可以以任意大小写输出 “YES” 和 “NO”（例如，字符串 “yEs”、“yes”、“Yes” 和 “YES” 将被识别为肯定的回答）。

样例

standard input	standard output
2 3 10 10 1 0 1 1 2 2 0	YES
2 2 10 10 1 0 1 1 2	NO
6 10 3 1 3 2 2 2 3 0 1 0 2 0 5 1 3 1 4 2 3 2 5 3 6 4 6 5 6	YES

提示说明

测试样例三的图如下所示：


0
1
2
3
4
5
6