

“国教 AC 杯”大学生程序设计竞赛

环海岸线联盟联合校赛



国际教育学院
SCHOOL OF INTERNATIONAL EDUCATION

问题列表

A	AK IOI
B	管道贯通
C	懒狗归雨
D	右手持盾左手持剑
E	通风管运输任务
F	王老师的烦恼
G	钓鱼大法
H	Fast Mex Transform(FMT)
I	GCD AND LCM
J	README.md
K	柴郡猫的挑战
L	寻找迷路的 T
M	最性价比的作业

November 30, 2025 13:00^{UTC+8}

时长：5 小时 | 13 题

在比赛正式开始前请勿翻阅

DO NOT OPEN THIS PROBLEM SET

UNTIL TOLD TO DO SO

Problem A. AK IOI

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

某测评平台为了给参赛者提供专业的竞赛体验，开发者需要实现一个实时排名系统。比赛遵循 IOI 计分规则。你的任务是编写一个程序，处理所有的提交记录并生成最终的官方排名。

比赛共有 N 位选手（编号从 1 到 N ）和 M 道题目（编号从 1 到 M ）。在整个比赛过程中，系统共产生 K 条提交记录。

计分规则如下：

- 对于每道题目，一位选手的得分，是在该项目的所有提交记录中所获得的最高分。如果选手从未提交过某道题目，则该题得分为 0。
- 一位选手的总分，是其在所有 M 道题目上得分的总和。

排名依据以下标准确定：

- 选手首先按照总分从高到低（降序）进行排名。
- 如果两位或多位选手的总分相同，则按照他们的选手 ID 从小到大（升序）进行排名。

此外，你需要找出所有“完全解出”（All-Killed，简称 AK）的选手。一位选手被视为 AK，当且仅当他/她在从 1 到 M 的每一道题目上都获得了 100 分的满分。

你的任务是处理 K 条提交记录，生成从第 1 名到第 N 名的完整榜单，并列出了所有 AK 的选手。

Input

输入第一行包含三个由空格分隔的整数 N ($1 \leq N \leq 10^3$)、 M ($1 \leq M \leq 10$) 和 K ($1 \leq K \leq 10^5$)，分别表示选手的数量，题目的数量，以及提交记录的总数。

接下来的 K 行，每行描述一条提交记录，包含三个由空格分隔的整数 c_{id} ($1 \leq c_{id} \leq N$)、 p_{id} ($1 \leq p_{id} \leq M$) 和 $score$ ($0 \leq score \leq 100$)，分别表示提交该记录的选手 ID，提交的题目 ID，以及本次提交获得的分数。

Output

首先，输出 N 行，代表最终的排名榜单。每行的格式应为：

‘排名 选手ID 总分 题目1得分 题目2得分 ... 题目M得分’。

同一行中的所有数值必须由单个空格分隔。

在输出 N 行榜单之后，如果有选手达成了 AK，则额外输出一行。该行应包含所有 AK 选手的 ID，按 ID 升序排列，并由单个空格分隔。如果没有选手 AK，则不应输出此行。

Example

standard input	standard output
4 2 7	1 2 200 100 100
2 1 100	2 1 190 100 90
1 1 100	3 3 90 90 0
3 1 90	4 4 0 0 0
1 2 90	2
2 2 80	
2 2 100	
3 2 0	

Problem B. 管道贯通

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

归雨偷懒被老师发现了！老师超级无敌生气震怒，责令他立即贯通全校堵塞的管道，否则老师将会让归雨知道什么叫残忍。

归雨从老师手中接过一根特殊的管道贯通器——该设备能将数字序列连接成整体并通过一端爆发来疏通管道。然而，由于贯通器本身是管道结构，归雨每次插入数字时，只能选择当前管道的左端或右端进行操作。

归雨有 n 个个位数字（范围 $0 - 9$ ），分别为 a_1, a_2, \dots, a_n 。他必须严格按照给定顺序依次插入这些数字（即先插入 a_1 ，再插入 a_2 ，依此类推）。每次插入时，他可以选择将新数字添加到当前数字序列的左端或右端。

归雨的目标是使最终形成的 n 位数字的数值最大（例如，序列 $[3, 2, 1]$ 对应数值 321 ）。

Input

第一行输入一个整数 T ($1 \leq T \leq 10^5$)，表示测试用例的数量。

接下来 T 组测试用例，每组格式如下：

第一行输入一个整数 n ($1 \leq n \leq 10^6$)。

第二行输入 n 个整数 a_i ($0 \leq a_i \leq 9$)，表示按顺序需要插入的数字。

* 数据保证所有测试用例的 n 之和不超过 10^6 ，并且不会出现全零情况。

Output

对于每个测试用例，输出一行：一个长度为 n 的字符串，表示归雨能形成的最大数值。

Example

standard input	standard output
3	321
3	991
1 2 3	4312
3	
9 1 9	
4	
1 3 2 4	

Note

在样例中：

第一组：依次在左端插入得到 321 。

第二组：先插入 9 ，然后在右端插入 1 得到 91 ，最后在左端插入 9 得到 991 。

第三组：通过合理的左右端插入策略得到 4312 。

Problem C. 懒狗归雨

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

归雨是个懒狗，有一天，老师实在看下去他的懒惰，决定给他布置一个特殊的作业。老师承诺：如果归雨能正确完成作业，就会给他奖励；但如果做错了，就要接受惩罚。

老师给了归雨一个长度为 n 的数组 a ，然后会快速进行 q 次查询。每次查询给出两个整数 l 和 r ，要求计算区间 $[l, r]$ 内所有元素的按位与结果：

$$a_l \& a_{l+1} \& \dots \& a_r$$

其中 $\&$ 表示按位与运算。

归雨看了一眼觉得这太简单了，于是他又犯懒了！现在他请求你帮忙完成这项作业。如果你做对了，就能 AC 这道题！

Input

第一行包含两个整数 n 和 q ($1 \leq n, q \leq 2 \times 10^5$)，分别表示数组长度和查询次数。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$)，表示数组元素。

接下来 q 行，每行包含两个整数 l 和 r ($1 \leq l \leq r \leq n$)，表示一次查询的区间。

Output

对于每次查询，输出一个整数，表示区间按位与的结果。每个查询结果独占一行。

Example

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

Note

按位与运算（Bitwise AND），通常记作 $\&$ ，是对两个整数在二进制表示下进行的逐位逻辑运算。

对于两个二进制位 x 和 y ，运算规则如下：

- 若 $x = 1$ 且 $y = 1$ ，则 $x \& y = 1$ ；
- 否则 $x \& y = 0$ 。

对于多个整数的按位与运算，结果的某一位为 1，当且仅当参与运算的所有整数在该位上均为 1。

Problem D. 右手持盾左手持剑

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 256 megabytes

"城之内，你的生命值犹如风中残烛！—— 某社长

城之内被卷入了一场黑暗决斗。在与对手纠缠数回合后，对手召唤了 N 只强力怪兽，并破坏了城之内场上所有的防御手段。

现在的局势十分危急：

1. 对手场上有 N 只怪兽，它们将按照从第 1 只到第 N 只的顺序，依次对城之内发动直接攻击。
2. 每只怪兽拥有两个属性：攻击力 A 和 守备力 D 。在通常情况下，怪兽的攻击造成等于其攻击力 A 的伤害。

城之内手里没有手牌，但在场上盖放了一张速攻魔法卡——「右手に盾を左手にを」（右手持盾左手持剑）。卡片效果：这张卡发动后，直到回合结束前，场上所有怪兽的原本攻击力和原本守备力互换。

城之内可以在任意一只怪兽进行攻击宣言时发动这张卡（也可以选择整场决斗都不发动）。

- 如果在第 i 只怪兽攻击时发动：
 - 第 1 到第 $i - 1$ 只怪兽已经攻击完毕，造成攻击力 A 的伤害。
 - 第 i 只怪兽及之后的所有怪兽（第 i 到 N 只），将造成守备力 D 的伤害。

为了不输掉决斗，请你帮助城之内计算：通过合理选择发动时机（或不发动），他受到的总伤害最小值是多少？

Input

第一行输入一个整数 T ($0 \leq T \leq 10^4$)，表示测试数据组数。

对于每组测试数据：

第一行输入一个整数 N ($1 \leq N \leq 10^5$)，表示对手场上怪兽的数量。

接下来 N 行，每行包含两个整数 A 和 D ($0 \leq A, D \leq 10^8$)，分别表示怪兽的攻击力和守备力。

注意：答案可能超过 32 位整数的范围，请使用合适的数据类型。

Output

输出 T 行，每一行包含一个整数，表示该轮决斗中城之内可能受到的最小伤害总和。

Example

standard input	standard output
2	900
5	500
100 300	
200 400	
500 100	
800 100	
200 400	
2	
200 400	
300 600	

Note

第一组数据:

对手有 5 只怪兽。

- 如果全程不发动: $100 + 200 + 500 + 800 + 200 = 1800$
- 如果在第 3 只怪兽攻击时发动:
 - 第 1、2 只怪兽造成攻击力伤害: $100 + 200 = 300$
 - 此时发动魔法卡, 攻守互换。
 - 第 3、4、5 只怪兽造成守备力伤害: $100 + 100 + 400 = 600$
 - 总伤害: $300 + 600 = 900$ 。这是所有方案中的最小值。

第二组数据:

怪兽属性为 (200, 400) 和 (300, 600)。

由于每只怪兽的守备力都比攻击力高, 发动魔法卡反而会增加伤害。

因此城之内选择不发动这张卡, 承受 $200 + 300 = 500$

Problem E. 通风管运输任务

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

看看要给沿高速公路的一系列服务区运输一批通风管。他驾驶的是一辆电动车，行驶途中需要注意电量问题。共有 n 个服务区（编号为 1 到 n ），部分服务区配备充电桩。高速上共有 m 条双向道路，每条道路连接两个服务区并具有一定长度。已知车辆每次充满电后最多可行驶 D 的距离。

看看想知道：能否从起点服务区出发到达终点服务区；如果可以，在途中最少需要充几次电？

注意：出发时电量为满电，若起点本身有充电桩，不计入充电次数。

Input

第一行包含三个整数 n, m, D ($1 \leq n \leq 1000, 1 \leq m \leq 5000, 1 \leq D \leq 10^9$)。

接下来 m 行，每行包含三个整数 u, v, w ($1 \leq w \leq 10^9$)，表示服务区 u 与 v 之间有一条长度为 w 的双向道路。

接下来一行包含 n 个整数 a_1, a_2, \dots, a_n 。若 $a_i = 1$ ，表示第 i 个服务区有充电桩；若 $a_i = 0$ ，表示没有充电桩。

最后一行包含两个整数 s, t ($1 \leq s, t \leq n$)，表示起点和终点服务区的编号。

道路为双向，无自环，可有重边。

Output

输出一个整数：若无法到达终点，输出 -1 ；否则输出从起点到终点最少需要充电的次数。

Examples

standard input	standard output
5 6 10 1 2 6 2 3 5 3 4 3 4 5 8 2 5 12 1 3 9 0 1 0 1 0 1 5	2
4 3 5 1 2 4 2 3 6 3 4 4 1 0 1 0 1 4	-1

Note

对于第一组样例：从 1 出发，初始电量满。路径为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ，需在 2 号点，4 号点各充电一次，共两次

对于第二组样例：从 1 出发可到达 2，但 2 没有充电桩，且无法在电量 5 内到达 3。因此无法抵达终点。

Problem F. 王老师的烦恼

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

新学期伊始, "程序设计入门"课程的王老师正在批改期末大作业。王老师并非一次性批改完所有作业, 而是每批改完一份, 就立即将成绩录入系统。

教务主任非常关心班级的成绩分布, 他会不时走进王老师的办公室, 询问当前已录入成绩的中位数。由于录入过程是动态的, 王老师每次都需要重新整理所有成绩来回答问题, 这让他感到非常烦恼。

在此问题中, 中位数的定义如下: 假设当前已录入 N 份作业的成绩, 将这些成绩按照非降序 (从低到高) 排列后, 处于第 $\lfloor \frac{N+1}{2} \rfloor$ 个位置的成绩即为中位数 (其中 $\lfloor x \rfloor$ 表示对 x 向下取整)。

例如:

- 若成绩为 $\{80, 95, 60\}$, 排序后为 $\{60, 80, 95\}$ 。 $N = 3$, 中位数在第 $\lfloor \frac{3+1}{2} \rfloor = 2$ 个位置, 即 80。
- 若成绩为 $\{100, 75, 85, 60\}$, 排序后为 $\{60, 75, 85, 100\}$ 。 $N = 4$, 中位数在第 $\lfloor \frac{4+1}{2} \rfloor = 2$ 个位置, 即 75。

形式化地:

维护一个初始为空的多重集合 S 。你需要支持 Q 次操作, 操作分为以下两种:

- 1 x : 将整数 x 加入多重集合 S 中。
- 2: 查询当前多重集合 S 的中位数。

对于一个大小为 N 的多重集合 S , 将其元素从小到大排序为 a_1, a_2, \dots, a_N , 则其中位数为 a_k , 其中 $k = \lfloor \frac{N+1}{2} \rfloor$ 。

现在, 请你编写一个程序, 帮助王老师自动完成成绩的录入和中位数的查询。

Input

第一行输入一个整数 Q ($1 \leq Q \leq 2 \times 10^5$), 表示王老师的操作总次数。

接下来的 Q 行, 每行描述一个操作, 格式如下:

- 1 x : 表示王老师录入了一个成绩 x ($0 \leq x \leq 10^9$)。
- 2: 表示教务主任发起了一次查询, 询问当前所有成绩的中位数。

数据保证在进行查询操作 (即操作 2) 时, 系统中至少已录入一个成绩。

Output

对于每一次查询操作, 输出一行一个整数, 表示当前成绩的中位数。

Examples

standard input	standard output
10 1 10 2 1 20 2 1 5 2 1 15 2 1 12 2	10 10 10 10 12
20 1 26 1 83 1 830 1 190 1 142 2 1 320 1 218 2 2 1 563 2 2 2 1 632 2 1 877 1 715 2 2	142 190 190 190 190 190 218 320 320

Problem G. 钓鱼大法

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

鱼塘里有 n 条鱼，每条鱼都有各自的重量。

桂鱼的价值是鲨鱼的 2 倍。爱音是鱼塘主，这天狐狸来找爱音比赛钓鱼。

爱音知道自己比不过狐狸，于是提出了一个新规则：爱音先钓，然后狐狸再钓。

爱音可以在狐狸钓鱼时耍赖，把狐狸的鱼竿搞坏。

如果爱音钓到的鱼价值等于鱼塘里剩下的鱼价值，爱音就胜利。

但是，爱音钓完鱼后，鱼塘就要低价转让给通风管了。

有个神秘人自称勿忘，他能提前预测钓鱼的顺序，但是爱音花钱了，所以他只告诉爱音一人。

爱音希望既能胜利，又能让自己钓到的鱼价值最多。假设爱音绝顶聪明，狐狸只能获得价值为多少的鱼？

具体来说，钓鱼的顺序是固定的。爱音会钓走序列最开始的一段鱼（可以为空），狐狸会钓走紧接着的一段鱼（可以为空），剩下的鱼即为鱼塘里剩下的鱼。

我们要求在满足爱音钓到的鱼的总价值 (sum_1) 等于鱼塘里剩下的鱼的总价值 (sum_3) 的前提下，狐狸钓到的鱼的总价值 (sum_2) 的最小值。

形式化地：

给定长度为 n 的整数序列 d_1, d_2, \dots, d_n ，长度为 n 的字符串 s ，以及一个常数 k 。

定义第 i 条鱼的价值 v_i 如下：

$$v_i = \begin{cases} d_i \times k & \text{if } s_i = 'S' \\ 2 \times d_i \times k & \text{if } s_i = 'G' \end{cases}$$

你需要选择两个整数 x, y 满足 $0 \leq x \leq y \leq n$ ，将序列划分为三部分，其价值和分别为：

$$sum_1 = \sum_{i=1}^x v_i, \quad sum_2 = \sum_{i=x+1}^y v_i, \quad sum_3 = \sum_{i=y+1}^n v_i$$

（注：若上限小于下限，则和为 0）。

请在满足 $sum_1 = sum_3$ 的条件下，最小化 sum_2 的值。

Input

第一行输入两个整数 n, k ($1 \leq n < k \leq 2 \times 10^5$)，分别表示鱼塘里有多少条鱼，和今日鱼的价格。

第二行输入 n 个整数 d_1, d_2, \dots, d_n ($1 \leq d_i \leq 10^9$)，表示鱼塘里鱼的重量。

第三行输入一个长度为 n 的字符串 s 。如果 s_i 为 'S' 则代表第 i 条鱼为鲨鱼，反之为桂鱼。

Output

输出一个整数，表示在满足 $sum_1 = sum_3$ 的条件下， sum_2 能达到的最小值。

答案对 $10^9 + 7$ 取模。

Example

standard input	standard output
5 7 1 1 1 1 1 GGGGG	14

Problem H. Fast Mex Transform(FMT)

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

我们发现了一种在 $O(\log n)$ 时间复杂度下查询有修改操作的区间 mex 的算法:

首先有修改操作的区间 \min 可以用线段树等数据结构在 $O(\log n)$ 时间复杂度下完成, 那么我们有 $mex = \frac{\min \times mex}{\min}$, 现在已知 \min , 那么我们想办法知道 $\min \times mex$ 的值即可。

现在给你一个数组, 支持两个操作: 区间修改, 把一个区间内的所有数都修改为 x , 或者询问区间内的 \min 和 mex 的乘积。

\min 指给定集合内数的最小值, 如 $\min(1, 2, 4, 6) = 1$ 。 mex 指给定集合内最小的没有出现的自然数, 如 $mex(0, 1) = 2$, $mex(1, 1, 4, 5, 1, 4) = 0$ 。

Input

第一行两个整数 n, m ($1 \leq n, m \leq 10^5$), 表示数组内数的个数, 以及操作的个数。

第二行 n 个整数, 表示数组内的元素 ($0 \leq a_i \leq 10^5$)。

以下 m 行每行 3 或 4 个整数, 第一个整数 op ($1 \leq op \leq 2$) 表示操作类型, 若 $op = 1$, 则接下来输入 3 个整数 l, r, x ($1 \leq l, r \leq n, 0 \leq x \leq 10^5$), 表示把区间 $[l, r]$ 内的数全部修改为 x , 若 $op = 2$, 则输入 2 个整数 l, r , 表示询问区间 $[l, r]$ 内的 \min 值与 mex 值的乘积。

Output

对于每个 $op = 2$ 操作, 输出一行一个整数, 表示答案。

Example

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

Note

对于样例, 数据初始是 $[1, 1, 4, 5, 1, 4]$, 第一次操作后, 数据变为 $[1, 1, 1, 1, 1, 1]$, 然后询问 $[3, 4]$ 内的 \min 和 mex , 则 $\min(1, 1) = 1, mex(1, 1) = 0, 1 \times 0 = 0$ 。

Problem I. GCD AND LCM

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

定义两个数的 gcd 和 lcm 分别为两个数的最大公约数和最小公倍数。

其中 $\text{gcd}(a, b)$ 表示最大的可以同时整除 a, b 的数, $\text{lcm}(a, b)$ 表示最小的可以同时被 a, b 整除的数。例如, $\text{gcd}(4, 6) = 2$, $\text{lcm}(4, 6) = 12$ 。

接下来, 给定数组 a , 你需要计算: 对于所有 $1 \leq i, j \leq n$, $\text{gcd}(a_i, a_j) \times \text{lcm}(a_i, a_j)$ 的和。

Input

第一行输入一个整数 n ($1 \leq n \leq 10^5$), 表示数组元素个数。

接下来一行输入 n 个整数, 表示数组内的元素 a_i ($1 \leq a_i \leq 10^4$)。

Output

输出一行一个整数, 表示答案。

Examples

standard input	standard output
1 1	1
3 1 3 8	144

Note

对于样例 1, 显然有 $\text{gcd}(1, 1) = 1, \text{lcm}(1, 1) = 1, 1 \times 1 = 1$, 所以答案为 1。

请注意计算中可能出现的溢出问题。

Problem J. README.md

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

为促进高校间程序设计能力的交流与提升，我们计划举办环海岸线联盟联合校赛 - 大学生程序设计竞赛。本次竞赛完全由学生自发组织，不代表任何官方机构或组织，旨在为热爱编程的大学生提供一个展示才华、切磋技术的平台。

本次联合校赛是各校独立的程序设计竞赛整合而成。我们通过 "统一时间、统一赛题"的模式，为各校同学提供一个更广阔的平台。这是一个纯粹由学生自发组织的联盟，真诚欢迎更多高校加入，共同成长。

竞赛详情

- 赛事形式：采用国际大学生程序设计竞赛 (ICPC) 赛制，团队形式参赛，每队最多 3 人。
- 参与高校 (按拼音首字母排序):
 - 福建商学院
 - 福州工商学院
 - 福州外语外贸学院
 - 龙岩学院
 - 宁德师范学院
 - 泉州信息工程学院
 - 阳光学院
 - 仰恩大学
- 竞赛时间：2025 年 11 月 30 日 13:00 - 18:00。

竞赛日程

- 12:30 - 12:50：选手签到入场
- 12:50 - 13:00：宣读竞赛规则
- 13:00：比赛正式开始
- 18:00：比赛正式结束
- 18:00 - 19:30：休息
- 19:30 - 20:30：赛后讲解与颁奖 (各校自行安排)

竞赛规则

为保证竞赛公平，所有参赛选手必须遵守以下规则：

1. 可携带物品：允许携带任意纸质资料 (如书籍、打印稿、笔记)、草稿纸、饮品及零食。所有入场物品需接受志愿者检查。
2. 电子设备：竞赛期间，必须关闭并妥善保管所有电子通讯设备 (如手机、智能手表等)。
3. 计算机使用：

- (a) 仅允许使用指定的竞赛平台 (牛客) 和编程环境 (编程环境如需定制请联系校赛负责人)。
 - (b) 禁止访问任何其他网页、软件或使用人工智能辅助工具。
 - (c) 禁止使用任何外部存储设备。
 - (d) 不得关闭赛场指定的监控软件。
4. 交流规范: 允许队内成员之间交流。禁止与任何其他队伍或场外人员进行任何形式的交流。
5. 封榜机制: 为了增加比赛的悬念与竞技性, 比赛结束前一小时 (即 17:00 至 18:00) 将进入封榜阶段。在此期间, 排行榜将不再实时更新各队伍的题目情况与排名 (即无法看到其他队伍在最后 1 小时的提交结果), 但参赛队伍仍可正常提交代码, 并获得实时的评测结果反馈 (AC、WA、TLE 等)。最终排名将在赛后的颁奖环节揭晓。

违反以上任何规则, 将被立即取消比赛资格。

排名规则

竞赛排名遵循 ICPC 国际标准:

1. 按解题数量排名, 解题数多者排名靠前。
2. 若解题数相同, 则按总罚时排名, 罚时少者排名靠前。
3. 总罚时 = \sum (每道通过题目的解答用时) + \sum (每道通过题目的赛前错误提交次数 \times 20 分钟)。

出题情况 (按校名拼音首字母排序)

- 福建商学院: D
- 福州工商学院: G
- 福州外语外贸学院: B, C, K
- 宁德师范学院: H, I
- 泉州信息工程学院: A, F, J, L, M
- 阳光学院: E

测评环境

本次竞赛支持以下主流编程语言及对应版本:

- C/C++: G++ 13, Clang++ 18, GCC 10
- Java: Java JDK 21
- Python: Python 3.10, PyPy3

祝你好运!

Input

本题没有输入。

Output

请输出一行字符串: "Hello, Coastal Alliance!"(不包含引号)。

Problem K. 柴郡猫的挑战

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

"如果你不知道自己去哪里，那么走哪条路都行。——柴郡猫

归雨十年如一日地寻找着爱丽丝，终于在一次探索中掉入了不可思议之国。在迷雾笼罩的森林里，他遇见了神秘的柴郡猫。柴郡猫告诉他：

"回答我的问题，我就带你去找爱丽丝。"

对于从 1 到 N 的每一个整数 x ，找出所有满足以下条件的有序正整数对 (a, b) ：

1. $a + b = x$;
2. $a \times b$ 是 3 的倍数.

注意： (a, b) 和 (b, a) 在 $a \neq b$ 时被视为不同的配对。

令 $C(N)$ 为对从 1 到 N 的所有 x 找到的满足条件的配对总数。给定一个 N ，请你判断 $C(N)$ 是奇数还是偶数。

Input

第一行包含一个整数 T ($1 \leq T \leq 10^5$)，表示测试用例的数量。

接下来 T 行，每行包含一个整数 N ($5 \leq N \leq 10^{18}$)。

Output

对于每个测试用例，输出一行：如果 $C(N)$ 是偶数，输出 YES；否则输出 NO。

Example

standard input	standard output
7	NO
41420	YES
89741	YES
59629	NO
51850	YES
336	YES
5678	YES
87662	

Problem L. 寻找迷路的 T

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 32 megabytes

在守望先锋的一个对局中，一支队伍正在集结。一支标准的队伍由若干"奶位-C 位"配对组成，每个配对中的奶和 C 拥有完全相同的"能量编号"，以便协同作战。然而，今天队伍里可能混进了一个迷路的"T 位"。T 位独来独往，不需要配对。现在给你一份队伍所有成员的能量编号列表，请你找出那个迷路的 T。如果队伍的构成是完美的"奶 C"配对，没有任何 T，那么这支队伍就是 "Correct"。

形式化地：

给定一个包含 N 个正整数的序列 A 。该序列具有以下特性：序列中至多只有一个元素恰好出现一次，而其他所有元素都恰好出现两次。你的任务是找出那个只出现一次的元素。如果序列中不存在这样的元素（即所有元素都恰好出现两次），则报告该序列是符合配对规则的（Correct）。

Input

第一行的输入一个整数 N ($1 \leq N \leq 2 \times 10^7$)，表示队伍的总人数。

第二行输入 N 个正整数 ID_i ($1 \leq ID_i \leq 10^{18}$)，代表每个成员的能量编号。

Output

如果找到了迷路的 T ，输出他/她的能量编号。

如果没有 T （即队伍是完美的配对），则输出 "Correct"。

Examples

standard input	standard output
5 101 202 303 101 202	303
4 101 202 101 202	Correct

Problem M. 最性价比的作业

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

你有 n 个拖了很久的作业需要完成。第 i 个作业的初始难度为 a_i 。
从今天开始, 你每天会经历以下两个阶段:

1. 选择与攻克: 你必须选择一个尚未完成的作业并完成它。如果该作业当前的难度大于你的精力 c , 你就必须喝一杯咖啡来"爆肝" (花费 1 枚金币)。如果难度不大于 c , 你则可以轻松搞定 (花费 0 枚金币)。
2. 拖延的代价: 每当你完成一个作业后, 所有剩下的作业难度都会翻倍, 因为拖延会使问题变得更复杂。

你的目标是, 用最少的金币 (也就是喝最少的咖啡) 来完成所有 n 个作业。请问最少需要花费多少金币?

形式化地:

给定一个长度为 n 的数组 a 和一个常数 c 。你需要确定一个 $1, 2, \dots, n$ 的排列 p , 使得以下式子的值最小:

$$\sum_{i=1}^n [a_{p_i} \cdot 2^{i-1} > c]$$

其中 $[P]$ 为艾弗森括号, 当条件 P 为真时值为 1, 否则为 0。

Input

第一行包含两个整数 n 和 c ($1 \leq n \leq 30, 1 \leq c \leq 10^9$), 分别代表作业数量和你的精力上限。
接下来一行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), 代表每个作业的初始难度。

Output

输出一个整数, 表示完成所有作业所需的最少金币数。

Examples

standard input	standard output
5 10 10 4 15 1 8	2
3 42 1000000000 1000000000 1000000000	3
10 30 29 25 2 12 15 42 14 6 16 9	6
10 1000000 1 1 1 1 1 1 1 1 1 864026633	1