

# 2024年CUGACM入队选拔Statement

2024/11/02



## Problems

|   |         |
|---|---------|
| A | 数组      |
| B | 借阅图书    |
| C | 文神的序列   |
| D | 低谷      |
| E | 伤害最大化   |
| F | 猫猫虫困境II |
| G | 奥运会     |
| H | 文神的比赛   |
| I | 智力博弈    |
| J | 星穹列车    |
| K | 冰红茶     |
| L | 数组交换    |
| M | 猜猜看     |

本试题册共13题，15页。

如果您的试题册缺少页面，请立即通知志愿者。

(比赛开始前请勿翻阅)

## Problem A. 数组

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

在计算机科学中，数组是一种常用的数据结构。数组的一个重要特性是其下标（索引）通常从 0 开始。对于一个长度为  $n$  的数组，其元素下标范围是从 0 到  $n - 1$ 。

作为刚入门 C++ 编程的新生，lzt 需要完成一个简单的任务来巩固对数组下标的理解。具体任务如下：

给定一个整数  $n$ ，请输出  $n - 1$ 。这个操作模拟了获取数组最后一个元素的下标。

### 输入格式

共一行。第一行包含一个整数  $n$  ( $1 \leq n \leq 100$ )，表示数组长度。

### 输出格式

输出一个整数，表示这个数组最后一个元素的下标。

### 样例

| standard input | standard output |
|----------------|-----------------|
| 43             | 42              |

## Problem B. 借阅图书

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

图书馆新进了一批特别火的图书,吸引了一大批读者借用。lzt作为图书管理员,想要维护借书秩序。于是他先把这 $n$ 本书依次标号为 $1, 2, \dots, n$ , 然后找到一个超级大箱子, 把这些书按顺序放在箱子里面(序号为 $n$ 的图书在箱子顶部), 之后他打算对这些书进行图书管理, 并向读者公布借用规则, 规则如下:

- 开放预约制。若此时读者没有正在借阅的图书, 则该读者可以预约 $d$ 天后进行借阅, 但同时图书馆无法保证读者能在 $d$ 天后能借到书。同时, 该读者只能等到 $d$ 天之后进行下一次预约。
- 在每一天的一开始(即在当天进行操作之前), 图书馆会从箱子顶端依次取出图书来试图满足之前预约过当天借阅的读者的需求。越早预约的读者越先被分配到, 并且给分配到图书的读者的状态变为“正在借阅”。若在此过程中lzt发现箱子里的图书空了, 则之后预约在当天的读者将不会分配到图书, 这些预约也会作废。
- 可以随借随取。只要箱子里的图书不空, 并且这个读者此时没有预约该天之后的图书, 同时也没有正在借阅的图书, 则请求立即借阅的读者将立即取得箱子顶部的图书。
- 读者归还图书后, lzt会把该图书放到箱子顶端。
- 每个读者同一时间最多只能借阅一本书。

此外, 每个读者都有一个唯一的 **读者ID**。图书馆需要支持查询特定读者当前借阅的图书编号。

现在, lzt 需要你维护图书馆的借阅过程。

### 输入格式

第一行包含两个整数  $n, m (1 \leq n, m \leq 1000)$ , 分别表示图书的数量和操作的数量。

接下来  $m$  行, 每行描述一个操作, 操作的类型有四种:

1. "t RESERVE id d": 表示第  $t$  天读者  $id$  预约在当前操作所在的第  $d (1 \leq d \leq 1000)$  天后借书。如果读者  $id$  已经有未归还的图书, 或者之前已经进行过预约, 则此次预约无效, 忽略该操作。
2. "t BORROW id": 表示第  $t$  天读者  $id$  请求立即借书。
3. "t RETURN id": 表示第  $t$  天读者  $id$  归还所借的图书。如果读者  $id$  没有借阅任何图书, 则忽略该操作。
4. "t QUERY id": 表示第  $t$  天查询读者  $id$  当前借阅的图书编号。

其中:

- 操作按照时间顺序依次进行。
- 多个操作可以发生在同一天。输入中每个操作前会指定其执行的天数。

- 天数从 1 开始(但第一天不一定有操作), 且操作按非递减的天数顺序给出。
- 每个读者的  $id(1 \leq id \leq 1000)$  是一个唯一的正整数。
- $t(1 \leq t \leq 1000)$  是一个整数, 表示该操作发生时所在天数。

## 输出格式

- 对于每一个 "RESERVE" 操作, 输出一行一个整数。若操作成功, 则输出1, 否则输出0。
- 对于每一个 "BORROW" 操作, 输出一行一个整数, 表示被借出的图书编号。如果借书请求无效或箱子为空, 则输出 0。
- 对于每一个 "RETURN" 操作, 输出一行一个整数, 表示归还的图书编号。如果归还操作无效 (该用户没有正在借阅的书), 则输出 0。
- 对于每一个 "QUERY" 操作, 输出一行一个整数, 表示读者当前借阅的图书编号。如果读者未借阅任何图书, 则输出 0。

## 样例

| standard input | standard output |
|----------------|-----------------|
| 2 10           | 1               |
| 1 RESERVE 1 2  | 0               |
| 2 BORROW 1     | 0               |
| 2 RESERVE 1 3  | 2               |
| 2 BORROW 2     | 1               |
| 2 RESERVE 3 4  | 0               |
| 4 BORROW 3     | 1               |
| 5 RETURN 1     | 2               |
| 5 QUERY 2      | 1               |
| 6 QUERY 3      | 0               |
| 6 QUERY 1      |                 |

## 样例解释

第 1 天, 读者 1 预约第 3 天借书, 预约操作有效, 输出 1。第 2 天, 读者 1 尝试立即借书。由于读者 1 已经预约了第三天的借书操作, 所以立即借书的请求无效, 输出 0。随后, 读者 1 尝试预约借书, 但是由于读者 1 此时已经有预约了, 因此当前预约无效, 输出 0。之后, 读者 2 尝试立即借书。此时箱子顶部有书且读者 2 没有预约和正在借阅的书, 所以读者 2 成功借走编号为 2 的书, 输出 2。第 2 天最后, 读者 3 预约第 6 天借书, 预约有效。第 3 天一开始, 读者 1 拿到了之前所预约的书, 编号为 1。第 4 天, 读者 3 尝试立即借书。箱子此时空了。由于没有可供借阅的书, 读者 3 的请求无效, 输出 0。第 5 天, 读者 1 归还了编号为 1 的书, lzt 把这本书放回箱子顶部, 输出这本书的编号 1。随后, lzt 查询读者 2 当前借阅的图书编号。读者 2 之前借阅了编号为 2 的书, 且尚未归还, 输出 2。第 6 天一开始, 读者 3 拿到了之前所预约的书, 编号为 1。之后, lzt 分别查询读者 3 当前借阅的图书编号, 读者 3 输出其书的编号 1。最后查询读者 1 当前借阅的图书编号, 由于此时读者 1 没有正在借书, 因此输出 0。

Problem C. 文神的序列

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

文神有一个长度为  $n$  的序列  $a$ ，所有元素均为正整数，她可以对序列中的元素进行操作。在每次操作中，她都会选择两个数字  $a_i$  和  $a_j$  ( $i \neq j$  且  $1 \leq i, j \leq n$ )，将  $a_i$  改为  $a_i \& a_j$ 。文神想知道经过至多  $n$  次操作后，是否能将序列中的所有数字改为 0。

定义 '&' 运算：对参与运算的两个数据的二进制位进行”与”运算。

运算规则：

$$\begin{aligned} 0 \& 0 &= 0 \\ 0 \& 1 &= 0 \\ 1 \& 0 &= 0 \\ 1 \& 1 &= 1 \end{aligned}$$

例如：3&5 即  $0000\ 0011 \& 0000\ 0101 = 0000\ 0001$ ，因此 3&5 的值为 1。

输入格式

第一行输入一个整数 $n(1 \leq n \leq 3000)$ ，代表序列中元素的个数。  
第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 2^{30} - 1$ ) - 序列 $a$ 。

输出格式

若经过至多  $n$  次操作后，能将序列中的所有数字改为 0，输出”Yes”；否则输出”No”。  
注意输出的是引号里面的内容，输出不包含引号。

样例

| standard input | standard output |
|----------------|-----------------|
| 3<br>1 2 5     | Yes             |

样例解释

本样例共进行三次操作：首先，令 $a_1 = a_1 \& a_2$ ,此时数组为{0, 2, 5};  
其次，令 $a_3 = a_1 \& a_3$ ,此时数组为{0, 2, 0};  
最后，令 $a_2 = a_1 \& a_2$ ,此时数组为{0, 0, 0}.  
因此可以将序列中的所有数字改为 0.

## Problem D. 低谷

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 seconds  
内存限制: 512 mebibytes

有一个长度为 $n$ 的数组，现在可以对其进行 $k$ 次操作，每次操作过程如下：

- 对于数组 $a$ ，选定一个下标 $i(1 \leq i \leq n)$ ，令 $a_i$ 的值减一。

现定义低谷：

- 对于数组 $a$ ，若对于下标 $i(2 \leq i \leq n-1)$ ，使得 $a_i < a_{i-1}$ 且 $a_i < a_{i+1}$ ，则称 $i$ 为数组 $a$ 的一个低谷。

lzt想知道 $k$ 次操作后，该数组的低谷最多有多少个。

### 输入格式

共两行。第一行包含两个整数 $n, k(1 \leq n \leq 3000, 1 \leq k \leq 10^9)$ ，分别表示数组长度 $n$ 和操作次数 $k$ 。

第二行包含 $n$ 个整数 $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^9)$ ，表示数组 $a$ 。

### 输出格式

输出一行一个整数 $ans$ ，表示数组的低谷最大个数。

### 样例

| standard input       | standard output |
|----------------------|-----------------|
| 7 3<br>2 2 3 4 6 2 3 | 3               |

### 样例解释

将数组中第二个数减1，第四个数减2，就能得到三个低谷。

## Problem E. 伤害最大化

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

现在王者荣耀·原神铁道这款游戏出了一个新英雄，有 4 个技能可以释放：

1. 对一名敌人造成  $d_i$  点伤害，其中每一个时刻  $i$  使用该技能造成的伤害点数都不相同；
2. 积累  $a$  点怒气值（不造成伤害）；
3. 对一名敌人造成之前积累过的所有怒气值之和的伤害(怒气值不清零)；
4. 接下来一个时刻所造成的伤害翻倍。即若在第  $i$  个时刻释放该技能，第  $i+1$  个时刻释放的技能会对一名敌人造成的伤害翻倍。

现在有  $n$  个时刻，每个时刻必须需要释放一个技能，并且技能的冷却时间为 2 个时刻，即：某一个技能若在第  $i$  个时刻释放，则无法在第  $i+1, i+2$  时刻释放。在初始时所有技能都未进入冷却状态。lzt想知道， $n$  个时刻后该英雄能对敌人造成的最大伤害是多少。

### 输入格式

共两行。第一行输入两个整数  $n, a (1 \leq n \leq 18, 1 \leq a \leq 10^5)$ ，含义同题面所示。

第二行输入  $n$  个整数  $d_1, d_2, \dots, d_n (1 \leq d_i \leq 10^5)$ ，表示若在第  $i$  个时刻释放技能 1，对敌人造成的伤害点数。

### 输出格式

共一行。输出一个整数  $x$ ，表示经过  $n$  个时刻后该英雄能对敌人造成的最大伤害。

### 样例

| standard input    | standard output |
|-------------------|-----------------|
| 5 9<br>1 10 7 3 8 | 38              |

### 样例解释

在这5个时刻中，这个英雄分别使用技能4,技能1,技能2,技能4,技能3，共造成38点伤害。

## Problem F. 猫猫虫困境 II

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

猫猫虫去年全部逃出第一个困境出口后，今年猫猫虫们又来到了第二个困境。猫猫虫们发现自己都在 $x$ 轴上，其中有 $k$ 个传送门也分布在 $x$ 轴上，且每个传送门都分别与左侧和右侧最近的一个传送门连接（如果存在），且每次需要一个单位的时间进行传送。困境出口只有1个，即横坐标 $g$ 处。猫猫虫、传送门和出口都在横坐标为 $[1, n]$ 之间。 $p$ 个猫猫虫散布在 $x$ 轴的不同位置，它们需要从困境出口逃离。每次向左移动一格或向右移动一格都需要一个单位的时间。并且，每次猫猫虫到达传送门时，可以选择无视传送门、向左传送至左侧最近的传送门、向右传送至右侧最近的传送门三种中的一种操作。lzt想知道每只猫猫虫到达困境出口，即到达横坐标为 $g$ 的点，需要的最少时间。

### 输入格式

第一行输入四个整数 $n, p, k, g$  ( $1 \leq g \leq n \leq 10^9, 1 \leq p, k \leq 10^5$ )，分别表示 $x$ 轴长度、猫猫虫数量、传送门数量、困境出口横坐标。

第二行输入 $p$ 个整数 $a_1, a_2, \dots, a_p$ ，其中第 $i$  ( $1 \leq i \leq p$ ) 个整数表示第 $i$ 个猫猫虫的横坐标 $a_i$  ( $1 \leq a_i \leq n, a_i$ 互不相同)。

第三行输入 $k$ 个整数 $b_1, b_2, \dots, b_k$ ，其中第 $i$  ( $1 \leq i \leq k$ ) 个整数表示第 $i$ 个传送门的横坐标 $b_i$  ( $1 \leq b_i \leq n, b_i$ 互不相同)。

### 输出格式

共一行。第一行输出 $p$ 个整数 $t_1, t_2, \dots, t_p$ ，其中第 $i$  ( $1 \leq i \leq p$ ) 个整数表示第 $i$ 个猫猫虫到达困境出口最少经过的时间单位。

### 样例

| standard input         | standard output |
|------------------------|-----------------|
| 10 2 2 7<br>2 8<br>4 7 | 3 1             |



## Problem G. 奥运会

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

奥运会，可以分为夏季奥运会和冬季奥运会，是全球范围内规模最大、最具影响力的体育盛会。夏季和冬季奥运会均为每4年举办一届。其中，最近的一场夏季奥运会为在2024年举办的巴黎奥运会，最近的一场冬季奥运会为在2022年举办的北京冬季奥运会。lzt想知道，在之后的时间里，若奥运会均正常举办，即夏季和冬季奥运会每四年正常举办一届，在 $L$ 年初到 $R$ 年末，共举办了几场奥运会？（夏季奥运会和冬季奥运会各单独算一场）

### 输入格式

输入共一行。第一行输入两个整数 $L, R$  ( $2025 \leq L \leq R \leq 9999$ )，含义同上述题面。

### 输出格式

输出共一行。第一行输出一个整数 $x$ ，表示在 $L$ 年初到 $R$ 年末举办的奥运会的场数。

### 样例

| standard input | standard output |
|----------------|-----------------|
| 2030 2050      | 11              |

### 样例解释

在2030年到2050年间，夏季奥运会在2032，2036，2040，2044，2048年举办，冬季奥运会在2030，2034，2038，2042，2046，2050年举办，共举办11场。

## Problem H. 文神的比赛

输入文件: `standard input`  
输出文件: `standard output`  
时间限制: 1 second  
内存限制: 512 mebibytes

文神主办了一场ICPC赛制的竞赛，共有  $n$  名选手参与，每个人通过的题目情况用一个字符串表示。字符串中的每个字符代表通过的题目编号，字符由大写字母A到Z构成，字符串的长度不固定且不超过26，每种字符在每个字符串中只出现一次。

每个字符串中的字符表示该选手通过的题目。例如，"ACD" 表示该选手通过了 A 题、C 题和 D 题。

选手的排名规则如下：

- 通过题目数较多的选手排名靠前。
- 如果通过的题目数相同，则选手的排名按罚时(保证不存在罚时相同的不同选手)决定，罚时小的排名靠前。

好奇的文神想知道，在只告诉你每个人通过题目的数量，不告诉你每个人罚时的情况下，最终排名会出现多少种情况。

文神太懒了不想自己写代码，请你帮助文神，输出可能的不同排名情况数。结果对  $10^9 + 7$  取模。

### 输入格式

第一行输入一个整数  $n(1 \leq n \leq 10^5)$ ，表示参赛者人数。

接下来 $n$ 行，每行包含一个字符串，表示每个人通过的题目编号，字符串的长度不超过26，字符串仅包含大写字母 'A' 到 'Z'，每种字符在每个字符串中最多只出现一次。

### 输出格式

输出一个整数，表示可能的不同排名情况数，结果对 $10^9 + 7$  取模。

### 样例

| standard input                   | standard output |
|----------------------------------|-----------------|
| 4<br>ABC<br>ABCD<br>ABCD<br>DBAE | 6               |

### 样例解释

由于1号选手通过了3题，而2、3、4号选手通过了4题，所以1号选手必然排第四名，而2、3、4号通过题数相同，所以排名由罚时决定，具体会出现六种情况：

选手2的罚时 > 选手3的罚时 > 选手4的罚时

选手2的罚时 > 选手4的罚时 > 选手3的罚时

选手3的罚时 > 选手2的罚时 > 选手4的罚时

选手3的罚时 > 选手4的罚时 > 选手2的罚时

选手4的罚时 > 选手3的罚时 > 选手2的罚时

选手4的罚时 > 选手2的罚时 > 选手3的罚时

因此共有 6 种不同的排名情况。

## Problem I. 智力博弈

输入文件: *standard input*  
 输出文件: *standard output*  
 时间限制: 1 second  
 内存限制: 512 mebibytes

cxh 和 lzt 是两个好朋友，他们喜欢在日常生活中进行各种智力博弈。某天，他们设计了一场新的游戏，并决定用一组特殊的数组来进行比拼。

给定  $n, k$ ，有一个长度为  $2n$  的数组  $v = \{1, 1, 2, 2, \dots, n, n\}$ ，即 1 到  $n$  中每个数恰好出现两次。现在可以将这个数组打乱，然后 cxh 和 lzt 进行博弈，cxh 先手，每次可以选择删除最左边或最右边的元素，A, B 轮流操作，轮流操作完之后算 1 轮结束。若  $k$  轮后剩余元素中存在一个数  $x$ ，使得  $x$  在剩余元素中仅出现一次，则 cxh 获胜，否则 lzt 获胜。lzt 想知道有多少种不同数组满足双方都采取最优策略下能使 lzt 获胜。由于答案可能很大，需要你将答案对  $10^9 + 7$  取模。

- 设两个数组  $v, w$ ，若  $v, w$  内的元素个数不相同，或者存在下标  $i$ ，使得  $v_i \neq w_i$ ，则可认为  $v, w$  为两个不同数组。
- 若  $k = 0$ ，则跳过操作过程，无需博弈。

### 输入格式

本题有多组测试数据。

第一行输入一个整数  $t (1 \leq t \leq 10^5)$ ，表示测试数据组数。

对于每组测试数据：

共一行。第一行包含两个整数  $n, k (1 \leq n \leq 10^6, 0 \leq k \leq n)$ ，含义如上述题面所示。

### 输出格式

对于每个测试用例，输出一行一个正整数  $ans$ ，表示能使得 lzt 获胜的数组个数对  $10^9 + 7$  取模之后的值。

### 样例

| standard input | standard output |
|----------------|-----------------|
| 3              | 4               |
| 2 1            | 6               |
| 2 0            | 432             |
| 4 3            |                 |

### 样例解释

在第一个测试用例中，能使 lzt 获胜的数组有  $\{1, 2, 2, 1\}, \{2, 2, 1, 1\}, \{2, 1, 1, 2\}, \{1, 1, 2, 2\}$ 。

Problem J. 星穹列车

输入文件: `standard input`  
输出文件: `standard output`  
时间限制: 1 second  
内存限制: 512 mebibytes

在二维平面上有一班星穹列车从 $(x_0, y_0)$ 出发，沿着方向向量 $\mathbf{v} = (1, 0)$ （即 $x$ 轴正方向）行驶，一直驶向银河彼岸（即可看作目的地无穷远）。其中，有 $k$ 个宇宙风暴，这些宇宙风暴在该平面上呈现为圆心坐标为 $(x_d, y_d)$ ，半径为 $r_d$ 的圆形。lzt想知道，镜流在列车行驶过程中，需要经过的宇宙风暴长度。

输入格式

第一行输入三个整数 $x_0, y_0, k$  ( $-10^9 \leq x_0, y_0 \leq 10^9, 1 \leq k \leq 10^5$ )，分别表示星穹列车的初始横坐标 $x_0$ ，纵坐标 $y_0$ ，以及危险区域个数 $k$ 。

后面 $k$ 行，第 $i$ 行输入三个整数 $x_{d_i}, y_{d_i}, r_{d_i}$  ( $-10^9 \leq x_{d_i}, y_{d_i} \leq 10^9, 1 \leq r_{d_i} \leq 10^9$ )，表示第 $i$ 个宇宙风暴的圆心横坐标 $x_{d_i}$ ，纵坐标 $y_{d_i}$ ，半径 $r_{d_i}$ 。

输出格式

共一行。第一行输出一个浮点数 $ans$ ，表示在列车行驶过程中，需要经过的宇宙风暴长度。

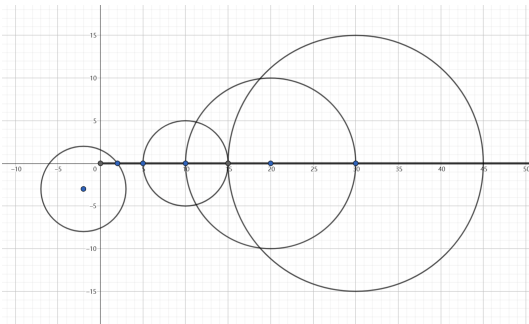
设你的答案为 $x$ ，标准答案为 $y$ ，只有当 $x, y$ 满足 $\frac{|x-y|}{\max(1,y)} \leq 10^{-5}$ 时，也就是说，只有当 $x$ 关于 $y$ 的绝对误差或相对误差小于等于 $10^{-5}$ 时，你的答案才会被视为正确。

样例

| standard input                                   | standard output                   |
|--|-----------------------------------|
| 0 0 4<br>10 0 5<br>20 0 10<br>30 0 15<br>-2 -3 5 | 42.000000000000000000000000000000 |

样例解释

样例如图所示，该路径需要经过42个单位长度的宇宙风暴(经过重叠的宇宙风暴区域时，路径不叠加计算)。



## Problem K. 冰红茶

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

给定 $n \times m$ 的网格 ( $n$ 行 $m$ 列), 再给定 $k$ 瓶冰红茶, 每瓶冰红茶有它自己的价格, 你需要将**所有**冰红茶放进这 $n \times m$ 的网格中, lzt会从每一行中都等概率随机拿取一瓶冰红茶 (若该行没有冰红茶, 则忽略该行), 并且支付这瓶冰红茶的钱, cxh作为老板, 想知道应该怎样放置可以使得lzt付的钱数期望最大(不要求每一行都必须有冰红茶, 同时一个方格最多只能放置一瓶冰红茶)

### 输入格式

共两行。

第一行输入三个整数 $n, m, k$  ( $1 \leq n, m \leq 1000, 1 \leq k \leq n \times m$ ), 含义如上述题面所示。

第二行输入 $k$ 个整数 $p_1, p_2, \dots, p_k$ , 其中第 $i$ 个整数表示第 $i$ 瓶冰红茶的价格 $p_i$  ( $1 \leq p_i \leq 10^9$ )。

### 输出格式

输出题意所求 $n$ 行 $m$ 列的网格 $v$ 。

具体地, 输出 $n$ 行, 其中第 $i$ 行输出 $m$ 个整数, 每个整数之间用空格隔开。若第 $i$ 行第 $j$ 列不放置冰红茶, 则 $v_{i,j} = 0$ ; 否则 $v_{i,j} = \text{index}$  ( $1 \leq \text{index} \leq k, \text{index}$ 为整数), 其中 $\text{index}$ 表示冰红茶对应的下标, 即: 第 $\text{index}$ 瓶冰红茶放置在第 $i$ 行第 $j$ 列的方格中。

若有多解, 输出任意一种即可。

### 样例

| standard input | standard output |
|----------------|-----------------|
| 2 3 3          | 1 2 0           |
| 1 4 6          | 3 0 0           |

### 样例解释

在该样例中, lzt付的钱期望是8.5. 可以证明这个期望是所有解决方案中最大的。

## Problem L. 数组交换

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

给定一个大小为 $n$  ( $n$ 为奇数)的数组 $a$ 。现在进行 $\frac{n-1}{2}$ 轮操作，每轮操作必须要：

1. 选定一个下标 $i$  ( $1 \leq i \leq n-1$ )，交换 $a_i$ 和 $a_{i+1}$ ；
2. 删除数组 $a$ 最前面的两个元素 $a_1$ 和 $a_2$ ，得到一个新的数组 $a' = \{a'_1, a'_2, \dots, a'_{n-2}\} = \{a_3, a_4, \dots, a_n\}$ ，以及新的数组元素大小 $n' = n - 2$ ；
3. 令 $a = a'$ ,  $n = n'$ ，更新数组 $a$ 和数组 $a$ 的大小 $n$ 。

完成之后，此时数组仅剩下一个元素。lzt需要采取一定策略，使得这个元素最大。求这个元素的值。

### 输入格式

第一行输入一个整数 $n$  ( $1 \leq n \leq 2 \times 10^5$ ,  $n$ 为奇数)，表示数组 $a$ 的元素个数。

第二行输入 $n$ 个整数 $a_i$  ( $1 \leq a_i \leq 10^9$ )，表示数组 $a$ 中的第 $i$ 个元素 $a_i$ 。

### 输出格式

共一行。第一行输出一个整数 $x$ ，表示所求元素的值。

### 样例

| standard input      | standard output |
|---------------------|-----------------|
| 7<br>10 6 2 1 5 3 4 | 5               |

### 样例解释

第一轮交换3和5，更新后数组为 $\{2, 1, 3, 5, 4\}$ ；

第二轮交换5和4，更新后数组为 $\{3, 4, 5\}$ ；

第三轮交换3和4，更新后数组为 $\{5\}$ 。

可以证明，最后剩下的元素的最大值就是5。

Problem M. 猜查看

输入文件: *standard input*  
输出文件: *standard output*  
时间限制: 1 second  
内存限制: 512 mebibytes

交互

一开始lzt会心中想一个正整数 $x_0(1 \leq x_0 \leq 10^9)$ ，然后你有三种操作可以进行：

- 1. " ? a"：询问lzt这个数能不能表示为正整数 $a(1 \leq a \leq 10^9)$ 与另一个非负整数的乘积。形式化地，记其此时心中所想的数为 $x$ ，询问是否存在 $d \in \mathbb{Z}(\mathbb{Z}$ 为全体整数的集合)，使得 $|ad| = x$ 。如果可以，lzt会输出 "Yes"；否则输出"No"。
- 2. "- b"：让lzt将心中所想的这个数减 $b(1 \leq b \leq 10^9)$ 。形式化地，记其此时心中所想的数为 $x$ ，进行这次操作之后令 $x := x - b$ 。
- 3. "! c"：输出一开始lzt心中所想的数 $x_0$ 。若输出的值 $c$ 与 $x_0$ 相等，则评测机将认为你的答案正确；否则会被视为 "Wrong Answer".

注意，进行操作3之后，你的程序应**立即终止**，否则评测机可能将给出除了"Accepted"之外的任意评测结果。

现在你能进行至多70次操作，问你能不能在70次操作内猜出lzt心中所想的数。

样例

| standard input | standard output |
|----------------|-----------------|
| ? 4            | No              |
| ? 5            |                 |
| - 5            | Yes             |
| ? 21           | Yes             |
| ? 22           |                 |
| ! 5            | Yes             |

样例解释

一开始lzt心中想的数是5。第一次你询问这个数能不能被4整除，lzt回答No，然后你再问这个数能不能被5整除，lzt回答Yes，然后你再把lzt心中所想的数减5，此时lzt心中所想的数是0.再问能不能被21,22整除，lzt回答Yes，于是你给出结果，猜测lzt原来心中所想的数是5.

注意，这个样例采取的策略可能并不明智，可能有很多其它答案，该样例只是作出一个示范。