

# RUC ICPC 2020 Individual Contest（硬核组）

RUC 2020校赛命题委员会

2020 年 12 月 19 日

## Contents

<b>A Ogi的ID</b>	<b>2</b>
<b>B 调分</b>	<b>3</b>
<b>C 合并数字</b>	<b>5</b>
<b>D 地毯</b>	<b>6</b>
<b>E 比赛</b>	<b>8</b>
<b>F 联机</b>	<b>10</b>
<b>G 逃离地牢</b>	<b>12</b>
<b>H Masou象棋</b>	<b>14</b>
<b>I 等式</b>	<b>17</b>
<b>J 斗地主</b>	<b>18</b>
<b>K 领口罩</b>	<b>20</b>
<b>L Sakuzyo的头套</b>	<b>22</b>

## A Ogi的ID

### A.1 Description

宇宙人民大学(**Renmin University of Cosmos, RUC**)的学生总是敢于探索新鲜事物。

而*Ogi*则喜欢尝试各种不同的游戏,但是给角色取名却总是困扰着他。为了便于取名,他希望自己的ID满足以下要求:

- ID以若干个O或o开头;
- 接着是若干个g;
- 然后是若干个i结尾.

现在给定一个字符串s,请你判断它是否能成为*Ogi*的ID,若能请输出YES,反之,请输出NO。

### A.2 Input

一行,一个字符串s( $|s| \leq 10^5$ ).

### A.3 Output

YES or NO.

### A.4 Sample Input1

0o0o0gggi

### A.5 Sample Output1

YES

### A.6 Sample Input2

gi

### A.7 Sample Output2

NO

## B 调分

### B.1 Description

*Ogi*助教生涯大危机!

噢,原来是昨天,算法课期中考试考完,同学们说他出的题太难了,什么线性复杂度最大团, $n$ 个球往盒子里来回取放,同学们全都没防出去,一题就把大家搞挂科了。

*Ogi*心想这可不是办法,得把分数调得好看一点,否则大家向老师投诉就没了。

*Ogi*的课有 $n$ 个学生( $n$ 为奇数),每个学生的分数为 $a_i$ ,每次可以对一个学生的分数做一点修改,将TA的分数+1或-1;*Ogi*希望经过若干次修改后,所有分数的中位数恰好为 $s$ ,同时,他希望修改次数尽可能小。

但是*Ogi*太摸了,于是他把任务交给了你,请你帮他解决这个问题。

### B.2 Input

本题有多组数据。

第一行一个数 $T$ ,表示接下来会有 $T$ 组数据,其中每组数据的格式如下:

第一行两个数 $n$ 和 $s$ ,表示学生的个数和目标的中位数的值;

第二行 $n$ 个数 $a_1 \cdots a_n$ ,表示每个同学的分数。

保证 $n$ 为奇数。

$$1 \leq \Sigma n \leq 10^6, 1 \leq s \leq 10^9$$

$$1 \leq a_i \leq 10^9$$

### B.3 Output

输出一个数表示最小的操作次数。

注意一个人的分数可以被加减无限次(但不能为负)。

### B.4 Sample Input

```
2
3 8
6 5 8
7 20
21 15 12 11 20 19 12
```

## B.5 Sample Output

2

6

## B.6 Sample Explanation

对于第一组数据，可以把6增加两次，分数变为8，5，8，中位数是8；

对于第二组数据，可以把19增加一次，15增加五次，数组变为21，20，12，11，20，20，12，排序后为11，12，12，20，20，20，21，中位数是20.

## C 合并数字

### C.1 Description

*Ogi*在实验室摸鱼的时候想到了一个合并数字的游戏，于是他邀请YYQ一起来玩。

游戏规则如下：初始共有 $n$ 个数字排成一个序列，YYQ先手，玩家每次可以将这个序列左边的 $m$ （ $m \geq 2$ 且 $m$ 小于等于当前序列长度）个数字合并，即删除这 $m$ 个数字，然后将这 $m$ 个数字的和加入到序列最左边，并且将这 $m$ 个数字的和加到自己的分数上，然后轮到另一位玩家，当序列长度为1时游戏结束。*Ogi*和YYQ的分数初始为0。

但是这时*Ogi*被抓去干活了，于是他让电脑接管。

*Ogi*想知道，最坏情况下，YYQ会领先他多少分？

### C.2 Input

输入数据的第一行为一个正整数 $n$ ，表示共有 $n$ （ $1 \leq n \leq 10^5$ ）个数字。第二行为 $n$ 个整数，每个数字 $c_i$ 满足 $|c_i| \leq 10000$ 。

### C.3 Output

一个整数，即最大落后分值。

### C.4 Sample Input

```
3
2 -1 2
```

### C.5 Sample Output

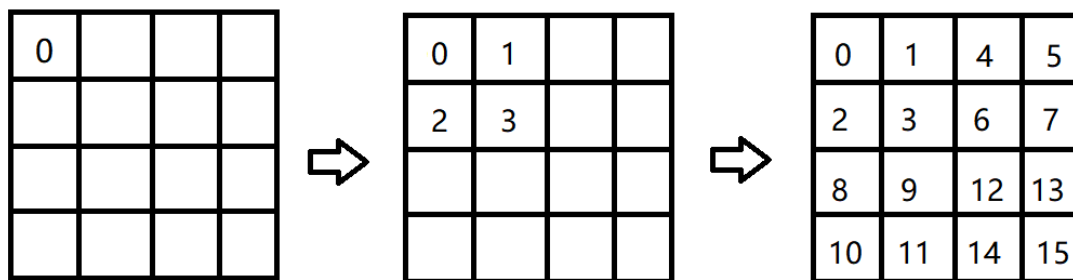
```
3
```

## D 地毯

### D.1 Description

*Ogi* 在宇宙杯自行车竞赛中拿到了一等奖！作为奖品，他拿到了一块面积足够覆盖一个星球的正方形地毯。现在他打算在上面写些数字：

假设这个地毯是 $2^n * 2^n$ 的，*Ogi* 从左上角开始，按一定的规律在每个 $1 * 1$ 的区域依次写下数字 $0..2^{2n} - 1$ 。规律如下：他每次先填满左上角 $2^k * 2^k$ 的区域，再接着填 $2^{k+1} * 2^{k+1}$ 的区域，他把 $2^{k+1} * 2^{k+1}$  分成四个长为 $2^k$ 的正方形区域，对于右上角的区域，把左上角区域的数字复制过来，再把每个数字加上 $2^{2k}$ ；对于左下角的区域，把左上角的复制过来再加上 $2^{2k} * 2$ ；右下角复制后每个加上 $2^{2k} * 3$ 。这样就把 $2^{k+1} * 2^{k+1}$ 的区域填满了。下面的图示可以帮你进一步理解整个过程。



作为解决 $NP = P?$  问题的第一步，*Ogi* 想知道地毯某些矩形区域上所有数字的异或和，但他懒得自己算，就请老工具人——你来帮帮忙了。

（异或：两个数 $a, b$ 的异或等于它们每一个二进制位依次求异或，规律遵循 $0 \wedge 0 = 0, 0 \wedge 1 = 1, 1 \wedge 0 = 1, 1 \wedge 1 = 0$ ）

### D.2 Input

第一行两个数 $n, q$ ，表示地毯的大小( $2^n * 2^n$ )和询问数；接下来 $q$ 行，每行4个数 $x_1, y_1, x_2, y_2$ ，表示询问左上角 $(x_1, y_1)$ 到右下角 $(x_2, y_2)$ 的矩形上数字的异或和。约定从上至下为 $x$ 轴，从左至右为 $y$ 轴。

$$n \leq 32, q \leq 2 * 10^5$$

$$0 \leq x_1 \leq x_2 \leq 2^{32} - 1, 0 \leq y_1 \leq y_2 \leq 2^{32} - 1$$

### D.3 Output

输出 $q$ 行，每行一个数字表示答案。(unsigned long long 可用%llu格式化输出。)

### D.4 Sample Input

```
2 3
0 0 1 1
1 1 2 3
3 3 3 3
```

### D.5 Sample Output

```
0
10
15
```

## E 比赛

### E.1 Description

为了减肥, *Ogi* 参加了宇宙杯自行车长途越野比赛 (**Cosmic Cup Bicycle Long-distance Cross-country Competition**), 这场比赛将持续很多宇宙天。为了保证足够的运动量, *Ogi* 事先准备了一堆糖果, 如果在早上吃了甜度为 $w$  的糖果, 就能在这一天以跃迁引擎骑行 $w$  km。

*Ogi* 想偷懒, 于是叫*Masou* 来帮忙。在每天早上, *Ogi*骑自行车, *Masou* 同时开太空车以 $K$  km/天的速度前进, 如果*Masou* 赶上了*Ogi*, 那么他就会在这一天蹭*Masou* 的车。注意, 两人一天中的移动速度是恒定的, 这意味着, 如果*Ogi* 的速度比*Masou* 快, 则在这一天必然不会蹭上车。

例子: 假设起点 $x = 0$ , 第1天, *Ogi*前进了 $5\text{km}(x = 5)$ , *Masou*前进了 $4\text{km}(x = 4)$ ; 第2天, 糖果甜度为1, *Ogi*预计前进 $1\text{km}$ , 而*Masou*前进了 $4\text{km}$ , 由于蹭车, 坐标为 $x=8$ ; 第3天, 糖果甜度为2, *Ogi*全天都蹭*Masou*的车, 坐标为 $x=12$ 。

现在比赛已经进行了 $n$ 天, *Ogi* 告诉了裁判员 (也就是你) 他 $n$ 天吃的糖果的甜度, 以及*Masou* 的太空车速度 $K$ , 想问你某一天停止移动时距离起点的总里程。另外, *Ogi* 可能记错了某些天吃的哪个糖果, 所以你可能需要多算一算。

特别地, 糖果的甜度 $w$  可能为负数, 这意味着*Ogi* 不知怎地往起点 (?) 骑行了 $-w$  km (仍然可以蹭车)。

### E.2 Input

第一行两个数 $n, q, K$ , 表示目前比赛天数和询问数以及汽车速度;

接下来一行 $n$ 个数 $w_i$ , 第 $i$ 个数表示第 $i$ 天吃的糖果甜度;

接下来 $q$ 行, 如果是询问, 则包含两个数 $a, b$ , 其中 $a = 0$ ,  $b$ 表示询问第 $b$ 天的总里程;

如果是修改, 则包含三个数 $a, b, c$ , 其中 $a = 1$ ,  $b, c$  表示第 $b$ 天的甜度改为 $c$ 。

$$n, q \leq 2 * 10^5$$

$$|w_i|, K \leq 10^9, K \geq 0$$

### E.3 Output

对于每个询问, 输出一行答案。注意当糖果甜度修改后, 之前的询问不需要重新计算。



#### **E.4 Sample Input**

```
4 3 4
5 1 2 6
0 4
1 2 6
0 4
```

#### **E.5 Sample Output**

```
18
19
```

## F 联机

### F.1 Description

上机课上，*Ogi*和*YYQ*经常一起联机打游戏。

机房里一共有 $n$ 台电脑，他们按 $1 \sim n$ 编号。由于机房太大了，学校使用了 $m$ 台路由器管理这些电脑，其中，第 $i$ 台路由管理 $l_i \sim r_i$ 号电脑，由于电脑和路由是随着机房的扩建和整修不断增加的，所以会出现一台电脑被多个路由管理的情况。

*Ogi*和*YYQ*需要在同一个局域网下（即使用由相同路由器管理的电脑）进行游戏，但是这使得他们能够选择的位置过于有限，所以他们想到利用一台电脑作为“跳板机”进行联机（即将一台电脑所属的局域网合并）。但是为了保证游戏体验，他们至多选择使用一台跳板机。

*Ogi*很好奇，他们有多少种不同的选位置方案呢（*Ogi*选择 $a$ 号电脑，*YYQ*选择 $b$ 号电脑与*Ogi*选择 $b$ 号电脑，*YYQ*选择 $a$ 号电脑是不同的方案。是否使用跳板机、使用哪台跳板机不影响方案数。）

### F.2 Input

第一行两个正整数 $n$ 和 $m$  ( $n \leq 10^9$ ,  $m \leq 10^6$ )。接下来 $m$ 行，每行两个整数 $l_i$ 和 $r_i$ 。

### F.3 Output

输出一个整数，表示不同的方案的数量。

### F.4 Sample Input

```
5 3
2 4
2 3
3 5
```

### F.5 Sample Output

```
12
```

### F.6 Sample Explanation

电脑1不在任何局域网里；

电脑2在局域网1、2里；

电脑3在局域网1、2、3里；

电脑4在局域网1、3里；

电脑5在局域网3里；

不使用任何跳板机，有(2,3)、(2,4)、(3,2)、(3,4)、(3,5)、(4,2)、(4,3)、(4,5)、(5,3)、(5,4)共10种方案；

以电脑3或4作为跳板机，还能多出(2,5)、(5,2)共2种方案；

总计12种方案。

## G 逃离地牢

### G.1 Description

YYQ被地牢之主（**dungeon master**）抓壮丁了...并被关在了地牢的最深处。

YYQ必须赶快逃离地牢，并且尽可能地保留力量。但是地牢危机四伏，一个区域要么有守卫看守，你需要损失一些力量将其击败，要么无法通行。

地牢可以被描述为一个 $n * m$ 的网格图，每一个格子中都有一个权值，保证权值为非负整数，即代表该区域的守卫值，若为0 则不能通行。

现在你们位于该网格图的左上角位置，YYQ的初始力量即为该格子权值，地牢的出口在网格图的右下角，现在你需要找到若干条路径，使得最终力量值最大，并且到达出口处。格子可以重复走，但是力量只会影响一次，我们认为只要走过的区域一样那么两种走法就是相同的。

力量计算公式：

$$x = \gcd(x, y)$$

$x$ 代表当前力量， $y$ 代表区域守卫值。

### G.2 Input

第一行输入一个正整数表示数据组数。

对于每一组数据，第一行两个正整数 $n, m$ 。

接下来 $n$ 行，每行 $m$ 个正整数描述迷宫。

### G.3 Output

对于每组询问输出一行两个答案，首先输出力量的最大值，其次输出可以得到该力量值的方案数，方案数很大，请将方案数对9819取模，若不存在任何到达出口的方案，YYQ只能接受命运，所以此时你只需输出两个0 即可。

### G.4 Sample input

```
1
3 3
4 4 4
4 0 2
4 2 4
```

## G.5 Sample output

2 13

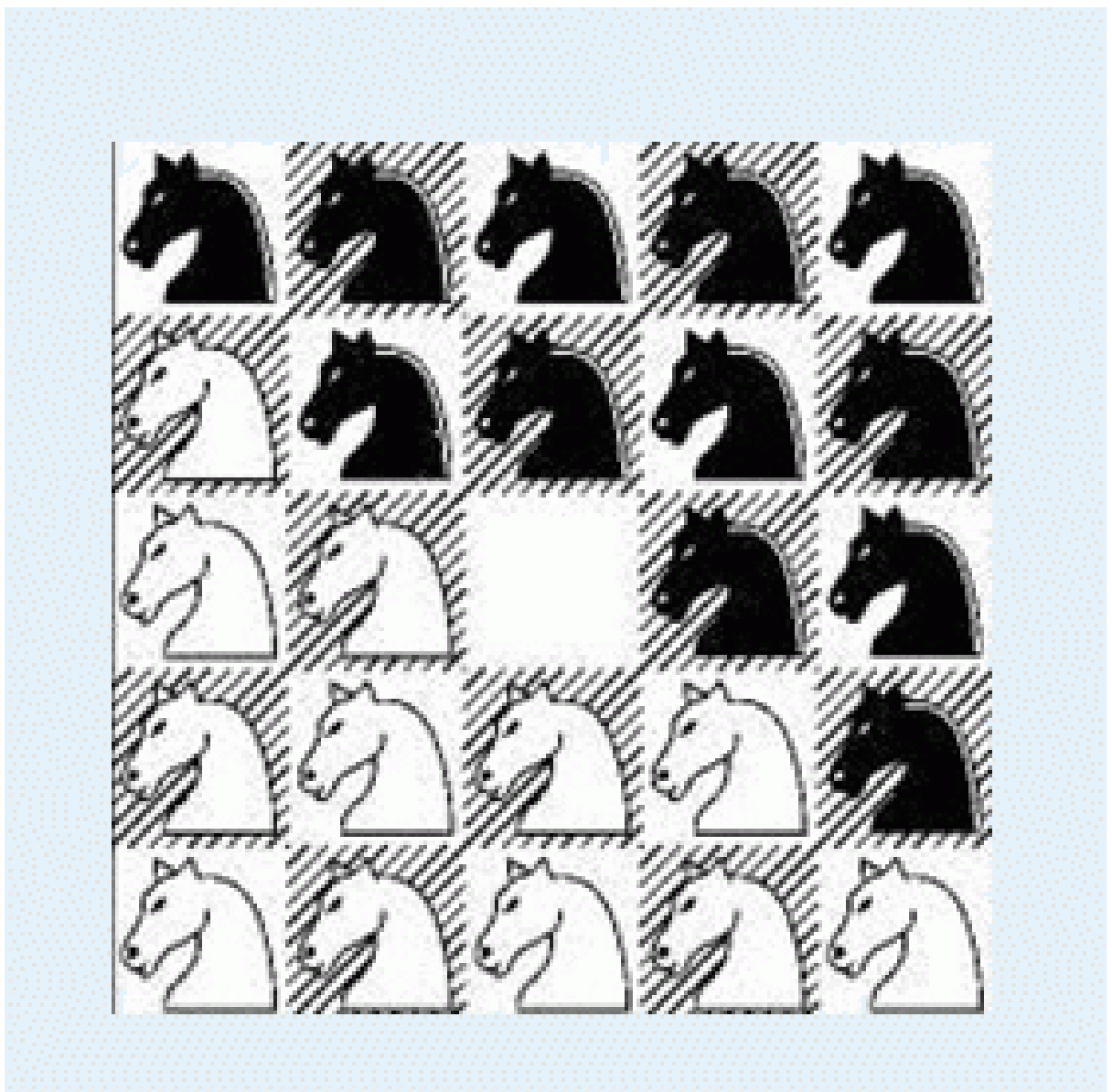
## H Masou象棋

### H.1 Description

今天是*Ogi*的生日，*YYQ*送他一副Masou象棋——一个 $5 \times 5$ 的类国际象棋棋盘、12个白色的骑士和12个黑色的骑士。

*Ogi*在研究一种新玩法：在棋盘上摆着12个白色骑士和12个黑色骑士，留下一个空位。任何时候一个骑士都能按照骑士的走法（它可以走到和它横坐标相差为1，纵坐标相差为2或者横坐标相差为2，纵坐标相差为1的格子）移动到空位上。

现在给你一个初始的棋盘，问怎样才能经过最少步数移动变成目标棋盘（下图），超过13步视为无解。



## H.2 Input

一个 $5 \times 5$ 的字符矩阵表示棋盘，其中\*表示空位，1表示黑棋，0表示白棋。

## H.3 Output

仅一行，一个整数，无解输出-1.

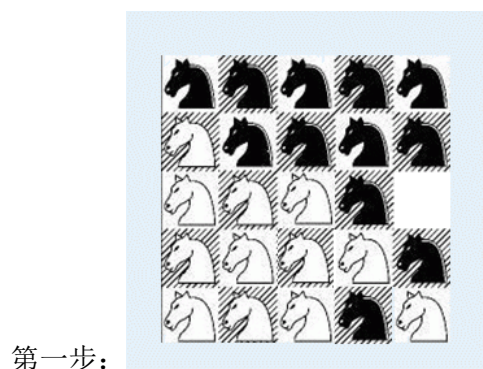
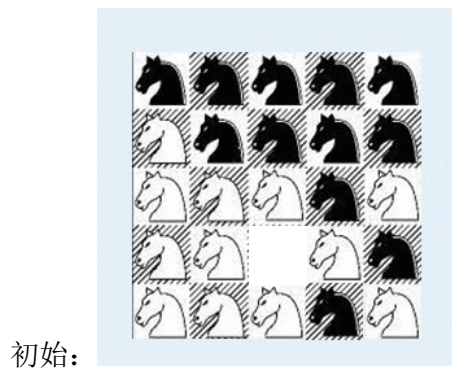
## H.4 Sample Input1

```
11111
01111
00010
00*01
00010
```

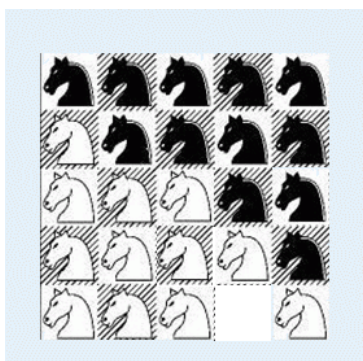
## H.5 Sample Output1

3

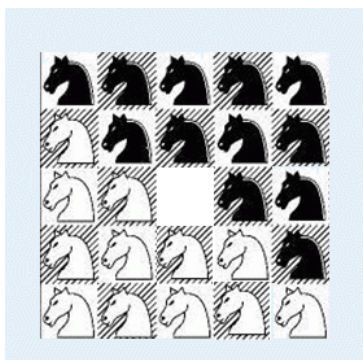
## H.6 Sample Explanation1



第二步:



第三步:



## H.7 Sample Input2

```
11111
01111
00*11
00001
00000
```

## H.8 Sample Output2

```
0
```

## H.9 Sample Explanation2

已经是目标棋盘，需0步移动。



## I 等式

### I.1 Description

*Masou*正在学习多位数的加法。今天老师布置的作业是：给出一个形如“ $a+b=c$ ”的等式，判断这个等式左右两边是否相等。不过*Masou*是个小机灵鬼，当别人还在列竖式做加法时，他已经想着怎么让问题变得更加有难度了。

具体地，*Masou*想的问题是这样的：现在对于每一个0-9的数字串 $s$ ，是否能够在其中插入一个加号和一个等号，使得插入后的串形如“ $a+b=c$ ”，其中 $a,b,c$ 都是无前导0的非空数字串，且数字 $a$ 和数字 $b$ 的和等于数字 $c$ 。

现在，*Masou*把某些等式的加号和等号去掉后的串给了你，希望你能还原原始的等式。注意，如果有多个不同的分法，输出任意一个均可。

### I.2 Input

输入包含多组数据，第一行一个数 $T$ 表示数据组数。接下来 $T$ 行每一行一个字符串 $s$ ，表示*Masou*给你的数字串。保证每一个输入均存在解。

$$T \leq 10, \sum |s| \leq 10^6$$

### I.3 Output

输出 $T$ 行，每行一个合法的等式，这个等式需满足你移除加号和等号后剩下的串和输入完全相同。如果有多个分法输出任意一个即可。

### I.4 Sample Input1

```
4
12345168
099
199100
123123123456456456579579579
```

### I.5 Sample Output1

```
123+45=168
0+9=9
1+99=100
123123123+456456456=579579579
```

## J 斗地主

### J.1 Description

斗地主（**fighting the landlord**）是一款中国民间传统扑克游戏，由三人进行游玩，其中一人扮演地主，另两人扮演农民。但我们今天并不探讨它的玩法，而是关注它的积分。

当三人持续游玩斗地主时，一种经典的积分方式为：地主底分6分，每位农民底分3分。每局游戏的倍数初始为1，游玩过程中，可能产生翻倍的操作。每次产生翻倍的操作，这一局的倍数乘以2。游戏结束后只有两种局面：地主胜利，两位农民失败；地主失败，两位农民胜利。胜利的一方中，每人获得底分 $\times$ 倍数的分数，失败的一方中，每人失去底分 $\times$ 倍数的分数。

一局比赛后，其中一名玩家得分的形式化表示为：

$$3 \cdot 2^l \cdot (-1)^{1-w} \cdot 2^m$$

其中 $l$ 代表是否为地主（若是地主 $l = 1$ ，否则 $l = 0$ ）， $w$ 代表是否胜利（若本局胜利 $w = 1$ ，否则 $w = 0$ ）， $m$ 代表本局游戏产生的翻倍操作的次数。注意， $l, w \in \{0, 1\}, m \in \mathbf{N}$

这一天，轩轩、凯凯和蓬蓬乘坐高铁从深圳返回北京，漫长的旅途中他们决定玩斗地主消磨时光。三人初始分均为0，蓬蓬负责记录三人的总分情况。若干局后（可认为经过了任意局，每局倍数任意值），蓬蓬记录三人的总分分别为 $a, b, c$ ，但他也不确定这是否为一个合法的分数。

蓬蓬想请聪明的你写一个程序帮他判断，一组总分是否是若干局斗地主游戏之后，可能产生的一组合法分数。

### J.2 Input

第一行一个正整数 $t(0 < t \leq 10^5)$ ，代表数据组数。

接下来 $t$ 行，每行一组以空格间隔的三个整数 $a, b, c(-10^5 \leq a, b, c \leq 10^5)$ ，代表一组需要你判断是否合法的总分。

注意各行数据之间没有联系，相互独立

### J.3 Output

输出 $t$ 行。

若是一组合法分数，输出YES；否则输出NO

### J.4 Sample Input

3 3 -5  
-57 42 15  
12 15 -27

### **J.5 Sample Output**

NO  
YES  
NO

## K 领口罩

### K.1 Description

为相应国家号召，积极防疫，蓬蓬所在的小区物业为住户分发口罩。在发放口罩前，蓬蓬观察到陆续前来的住户们会依次自动排成一排。

队列可以抽象为一条直线，这条直线上有编号 $1 \sim n$ 共 $n$ 个位置是可以站人的，每个位置能且仅能站一个人，且第 $i$ 个位置到第 $i + 1$ 个位置的距离均为1 ( $1 \leq i < n$ )。

初始队列中已经有 $k$ 个人，还有 $q$ 个人会依次前来（保证 $k + t \leq n$ ），这 $t$ 个人分为两种情况：

1. 坚定地走到队伍中空着的一个位置。
2. 犹豫不决，但是希望其所在的位置离已经在队伍中的人尽可能远。

现在蓬蓬想请你为犹豫不决的人指出满足他/她要求的位置。

### K.2 Input

第一行两个以空格分隔的正整数 $n, k$ 和 $q$  ( $0 < k, q, n \leq 10^5, k + q \leq n$ )，代表数据组数。

第二行 $k$ 个以空格分隔的整数，代表队列中初始的 $k$ 个人的位置编号。数据保证这些位置不重合。

接下来 $q$ 行，每行的第一个整数 $op$ 指示出两种类型的输入：

1. 若 $op = 1$ ，则该行还有一个与 $op$ 以空格相连的整数 $a$  ( $1 \leq a \leq n$ )，代表来了一个人，径直走到了位置 $a$ 。数据保证该位置此前无人。
2. 若 $op = 2$ ，则代表来了一个犹豫不决的人，需要你为他/她分配一个满足题意条件的位置（该位置此前需无人）。若有多个位置满足条件，则需要为他分配满足条件的位置中编号最小的那个。

注意你分配了一个位置给犹豫不决的人后，他/她会立刻站到队列中。

### K.3 Output

对每个 $op = 2$ 的输入，输出一个整数，代表你分配的位置。

### K.4 Sample Input

```
8 1 3
2
1 7
```

2

2

## K.5 Sample Output

4

1

## L Sakuzyo的头套

### L.1 Description

G2R2018 冠军曲目[*Destr0yer*]的BGA 中，主人公为头戴立方体的Sakuzyo。



但是有一天，蓬蓬产生了疑问，这个画面里的头套是一个标准的立方体吗？

于是蓬蓬请聪明的你帮他判断一下，他将给出的三维空间的八个点，询问它们是否分别是一个立方体的八个顶点。PS：在几何学中，立方体，是由6个正方形面组成的正多面体。（摘自Wikipedia）

### L.2 Input

第一行一个正整数 $t$  ( $0 < t \leq 100$ )，代表数据组数。

接下来 $8t$  行。其中第 $8i - 6 \sim 8i + 1$  行 ( $1 \leq i \leq t$ ) 为第 $i$  组数据的输入。

一组数据的8 行中，每行都为以空格间隔的3 个整数 $x, y, z$  ( $-10^5 \leq x, y, z \leq 10^5$ )，代表一个三维空间中的点的三个坐标。

保证一组数据的8 个点互相不重合。注意每组数据之间没有联系，相互独立。

### L.3 Output

输出 $t$  行。

若一组数据的8 个点，可以构成一个立方体，输出YES；否则输出NO

### L.4 Sample Input

0 0 0  
0 1 0  
1 0 0  
1 1 0  
0 0 2  
0 1 2  
1 0 2  
1 1 2  
0 0 0  
4 3 0  
1 7 0  
-3 4 0  
0 0 5  
4 3 5  
1 7 5  
-3 4 5

#### **L.5 Sample Output**

NO  
YES