

# RUC ICPC 2020 Individual Contest（标准组）

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 逃离地牢</b>	<b>8</b>
<b>F Masou象棋</b>	<b>10</b>
<b>G A ? B Problem</b>	<b>13</b>
<b>H 等式</b>	<b>15</b>
<b>I Naive数模转换器</b>	<b>16</b>
<b>J 斗地主</b>	<b>19</b>
<b>K Sakuzyo的头套</b>	<b>21</b>
<b>L 哆啦A梦的铜锣烧</b>	<b>23</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

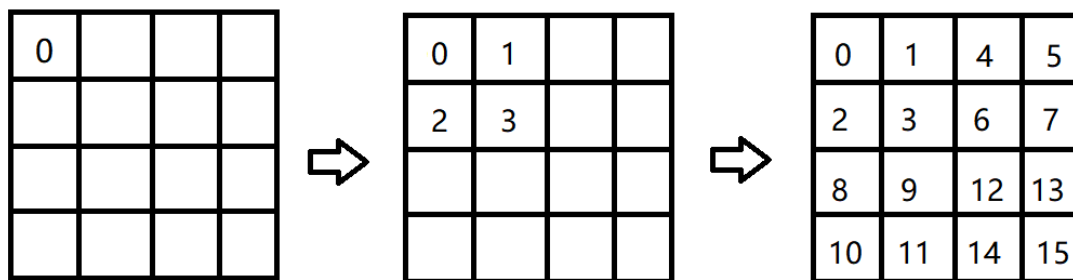
```
3
```

## 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 逃离地牢

### E.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。

### E.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。

### E.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。

### E.4 Sample input

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



## E.5 Sample output

2 13

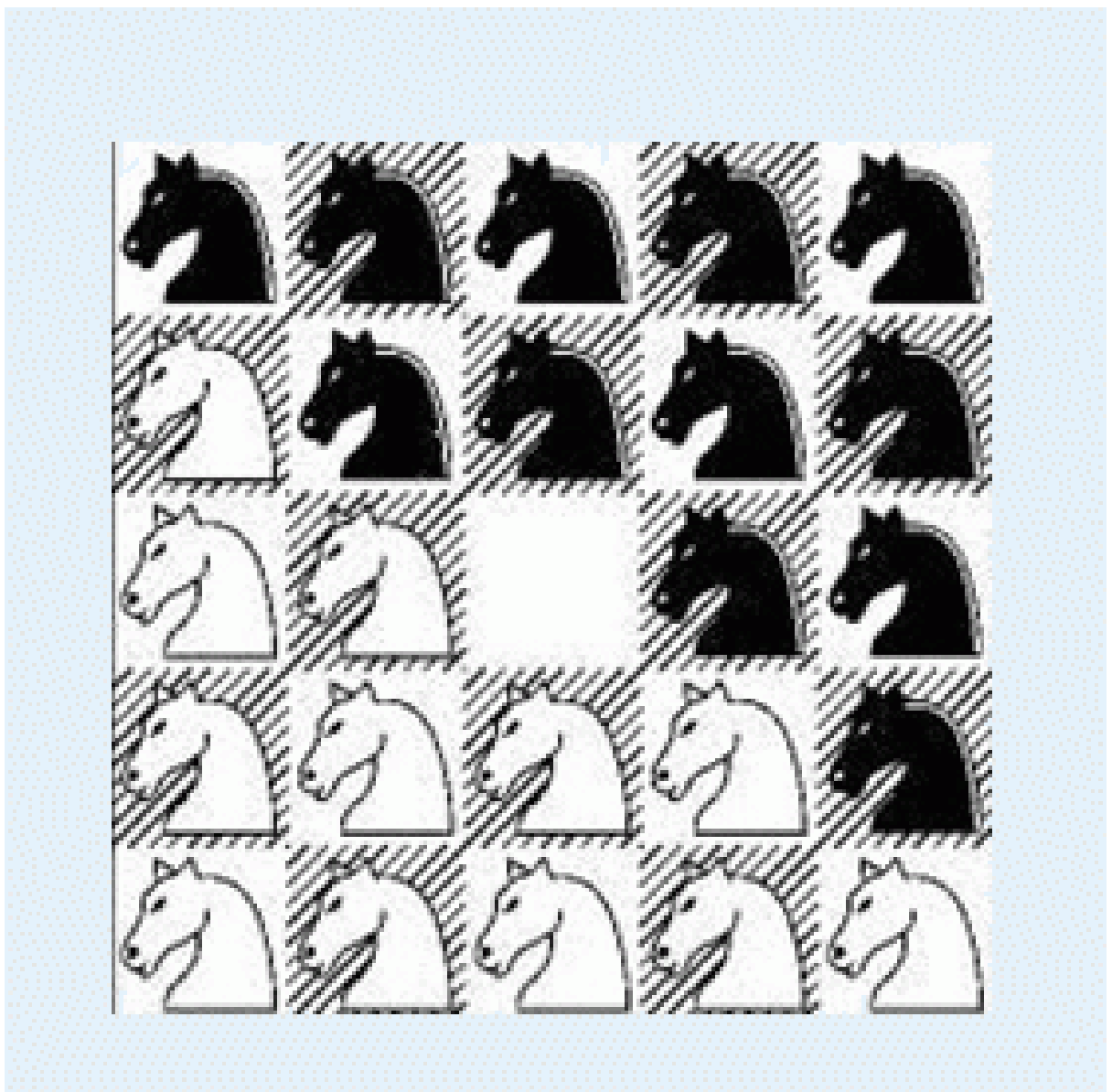
## F Masou象棋

### F.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。



## F.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。

## F.3 Output

仅一行，一个整数，无解输出-1.

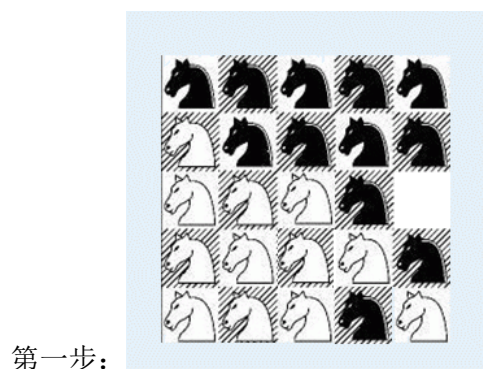
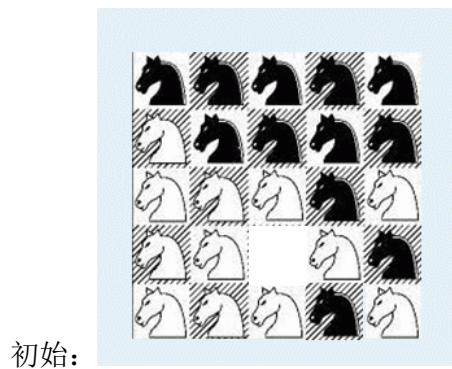
## F.4 Sample Input1

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

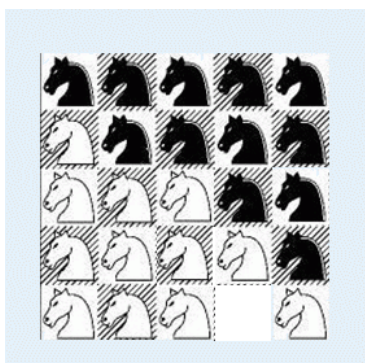
## F.5 Sample Output1

3

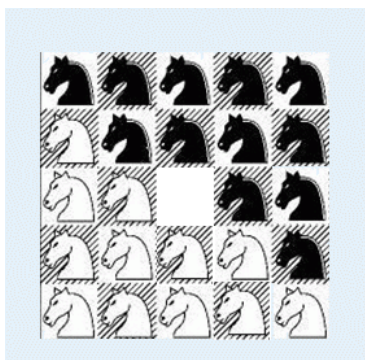
## F.6 Sample Explanation1



第二步:



第三步:



## F.7 Sample Input2

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

## F.8 Sample Output2

```
0
```

## F.9 Sample Explanation2

已经是目标棋盘，需0步移动。

## G A ? B Problem

### G.1 Description

本题是一道交互题

作为一名身经百战的算法竞赛选手，*Ogi*对于形如A ? B Problem的问题见得多了，例如：A + B Problem, A - B Problem 等等。

*Ogi*从自己的老电脑中翻出了一个未命名的程序，他只知道这个程序是A + B, A - B, A \* B, A  $\wedge$  B（异或）Problem中的一种，现在他想知道是哪一种Problem。

交互方式

最开始，*Ogi*会指定操作数A与B的值，你需要猜测这两个数字通过未命名程序计算后的结果。

随后，你可以输出“? a”向交互程序询问，计算结果是否为a( $|a| \leq 10^9$ )，若是，交互程序会输出“YES”，不然则输出“NO”。

当你确定答案后，输出“! a”向交互程序提交答案。请不要忘记每次输出后刷新缓存区。

注意：对于每个测试点，你最多进行10次询问。

各语言刷新缓存区方式：

- C/C++: `fflush(stdout)`
- Java: `System.out.flush()`
- Python: `stdout.flush()`
- Pascal: `flush(output)`

### G.2 Sample Input1

```
1 1
NO
NO
YES
```

### G.3 Sample Output1

```
? 114514
? 1919810
? 2
! 2
```

#### **G.4 Sample Input2**

0 0

#### **G.5 Sample Output2**

! 0

## H 等式

### H.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。

### H.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。

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

### H.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。

### H.4 Sample Input1

```
4
12345168
099
199100
123123123456456456579579579
```

### H.5 Sample Output1

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

## I Naive数模转换器

### I.1 Description

发烧友阿伟最近在学习音频编码知识：音频在计算机中以按照一定规则编码的数字流（二进制串）的形式储存，称为数字信号。数模转换器需要按照编码规则将数字流按从前往后的顺序，解码出一系列瞬时电平即模拟信号。

现实中的音频编码比较复杂，老师考虑到阿伟还是个初学者，只教了他两种简化后的编码：

**P编码：**对二进制串每16位做一次采样，记为 $x$ ，以补码形式储存的整数， $x$ 对应的瞬时电平 =  $x$ 的十进制值 / 32768，对从前往后每次采样依次输出对应的瞬时电平。为防止出现过载现象，所有采样的绝对值保证小于32768。

**D编码：**初始电平为0，对二进制每1位做一次采样，记为 $x$ ，若 $x$ 为1则瞬时电平+1，若 $x$ 为0则瞬时电平-1。因为模拟音频的一些规律，如果按P编码的方式解码D编码，一定会出现过载现象。

阿伟自己制作了一个输出数字流的编码器，现在需要你编写一个数模转换器验证阿伟的学习成果。

### I.2 Input

一行一个二进制串（长度不超过 $10^5$ ），可能是P编码或D编码。数据保证，只要正确解码，每个瞬时电平的绝对值小于32768。

### I.3 Output

一行一个整数，输出每次采样的瞬时电平。为了区分正负电平，我们规定：除了0，其他数值一定要加上正负号。

### I.4 Sample Input1

```
10011101010101110101010011110101
```

### I.5 Sample Output1

```
-25257
```

```
21749
```



## I.6 Sample Explanation1

这是P编码，采样两次：第一次1001110101010111，其原码是1110001010101001，十进制值为-25257；第二次0101010011110101，其原码是0101010011110101，十进制值为21749.

## I.7 Sample Input2

01001011111111111100000000000000

## I.8 Sample Output2

-1  
0  
-1  
-2  
-1  
-2  
-1  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0  
-1  
-2  
-3

-4  
-5  
-6

## **I.9 Sample Explanation2**

按照P编码解码会发生过载（1000000000000000十进制值为-32768），因此这是D编码。  
每1位采样一次，共32次，并输出瞬时电平。

## 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 Sakuzyo的头套

### K.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）

### K.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。

### K.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

### K.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

#### **K.5 Sample Output**

NO

YES

## L 哆啦A梦的铜锣烧

### L.1 Description

哆啦A梦和大熊正在金银岛旅行。他们发现了 $m$ 个美味诱人却大小各异的铜锣烧，并想把它们全部拿走。哆啦A梦的口袋里有 $n$  ( $n \leq m$ ) 个魔法背包，魔法背包的容量可以改变，但是哆啦A梦只能把所有背包的容量都换成 $C$ ；而且，背包的容量 $C$ 越大，他就会消耗越多的体力。

每个铜锣烧都有一个大小 $s_i$ ，意味着某个背包至少需要 $s_i$ 的剩余容量来携带第 $i$ 个铜锣烧 ( $1 \leq i \leq m$ )。哆啦A梦肯定不会放弃任何一个铜锣烧，并且出于强迫症，他想要在第 $i$ 个背包的最大铜锣烧尺寸不大于第 $i+1$ 个背包的最小铜锣烧尺寸 ( $1 \leq i < n$ )。当然，哆啦A梦不想花费太多的体力，因此他要求你帮助他确定最小容量 $C$ ，使他可以带走所有的铜锣烧。

### L.2 Input

第一行包含两个整数 $n$ 和 $m$  ( $n \leq m$ )，分别代表魔法背包的数量和铜锣烧的数量。

第二行包含 $m$ 个整数 $s_1, s_2, \dots, s_m$ ，第 $i$ 个数表示第 $i$ 个铜锣烧的尺寸大小。

$n, m \leq 2 * 10^5, s_i \leq 1000$

### L.3 Output

仅一个数字 $C$ ，表示背包的最小容量。

### L.4 Sample Input

```
4 6
8 3 3 5 2 7
```

### L.5 Sample Output

```
8
```