

## 2020 年牛客 NOIP 赛前集训营（第一场） 提高级

比赛地址：<https://ac.nowcoder.com/acm/contest/7605>

题目名称	牛牛的方程式	牛牛的猜球游戏	牛牛的凑数游戏	牛牛的 RPG 游戏
题目类型	传统型	传统型	传统型	传统型
每个测试点 时限	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒
内存限制	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB
子任务数目	10	10	10	10
测试点是否 等分	是	是	是	是

### 注意事项

- 所有参与 NOIP 赛前集训营的选手必须遵守约定的纪律：
    - 比赛账号不能外传。
    - 比赛中不能抄袭代码。
    - 比赛中不能恶意卡评测。
  - 报名支付账号即为比赛账号。
  - 一旦报名 NOIP 赛前集训营活动，不支持退费，请考虑清楚后报名。
  - 本活动解释权归牛客网所有，活动介绍未尽事宜以牛客网官方解释为准。
- 欢迎关注“比赛自动姬”公众号，关注更多比赛资讯~



## 牛牛的方程式

### 【题目描述】

牛牛最近对三元一次方程非常感兴趣。众所周知，三元一次方程至少需要三个方程组成一个方程组，才有可能得出一组解。

牛牛现在想要知道对于方程  $ax + by + cz = d$  中有没有至少存在一组  $\{x, y, z\}$  的解，且  $x, y, z$  都为整数，使得方程式成立。

### 【输入格式】

第一行输入一个正整数  $T$ ，表示测试点中测试样例的组数。

接下来  $T$  行，每行四个整数  $a, b, c, d$  表示方程  $ax + by + cz = d$  中的  $a, b, c, d$

### 【输出格式】

如果至少存在一组  $x, y, z$  能够满足方程式等式成立，且  $x, y, z$  均为整数，请输出 "YES"，否则请输出 "NO"。

### 【样例 1 输入】

2

3 1 2 0

2 8 8 3

### 【样例 1 输出】

YES

NO

### 【样例 1 解释】

$$1 \times 3 + (-1) \times 1 + (-1) \times 2 = 0$$

得到一组  $x, y, z$  的解为  $\{1, -1, -1\}$  为整数使得等式成立，所以输出 "YES"。

不存在  $x, y, z$  为整数使得方程  $2x+8y+8z=3$  成立，所以输出"NO"。

**【数据范围】**

对于10%的测试数据，保证  $T = 1, -10 \leq a, b, c \leq 10$

对于30%的测试数据，保证  $-100 \leq a, b, c \leq 100$

对于100%的测试数据，保证  $-10^{18} \leq a, b, c \leq 10^{18}, 1 \leq T \leq 100$

## 牛牛的猜球游戏

### 【题目描述】

牛牛和牛妹在玩猜球游戏,牛牛首先准备了 10 个小球, 小球的编号从 0~9。首先牛牛把这 10 个球按照从左到右编号为 0,1,2,3...9 的顺序摆在了桌子上, 接下来牛牛把这 10 个球用 10 个不透明的杯子倒扣住。

牛牛接下来会按照一定的操作顺序以极快的速度交换这些杯子。

换完以后他问牛妹你看清楚从左到右的杯子中小球的编号了么?

由于牛妹的动态视力不是很好, 所以她跑来向你求助。你在调查后发现牛牛置换杯子其实是有一定原则的。

具体来讲, 牛牛有一个长度大小为  $n$  的操作序列。

操作序列的每一行表示一次操作都有两个非负整数  $a,b$ , 表示本次操作将会交换从左往右数第  $a$  个杯子和从左往右数第  $b$  个杯子 ( $a$  和  $b$  均从 0 开始数)。请注意是换杯子, 而不是直接交换  $a$  号球和  $b$  号球。

牛牛和牛妹一共玩了  $m$  次猜球游戏, 在每一轮游戏开始时, 他都将杯子中的小球重置到从左往右依次为 0,1,2,3...9 的状态。

然后在第  $i$  轮游戏中牛牛会按照操作序列中的第  $l_i$  个操作开始做, 一直做到第  $r_i$  个操作结束 ( $l$  和  $r$  的编号从 1 开始计算)。

由于你提前搞到了牛牛的操作序列以及每一次游戏的  $l,r$ 。请你帮助牛妹回答出牛牛每一轮游戏结束时, 从左至右的杯子中小球的编号各是多少。

### 【输入格式】

首先输入一行两个正整数  $n,m$ , 表示操作序列的长度以及进行游戏的次数。

接下来  $n$  行每行两个非负整数  $a,b$ , 表示交换左数第  $a$  个杯子和左数第  $b$  个杯

子。(a,b 均从 0 开始数起)

接下来 m 行每行两个正整数 l,r 表示该轮游戏中牛牛从第 l 个操作开始做，一直做到第 r 个操作结束。(l 和 r 的编号从 1 开始计算)

### 【输出格式】

对于每一轮游戏，输出一行 10 个非负整数，表示从左至右每一个杯子中小球，输出的整数之间用空格隔开，行末不允许有多余空格。

### 【样例 1 输入】

5 3

0 1

1 2

2 3

0 1

9 0

1 5

5 5

3 5

### 【样例 1 输出】

9 1 3 0 4 5 6 7 8 2

9 1 2 3 4 5 6 7 8 0

9 0 3 2 4 5 6 7 8 1

### 【数据范围】

对于30%的测试数据，保证 $1 \leq n, m \leq 500$

对于60%的测试数据，保证 $1 \leq n, m \leq 4 \times 10^4$

对于60%以外另10%的数据，保证输入的 $a, b \in \{0, 1, 2\}$

对于100%的测试数据，保证 $1 \leq n, m \leq 10^5, 0 \leq a, b \leq 9, 1 \leq l \leq r \leq n$

## 牛牛的凑数游戏

### 【题目描述】

对于一个多重数集 $S$ ，对于一非负整数 $x$ ，若存在 $S \subseteq S'$  且 $S'$  中所有数字之和恰好等于 $x$ ，则说 $S$ 可以表示 $x$ 。

显然对于任意的多重数集都可以表示 0，因为空集是所有集合的子集。

牛牛定义集合 $S$ 的最小不能表示数为，一个最小的非负整数 $x$ ， $S$ 不能表示 $x$ 。

举个例子来说，例如 $S = \{1,2,3,8,9\}$ ，那么集合 $S$ 的最小不能表示数就为 7。

因为子集的和为 0，子集1的和为 1，子集2的和为 2，子集1,2的和为 3，子集1,3的和为 4，子集2,3的和为 5，子集1,2,3的和为 6。

但是无法找到子集权值和恰好为 7 的子集，所以 7 无法表示。

现在有一个长度大小为  $n$  的正整数数组，牛牛每次选择一个区间 $[l,r]$ ，他想要知道假定给出的多重数集为 $\{a_l, a_{l+1} \dots a_r\}$ 时，该集合的最小不能表示数是多少。

### 【输入格式】

第一行输入两个正整数  $n,m$ 。

接下来一行输入  $n$  个正整数 $a_i$ 表示输入的正整数数组。

接下来  $m$  行，每行输入两个正整数  $l,r$  表示查询的区间。

### 【输出格式】

对于每一个查询，输出最小不能表示数

### 【样例 1 输入】

8 6

1 2 3 4 5 17 1 99

1 5

2 5

1 6

1 7

1 8

3 8

### 【样例 1 输出】

16

1

16

34

34

2

### 【数据范围】

对于前10%的测试数据，保证 $1 \leq n, m \leq 10, 1 \leq a_i \leq 100$

对于前20%的测试数据，保证 $1 \leq n, m \leq 500$

对于另10%的测试数据，保证输入的 $a_i$ 单调非降

对于另10%的测试数据，保证输入的 $a_i$ 为 2 的非负整数幂。

对于100%的测试数据，保证 $1 \leq n, m \leq 10^5, 1 \leq a_i \leq 10^9, 1 \leq l \leq r \leq n$



## 牛牛的 RPG 游戏

### 【题目描述】

牛牛最近在玩一款叫做“地牢迷宫”的游戏，游戏中的每一层都可以看成是一个  $n \times m$  的二维棋盘，牛牛从左上角起始的  $(1,1)$  点移动到右下角的  $(n,m)$  点。

游戏中的每一个格子都会触发一些事件，这些事件将会影响玩家的得分。

具体来说，每到一个格子玩家触发事件时，首先会立即获得一个收益得分  $val(i,j)$ 。

注意这个得分不一定是正的，当它的值为负时将会扣除玩家一定的分数。

同时这个事件还会对玩家造成持续的影响，直到玩家下一次触发其他事件为止，

每走一步，玩家都会获得上一个事件触发点  $buff(i,j)$  的得分。

在游戏开始时牛牛身上还没有任何的  $buff$ ，所以在牛牛还未触发任何事件之前每走一步都不会产生任何影响。

牛牛使用“潜行者”这个职业，所以他路过地牢中的格子时，可以选择不触发这些事件。

同时牛牛是一个速通玩家，想要快速的到达终点，所以他每次只会选择往右走或者往下走。

牛牛想要知道，他玩游戏可以获得的最大得分是多少，你能告诉他么。

### 【输入格式】

第一行输入两个正整数  $n,m$

接下来输入  $n$  行，每行输入  $m$  个整数  $buff(i,j)$  表示该事件出发点被触发后，直到下一次触发事件，每移动一步改变的得分。

接下来输入  $n$  行，每行输入  $m$  个整数  $val(i,j)$  表示该事件出发点被触发后，分数的该变量。

输入保证, 对于起点和终点,  $val(1,1) = val(n,m) = buff(1,1) = buff(n,m) = 0$ 。

**【输出格式】**

输出仅一行一个非负整数, 表示牛牛从左上角走到右下角的最多得分。

**【样例 1 输入】**

```
3 3
0 1 -80
1 -1000 0
-100 0 0
0 -5 100
2 100 0
100 -1 0
```

**【样例 1 输出】**

```
20
```

**【样例 1 说明】**

一开始在(1,1)点, 得分 0, 身上没有事件影响 $buff$ 。

接下来移动到(1,2)点, 不触发事件, 得分 0, 身上没有事件影响 $buff$ 。

接下来移动到(1,3)点, 触发事件, 得分 100, 身上有 $buff$ 影响, 每走一步减少 80 点得分。

接下来移动到(2,3)点, 移动时扣除 80 得分, 身上还有 20 点得分, 然后触发事件, 得分+0, 同时  $buff$  被替换为每走一步+0。

接下来移动到(3,3)点, 结束游戏, 总得分 20。

**【数据范围】**

对于前10%的测试数据, 保证 $1 \leq n, m \leq 5$

对于前20%的测试数据, 保证 $1 \leq n, m \leq 30$

对于另20%的测试数据, 保证 $\min(n, m) = 1$

对于另20%的测试数据, 保证 $\min(n, m) = 2$

对于另10%的测试数据, 保证输入的 $buff(i, j) = 0$

对于100%的测试数据, 保证 $n \times m \leq 10^5, |val(i, j)| \leq 10^4$ ,

对于起点和终点,  $val(1, 1) = val(n, m) = buff(1, 1) = buff(n, m) = 0$