

Movie

题目描述

Ecrade_ 正在观看一个影片。

给定影片的时长，并假设该影片以恒定的 x 帧每秒的帧率播放，问该影片一共显示了多少帧？

输入格式

第一行一个整数 T ，表示测试数据组数。

对于每组测试数据，一行一个形如 AB:CD:EF 的字符串和一个整数 x ，表示该影片时长为 AB 时 CD 分 EF 秒，帧率为 x 帧每秒。

输出格式

对于每组测试数据，输出一行一个整数表示答案。

输入输出样例

【样例 1 输入】

```
2
00:00:01 1
06:06:06 120
```

【样例 1 输出】

```
1
2635920
```

【样例 1 解释】

- 对于第一组测试数据，可求得该影片共 1 秒，帧率为 1 帧每秒，故一共显示了 $1 \times 1 = 1$ 帧；
- 对于第二组测试数据，可求得该影片共 21966 秒，帧率为 120 帧每秒，故一共显示了 $21966 \times 120 = 2635920$ 帧。

数据规模与约定

本题采用捆绑测试。

- Subtask 1 (20 points) : $A = B = C = D = E = F = 0$ 。
- Subtask 2 (20 points) : $A = B = C = D = 0$ 。
- Subtask 3 (20 points) : $A = B = 0$ 。
- Subtask 4 (20 points) : $x = 1$ 。
- Subtask 5 (20 points) : 无特殊限制。

对于 100% 的数据, $1 \leq T \leq 100$, $0 \leq A, B, D, F \leq 9$, $0 \leq C, E \leq 5$, $1 \leq x \leq 240$ 。

Binary Matrix

题目描述

称一个矩阵为二进制矩阵，当且仅当该矩阵中所有元素为 0 或 1。

令 S 为所有同时满足如下两个条件的 $n \times m$ 的二进制矩阵 $B = (b_{i,j})$ 构成的集合：

- B 每行中所有数的异或和为 0，即 $\forall 1 \leq i \leq n, \bigoplus_{j=1}^m b_{i,j} = 0$;
- B 每列中所有数的异或和为 0，即 $\forall 1 \leq j \leq m, \bigoplus_{i=1}^n b_{i,j} = 0$ 。

Ecrade_ 有一个 $n \times m$ 的二进制矩阵 $A = (a_{i,j})$ 。他定义一个 $n \times m$ 的二进制矩阵 $B = (b_{i,j})$ 的权值为 $a_{i,j}$ 与 $b_{i,j}$ 中不同数的个数，即 $\sum_{i=1}^n \sum_{j=1}^m [a_{i,j} \neq b_{i,j}]$ 。

Ecrade_ 想知道， S 中矩阵权值的最小值为多少？

输入格式

第一行一个整数 T ，表示测试数据组数。

对于每组测试数据：

- 第一行两个整数 n, m 。
- 后 n 行每行一个长为 m 的数字串，其中第 $i + 1$ 行第 j 个字符表示 $a_{i,j}$ 。

输出格式

对于每组测试数据，输出一行一个整数 x ，表示 S 中矩阵权值的最小值。

输入输出样例

【样例 1 输入】

2
3 3
010
101
010
3 3
000
000
000

【样例 1 输出】

2
0

【样例 1 解释】

- 对于第一组测试数据， S 中权值最小的一个矩阵为 $\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ ；
- 对于第二组测试数据， S 中权值最小的一个矩阵为 $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ 。

数据规模与约定

本题采用捆绑测试。

- Subtask 1 (10 points) : $n = 1$ 。
- Subtask 2 (10 points) : $n, m \leq 2$ 。
- Subtask 3 (20 points) : $T \leq 10, n, m \leq 4$ 。
- Subtask 4 (20 points) : $n = 2$ 。
- Subtask 5 (40 points) : 无特殊限制。

对于 100% 的数据, $1 \leq T \leq 2 \times 10^5, 1 \leq n, m \leq 10^3, 1 \leq \sum n \times m \leq 10^6, a_{i,j} \in \{0, 1\}$ 。

Floor or Ceil

题目描述

Ecrade_ 有一个整数 x ，他会对其进行一些操作。

共有如下两种操作：

- 操作 1：令 $x \leftarrow \left\lfloor \frac{x}{2} \right\rfloor$ 。
- 操作 2：令 $x \leftarrow \left\lceil \frac{x}{2} \right\rceil$ 。

Ecrade_ 会以任意顺序进行恰好 n 次操作 1 和 m 次操作 2。他想知道在这 $n + m$ 次操作后， x 的值最小和最大分别是多少。

输入格式

第一行一个整数 T ，表示测试数据组数。

对于每组测试数据，一行三个整数 x, n, m 。

输出格式

对于每组测试数据，输出一行两个整数，分别表示在 $n + m$ 次操作后， x 的值最小和最大分别是多少。

输入输出样例

【样例 1 输入】

```
4
12 1 2
12 1 1
12 0 0
12 1000000000 1000000000
```

【样例 1 输出】

```
1 2
3 3
12 12
0 0
```

【样例 1 解释】

对于第一组测试数据：

- 依次进行操作 2,2,1, 则 x 会变为 $12 \rightarrow 6 \rightarrow 3 \rightarrow 1$, 可以证明这是 x 可变为的最小值;
- 依次进行操作 2,1,2, 则 x 会变为 $12 \rightarrow 6 \rightarrow 3 \rightarrow 2$, 可以证明这是 x 可变为的最大值。

数据规模与约定

本题采用捆绑测试。

- Subtask 1 (10 points) : $x = 0$ 。
- Subtask 2 (10 points) : $m = 0$ 。
- Subtask 3 (10 points) : $n = 0$ 。
- Subtask 4 (20 points) : $T, n, m \leq 10$ 。
- Subtask 5 (20 points) : $x \leq 10^5, n, m \leq 10$ 。
- Subtask 6 (30 points) : 无特殊限制。

对于 100% 的数据, $1 \leq T \leq 2 \times 10^5, 0 \leq x, n, m \leq 10^9$ 。

井字井字棋 (chess)

题目描述

井字井字棋的规则如下：

- 棋盘由 3×3 的大格组成，每个大格由 3×3 的小格组成，每个大格中的所有小格构成一个井字棋，而整个棋盘中的所有大格也构成一个井字棋。从整体来看棋盘如下：

- 对每个大格进行编号，第 i 行第 j 列的大格编号为 $3i + j - 3$ ；同理，对每个大格内的小格也可以如此编号。我们用 (i, j) 指代棋盘上编号为 i 的大格内编号为 j 的小格。
- 该游戏一定能决出胜负，且在胜负决出前双方轮流落子。
- 每个大格内的规则和井字棋一样，即若一方率先有三个子连成一线（横、竖、斜均可），则占领该大格。
- 整个棋盘的规则也和井字棋一样，即若一方率先有三个占领的大格连成一线（横、竖、斜均可），则获胜。
- 不能在有棋子的小格中落子；若一个大格被某一方占领了，则双方均不能再在该大格内的任意小格落子。
- 对于落子的限制：
 - 第一步不能下在 $(5, 5)$ ，即最中心的小格。

- 若该步不为第一步，设上一步对手下在了 (i, j) 。
 - 若编号为 j 的大格未被占领，则当前必须下在编号为 j 的大格里。
 - 若编号为 j 的大格被占领，则当前可下在棋盘上任意一个可以下的位置。
- 若直至无地方可落子还未决出胜负，则通过如下方法决出胜负：先比拼占领的大格数，多者获胜；若占领的大格数一样，则比拼未被占领的大格中占领的小格数，多者获胜。可以证明，按照此规则一定能决出胜负。

现在，给你一个棋谱，请你帮忙判断该棋谱是否合法、若合法是否决出了胜负、若决出了胜负是谁赢了。

输入格式

第一行一个整数 n ，表示棋谱长度。

接下来 n 行，第 i 行输入两个数 a_i, b_i ，表示第 i 回合落子的人下在了位置 (i, j) 。

输出格式

棋谱不合法，当且仅当存在 $i, 1 \leq i \leq n$ ，使得下第 i 步前每个人都下在能下的位置且没有决出胜负，而第 i 步下在了不能下的位置。此时输出两行，第一行为 `illegal`，第二行为 i 。

否则，棋谱合法。在此情况下，棋谱决出了胜负，当且仅当存在 $i, 1 \leq i \leq n$ ，使得下第 i 步前未决出胜负而下第 i 步后决出了胜负。此时输出两行，第一行为 `first` 或 `second`，分别表示是先手获胜还是后手获胜，第二行为 i 。

否则，棋谱没有决出胜负。此时输出一行 `to be continued`。

输入输出样例

【样例 1 输入】

```
2
5 5
5 4
```

【样例 1 输出】

illegal

1

【样例 2】

见附加文件中的 `chess2.in` 和 `chess2.out`。

【样例 3】

见附加文件中的 `chess3.in` 和 `chess3.out`。

【样例 4】

见附加文件中的 `chess4.in` 和 `chess4.out`。

数据规模与约定

- Subtask 1 (30 points)：棋谱不合法。
- Subtask 2 (30 points)：棋谱合法但未决出胜负。
- Subtask 3 (40 points)：无特殊限制。

对于 100% 的数据, $1 \leq n \leq 81$, $1 \leq a_i, b_i \leq 9$ 。