L. 大碗宽面 / Large

时间限制： 2.0 秒

空间限制： 512 MB

【题目描述】

Yazid 喜欢吃大碗宽面。现有 m 碗宽面，其中第 i 碗宽面 ($1 \leq i \leq m$) 共包含 n_i 根面条，它们的宽度分别为 $A_{i,1}, A_{i,2}, \dots, A_{i,n_i}$ 。

记 $f(u, v)$ 表示若混合第 u 碗宽面和第 v 碗宽面，将得到的超大碗宽面的第 $\lfloor \frac{n_u+n_v+1}{2} \rfloor$ 小的面条宽度 ($\lfloor x \rfloor$ 表示不超过 x 的最大整数)。

Yazid 想求出所有 $f(u, v)$ ，但为了节省你的输出时间，你只需要对所有 $1 \leq u \leq m$ 求出：

- $R(u) = \text{xor}_{v=1}^m (f(u, v) + u + v)$ (xor 指异或运算，在 C++ 语言中对应 `^` 运算符)。

【输入格式】

从标准输入读入数据。

第一行一个正整数 m ，表示宽面碗数。

接下来 m 行，每行若干个用单个空格隔开的整数描述一碗宽面：这部分的第 i 行的第一个正整数为 n_i ，表示第 i 碗宽面包含的面条数；接下来 n_i 个非负整数 $A_{i,1}, A_{i,2}, \dots, A_{i,n_i}$ 描述各面条的宽度。

保证 $m \leq 10000$ ， $n_i \leq 500$ ， $0 \leq A_{i,j} \leq 10^9$ 。

【输出格式】

输出到标准输出。

输出 n 行，每行一个整数，其中第 i 行的整数为 $R(i)$ 。

【样例 1 输入】

```
3
3 1 2 3
3 3 4 5
2 4 2
```

【样例 1 输出】

```
4
7
7
```

【样例 1 解释】

$$R(1) = (f(1, 1) + 2) + (f(1, 2) + 3) + (f(1, 3) + 4) = 4 \text{ xor } 6 \text{ xor } 6 = 4$$

$$R(2) = (f(2, 1) + 3) + (f(2, 2) + 4) + (f(2, 3) + 5) = 6 \text{ xor } 8 \text{ xor } 9 = 7$$

$$R(3) = (f(3, 1) + 4) + (f(3, 2) + 5) + (f(3, 3) + 6) = 6 \text{ xor } 9 \text{ xor } 8 = 7$$