Playground

Description

It has been more than half a year since Well graduated from UESTC; however, the wonderful memories of striving for ACM with his teammates never fade. He remembers that everyday when he finished his training in midnight, he would play games with WZP on the playground in front of Main Hall. One night, they played so crazily that they painted the playground with different colors. Before they left, they suddenly realized that if President Zhou saw the playground with lots of colored blocks, they would be punished. So they had to make the playground clean before the next day's sunrise.

Suppose that the playground is a square including $n * n$ small square blocks and some blocks have been painted by Well and WZP. Every time they would choose a row or a column of the playground and then begin to clean the painted blocks in the chosen row or column. One painted blocks would become clean after cleaning, however, if one painted block is cleaned twice, it will be become dirty again. Those blocks which are not painted before their cleaning starts will be always clean.

Well and WZP were very tired after a whole day's ACM training, so they wanted to finish this job as soon as possible. Given that choosing a row or a column and then do the cleaning would cost a minute, please help them find the minimum time they had to spend before the playground becomes all clean again.

Input

The input contains an integer T on the first line which indicates the number of test cases. In each test case, the first line is an integer n ($1 \le n \le 200$). Then n lines follow, each of which contains n integers separated by a blank. Each integer in these n lines, either 0 or 1, represents a small square block of the playground. Integer 0 means the block it represents is clean initially, while integer 1 means the block it represents is painted initially.

Output

For each test case, output a single line containing a single integer which represents the minimum minutes they had to spend to clean the playground.

Sample Input

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

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

Sample Output

```
0
1
2
```