Crazy Rows

(row.cpp/c)

You are given an **N** x **N** matrix with 0 and 1 values. You can swap any two *adjacent* rows of the matrix.

Your goal is to have all the 1 values in the matrix below or on the main diagonal. That is, for each X where 1 ≤ X ≤ N, there must be no 1 values in row X that are to the right of column X.

Return the minimum number of row swaps you need to achieve the goal.

Input **(row.in)**

The first line of input gives the number of cases, **T**. **T** test cases follow.
The first line of each test case has one integer, **N**. Each of the next **N** lines contains **N** characters. Each character is either 0 or 1.

Output **(row.out)**

For each test case, output

```
Case #X: K
```

where **X** is the test case number, starting from 1, and **K** is the minimum number of row swaps needed to have all the 1 values in the matrix below or on the main diagonal.

You are guaranteed that there is a solution for each test case.

Limits

1 ≤ **T** ≤ 60

1 ≤ **N** ≤ 8

Input	Output
3 2 10 11 3 001 100 010 4 1110 1100 1100 1000	Case #1: 0 Case #2: 2 Case #3: 4