Binary Matrix

You are given a matrix of size $\mathbf{r} \times \mathbf{c}$. Each of the elements can be either 0 or 1. In each operation you can flip any element of this matrix, i.e. convert 0 to 1 or convert 1 to 0. Your goal is to convert the matrix such that -

- 1. Each of the rows will have the same number of 1s and
- 2. Each of the columns will have the same number of 1s.

What is the minimum number of operations required to achieve this?

Input

Input starts with a positive integer T (~1000) which indicates the number of inputs.

Each case starts with two integers **m** and **n** ($1 \le r$, **c** ≤ 40), here **r** is the number of rows and **c** is the number of columns of the matrix. Each of the next **m** lines will have **n** integers each, either 0 or 1.

Output

For each test case, output "Case #: R" in a single line, where # will be replaced by case number and **R** will be replaced by the minimum number of steps required to achieve the target matrix. Replace **R** by -1 if it is not possible to reach target matrix.

Example

Sample Input:

Sample Output:

Case 1: 0 Case 2: 3 Case 3: 1