Kick Start 2021 - Round A

Checksum

Problem

Grace and Edsger are constructing a $\mathbf{N} \times \mathbf{N}$ boolean matrix $\mathbf{A}$. The element in i-th row and j-th column is represented by $\mathbf{A_{i,j}}$. They decide to note down the checksum (defined as bitwise XOR of given list of elements) along each row and column. Checksum of i-th row is represented as $\mathbf{R_i}$. Checksum of j-th column is represented as $\mathbf{C_j}$.

For example, if $\mathbf{N} = 2$, $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, then $\mathbf{R} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and $\mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}$.

Once they finished the matrix, Edsger stores the matrix in his computer. However, due to a virus, some of the elements in matrix $\mathbf{A}$ are replaced with -1 in Edsger's computer. Luckily, Edsger still remembers the checksum values. He would like to restore the matrix, and reaches out to Grace for help. After some investigation, it will take $\mathbf{B_{i,j}}$ hours for Grace to recover the original value of $\mathbf{A_{i,j}}$ from the disk. Given the final matrix $\mathbf{A}$, cost matrix $\mathbf{B}$, and checksums along each row ($\mathbf{R}$) and column ($\mathbf{C}$), can you help Grace decide on the minimum total number of hours needed in order to restore the original matrix $\mathbf{A}$?

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow.

The first line of each test case contains a single integer $\mathbf{N}$.

The next $\mathbf{N}$ lines each contain $\mathbf{N}$ integers representing the matrix $\mathbf{A}$. j-th element on the i-th line represents $\mathbf{A_{i,j}}$.

The next $\mathbf{N}$ lines each contain $\mathbf{N}$ integers representing the matrix $\mathbf{B}$. j-th element on the i-th line represents $\mathbf{B_{i,j}}$.

The next line contains $\mathbf{N}$ integers representing the checksum of the rows. i-th element represents $\mathbf{R_i}$.

The next line contains $\mathbf{N}$ integers representing the checksum of the columns. j-th element represents $\mathbf{C_j}$.

Output

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the minimum number of hours to restore matrix $\mathbf{A}$.

Limits

Memory limit: 1 GB.
$1 \le \mathbf{T} \le 100$.
$-1 \le \mathbf{A_{i,j}} \le 1$, for all i, j.
$1 \le \mathbf{B_{i,j}} \le 1000$, for i, j where $\mathbf{A_{i,j}} = -1$, otherwise $\mathbf{B_{i,j}} = 0$.
$0 \le \mathbf{R_i} \le 1$, for all i.

$0 \leq \mathbf{C_j} \leq 1$, for all j.
It is guaranteed that there exist at least one way to replace -1 in $\mathbf{A}$ with 0 or 1 such that $\mathbf{R}$ and $\mathbf{C}$ as satisfied.

Test Set 1

Time limit: 20 seconds.
$1 \leq \mathbf{N} \leq 4$.

Test Set 2

Time limit: 35 seconds.
$1 \leq \mathbf{N} \leq 40$.

Test Set 3

Time limit: 35 seconds.
$1 \leq \mathbf{N} \leq 500$.

Sample

Sample Input

```
3
3
1 -1 0
0 1 0
1 1 1
0 1 0
0 0 0
0 0 0
1 1 1
0 0 1
2
-1 -1
-1 -1
1 10
100 1000
1 0
0 1
3
-1 -1 -1
-1 -1 -1
0 0 0
1 1 3
5 1 4
0 0 0
0 0 0
0 0 0
```

Sample Output

```
Case #1: 0
Case #2: 1
Case #3: 2
```

In Sample Case #1, $\mathbf{A_{1,2}}$ can be restored using the checksum of either 1-st row or 2-nd column. Hence, Grace can restore the matrix without spending any time to recover the data.

In Sample Case #2, Grace spends one hour to recover $\mathbf{A_{1,1}}$. After that, she can use checksums of 1-st row and 1-st column to restore $\mathbf{A_{1,2}}$ and $\mathbf{A_{2,1}}$ respectively. And then she can use checksum of 2-nd row to restore $\mathbf{A_{2,2}}$. Hence, Grace can restore the matrix by spending one hour.

In Sample Case #3, Grace can spend one hour to recover $\mathbf{A_{1,1}}$ and another hour to recover $\mathbf{A_{2,2}}$. After that, she can use checksum to restore the rest of the matrix. Hence, Grace can restore the matrix by spending two hours in total.