Problem F. Logical Chain

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Every time you come across a problem you've never seen before, haven't you thought of something that is familiar to you? If so, you might think of something else, then more and more things will come to your mind. This is what is called a *logical chain*. Lu Xun's work also described this interesting phenomenon.

Assume there are n concepts, labeled by 1, 2, ..., n. Little Q's mind can be expressed by a $n \times n$ binary matrix g. If he can think of concept j when he comes across concept i, then $g_{i,j}$ is 1, otherwise it is 0. For two different concepts u and v, if u can lead to v directly or indirectly and v can also lead to u directly or indirectly, then the pair (u, v) is called a *looping pair*.

Little Q's mind changes all the time. On i-th day, a total of k_i positions (u, v) in matrix g are flipped (0s become 1s and 1s become 0s). Your task is to write a program to find the number of looping pairs each day after all changes on that day.

When counting, pairs (u, v) and (v, u) should be considered the same.

Input

The first line of the input contains two integers n and m, the number of concepts and the number of days ($1 \le n \le 250$, $1 \le m \le 25\,000$).

In the next n lines, line i contains n integers $g_{i,1}$, $g_{i,2}$, ..., $g_{i,n}$ ($0 \le g_{i,j} \le 1$, $g_{i,i} = 0$). Together, these lines define the matrix g.

Each of the next m parts describes a single day and starts with a line with a single integer k_i, the number of changes which happened on day i ($1 \le k_i \le 10$). Each of the next k_i lines contains two integers u and v denoting a changed position in g ($1 \le u, v \le n$, $u \ne v$).

It is guaranteed that each position will be changed at most once per day.

Output

For each day, print a single line containing a single integer: the number of looping pairs after all changes of that day.

Example

standard input	standard output
4 2 0010 1000 0000 0000 3 1 4 3 2 1 2 2 4 3 2 3	3 6