Lost in Dristor 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	256 megabytes

This is the legend of the old, long-lost package in Dristor 2.

A long time ago, AAP, MH, MFast and the other package hunters were tasked with finding a lost package in the Dristor 2 station. It is known that there are N package hunters and M secret locations in the Dristor 2 station. For each of the N package hunters, the secret locations he knows how to reach are known. Since none of them know the station well enough, they decided to split up to increase their chances of finding the package.

In one search, every package hunter must check at most one secret location he knows how to reach and no two hunters should check the same secret location. In each search, the number of package hunters that are not checking any of the secret locations is minimized (i.e. the number of secret locations that are checked is maximized). You should count the number of different searches the group can try in order to find the lost package. Since this number can be very large, you should print it modulo $10^9 + 9$. Two searches are considered different if at least one package hunter checks a different locations in the two searches.

Input

The first line of the input will contain N ($1 \leq N \leq 14$) and M ($1 \leq M \leq 100$). The next N lines of the input will each contain M binary values. If the jth value on the ith line is 1, then the ith package hunter knows how to reach the jth location. If the value is 0, then the ith package hunter does not know how to reach the jth location.

For tests worth 20 points, ($1 \leq N, M \leq 6$).

For tests worth 10 more points, each of the N package hunters knows where all the secret locations are.

Output

You should output the number of different searches the group can try (modulo $10^9 + 9$).

Example

standard input	standard output
4 4 0 1 1 0 1 1 0 1 1 0 0 0 0 1 1 1	3