Consider an n-by-n matrix A. We define $A^k = A * A * \ldots * A$ (k times). Here, $*$ denotes the usual matrix multiplication.

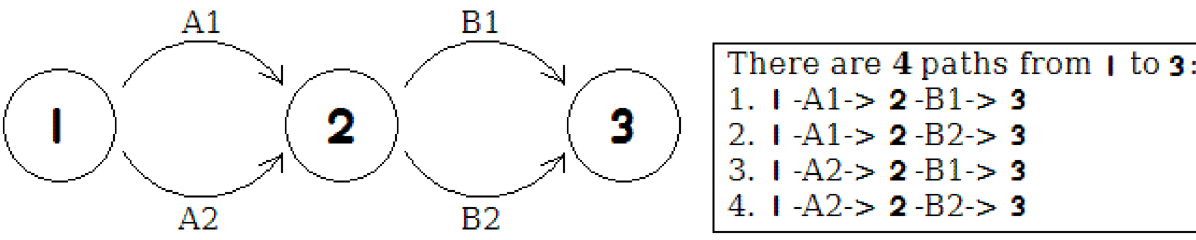
You are to write a program that computes the matrix $A + A^2 + A^3 + \ldots + A^k$.

Example

Suppose $A = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}$. Then $A^2 = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, thus:

$$A + A^2 = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 4 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 4 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

Such computation has various applications. For instance, the above example actually counts all the paths in the following graph:



Input

Input consists of no more than 20 test cases. The first line for each case contains two positive integers n (≤ 40) and k (≤ 1000000). This is followed by n lines, each containing n non-negative integers, giving the matrix A.

Input is terminated by a case where $n = 0$. This case need NOT be processed.

Output

For each case, your program should compute the matrix $A + A^2 + A^3 + \ldots + A^k$. Since the values may be very large, you only need to print their *last digit*. Print a blank line after each case.

Sample Input

```
3 2
0 2 0
0 0 2
0 0 0
0 0
```

Sample Output

```
0 2 4
0 0 2
0 0 0
```