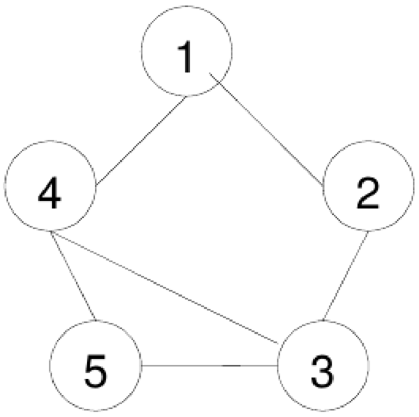
A computer network can be represented as a graph. Let G = (V, E) be an undirected graph, V = $(v_1, v_2, v_3, \ldots, v_m)$ represents all nodes, where m is the number of nodes, and E represents all edges. The first node is v_1 and the last node is v_m . The number of edges is k. Define the adjacency matrix $A = (a_{ij})_{m\times m}$ where

$$a_{ij} = 1 \text{ if } \{v_i, v_j\} \in \text{E, otherwise } a_{ij} = 0$$

An example of the adjacency matrix and its corresponding graph are as follows:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$



Calculate

$$A^n = \underbrace{A \cdot A \cdots A}_{n}$$

and use the Boolean operations where $0 + 0 = 0, 0 + 1 = 1 + 0 = 1, 1 + 1 = 1$, and $0 \bullet 0 = 0 \bullet 1 = 1 \bullet 0 = 0, 1 \bullet 1 = 1$. The entry in row i and column j of A^n is 1 if and only if at least there exists a walk of length n between the i-th and j-th nodes of V. In other words, the distinct walks of length n between the i-th and j-th nodes of V may be more than one. Note that the node in the paths can be repetitive.

The following example shows the walks of length 2.

$$A^2 = A \cdot A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Write programs to do above calculation and print out all distinct walks of length n. (In this problem we let the maximum walks of length n be 5 and the maximum number of nodes be 10.)

Input

The input file contains a number of test examples, the test examples are separated by '`-9999`'. Each test example consists of the number of nodes and the length of walks in the first row, and then the adjacency matrix.

Output

The output file must contain all distinct walks of the length n, and with all its nodes different, from the first node, listed in lexicographical order. In case there are not walks of length n, just print '`no walk of length` n'

Separate the output of the different cases by a blank line.

Sample Input

```
5 2
0 1 0 1 0
1 0 1 0 0
0 1 0 1 1
1 0 1 0 1
0 0 1 1 0
-9999
5 3
0 1 0 1 0
1 0 1 0 0
0 1 0 1 1
1 0 1 0 1
0 0 1 1 0
```

Sample Output

```
(1,2,3)
(1,4,3)
(1,4,5)

(1,2,3,4)
(1,2,3,5)
(1,4,3,2)
(1,4,3,5)
(1,4,5,3)
```