(Un)labeled graphs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

This is a run-twice problem.

Old man Pafnutiy wants to send a *labeled graph* G on n vertices to old man Infinitiy. Two labeled graphs on the set of vertices $[n] = \{1, 2, \ldots, n\}$ are different if and only if there are two distinct integers $i, j \in [n]$ such that vertices i and j are adjacent in one of the graphs but not in the other one.

Unfortunately, old man Pafnutiy's old computer is only capable of storing unlabeled graphs. Two unlabeled graphs on the set of vertices $[m] = \{1, 2, \ldots, m\}$ are different if and only if there is no permutation $\pi \in \mathbb{S}_m$ such that, if one renumbers the vertices of one of the graphs according to this permutation, then the two graphs become equal as labeled graphs. Therefore, if one tries to enter a labeled graph into old man Pafnutiy's old computer, the latter somehow shuffles all its vertices.

Old man Pafnutiy decided to send a bigger graph to old man Infinitiy: that is, if G has n vertices, old man Pafnutiy will send some graph H on $m = f(n) = n + \lceil \log_2 n \rceil + 3$ vertices. Then old man Infinitiy will only have to somehow decode the received graph H' back into G, given that H and H' are equal as unlabeled graphs. But how can they organize this transmission?

Input

The first line contains two integers v_1 and v_2: the number of vertices in the input and output graphs ($3 \le \min\{v_1, v_2\} \le 2024$). The next v_1 lines contain binary strings of length v_1: the adjacency matrix of the input graph.

The graph is undirected, so the adjacency matrix is symmetric. Additionally, it is guaranteed that there are no self-loops.

Output

Print v_2 binary strings of length v_2: the adjacency matrix of the output graph.

If $v_1 < v_2$, then $v_1 = n$, $v_2 = m = n + \lceil \log_2 n \rceil + 3$. In this case, you should act as old man Pafnutiy: you receive graph G and should output graph H.

Otherwise, $v_2 = n$, $v_1 = m = n + \lceil \log_2 n \rceil + 3$. In this case, you should act as old man Infinitiy: you receive graph H' (which is obtained from H after applying some permutation to its vertices) and should output the initial graph G.

The graph should be undirected, so the adjacency matrix should be symmetric. Additionally, there can be no self-loops in the graph.

Examples

standard input	*standard output*
5 11 01110 10110 11001 11001 00110	00010000000 00000000000 00000000000 10000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000
11 5 01000000000 10000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000	01110 10110 11001 11001 00110