Problem C. Choosing Best Friend

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

There are n people. Each two people among them are either friends or not. Friendship is bidirectional.

Each person wants to select best friends — some non-empty subset of their friends. The only condition they want to satisfy is that all sets of best friends among n people must be unique. Note that the "being the best friends" property may not be bidirectional (i.e., it is possible that X is best friend for Y, but Y is not the best friend for X).

You are given all friendships. Find any possible selection of sets of best friends such that the total size of sets of best friends is minimum possible. Or you should state that it is impossible to find such a selection.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases. Next lines contain descriptions of testcases.

The first line of each testcase contains a single integer n ($2 \le n \le 500$) — the number of people.

Then $n-1$ lines follow, the i-th line contains a string of length $n-i$ consisting of characters 0 and 1. For each j ($i < j \le n$), the $(j-i)$-th character of this string is 1 if i and j are friends, and 0 otherwise.

It is guaranteed that the sum of n^2 does not exceed $2.5 \cdot 10^5$.

Output

Print answers to testcases in the given order.

If it is impossible to select sets of best friends to satisfy the conditions, print a single integer -1.

Otherwise, print n lines. The i-th line should start with s_i ($s_i \ge 1$) — the number of selected best friends for the i-th person. Then s_i different integers $a_{i,1}, \ldots, a_{i,s_i}$ ($1 \le a_{i,j} \le n$, $a_{i,j} \ne i$) in the line should follow — best friends of the i-th person. For each j ($1 \le j \le s_i$) people i and $a_{i,j}$ should be friends.

All sets $\{a_{i,1}, \ldots, a_{i,s_i}\}$ should be different. The sum $\sum_{i=1}^{n} s_i$ should be minimum possible.

If there are multiple possible answers, you should print any.

Example

standard input	*standard output*
2 5 1000 011 10 1 3 11 0	1 2 1 1 1 4 1 3 2 2 4 -1