Problem J. Judge Error

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

In 2022, a terrible story happened at a local programming contest in Taiwan. A problem setter came up with the following problem:

> Given a number N, please construct a simple undirected graph with N vertices such that there is **exactly one perfect matching** in the graph, and under this condition, the number of edges is as large as possible. Two perfect matchings are distinct if the two sets of edges used in the matchings are different.

Back then, the problem setter D set the limit to $N \leq 500$ because another problem setter B tricked him into believing that there was an $O(N^3)$ algorithm that could be used to implement the checker. However, just before the problem submission deadline, B realized that his algorithm was fake and informed D that he might have to write an $O(N^4)$ algorithm instead. D followed through, but he chose not to reduce the limit on N, simply because the algorithm seemed to run fast enough.

As a result, during the official contest, one of the submissions caused the $O(N^4)$ checker to run for more than 10 seconds, causing an error in the judge system. What happened afterwards is another story.

Can you help implement a checker that always runs significantly faster? Oh, and to make things more interesting, let's solve the generalized version and increase the upper bound of N to 2000.

Input

The first line of input contains an integer N: the number of vertices ($2 \leq N \leq 2000$; N is even).

Then follow N lines. The i-th of these lines contains a binary string of length N, denoted by $g_{i,1}g_{i,2}\ldots g_{i,N}$ ($g_{i,j} \in \{\texttt{0}, \texttt{1}\}$). Together, these N lines define the graph: vertex i is connected to vertex j if and only if $g_{i,j}$ is `1`. The main diagonal consists of zeroes: $g_{i,i} = \texttt{0}$ for all i. The matrix is symmetric: $g_{i,j} = g_{j,i}$ for every pair (i, j).

Output

If the given undirected graph does not contain exactly one perfect matching (no need for the maximum number of edges), output "`No`" on a single line. Otherwise, output "`Yes`" on the first line, followed by $\frac{N}{2}$ lines: the i-th of these lines should describe the i-th matching pair by its endpoints, u_i and v_i ($1 \leq u_i < v_i \leq N$). Since there may be multiple possible orders for the matching, please output the matching such that $u_1 < u_2 < \ldots < u_{\frac{N}{2}}$.

Examples

standard input	*standard output*
4 0100 1010 0101 0010	Yes 1 2 3 4
4 0101 1010 0101 1010	No