Problem I. Imagined Holly

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Given a non-negative integer matrix A of size $n \times n$. For a tree with n vertices, numbered from 1 to n, we call it a *holly tree* if and only if:

- For any pair of vertices u and v $(1 \le u, v \le n)$, $A_{u,v}$ equals the bitwise XOR sum of the indices of the vertices on the simple path from u to v in the tree.

Your task is to construct a holly tree. It is guaranteed that such a holly tree always exists.

Input

The first line of the input contains an integer n $(2 \le n \le 2000)$, which is the size of matrix A.

The next n lines of the input describe the matrix A. The i-th line contains $n - i + 1$ integers $A_{i,i}, A_{i,i+1}, \ldots, A_{i,n}$ $(0 \le A_{i,j} < 2^{11})$. Note that $A_{i,j} = A_{j,i}$ holds for all $1 \le i, j \le n$.

It is guaranteed that such a holly tree always exists.

Output

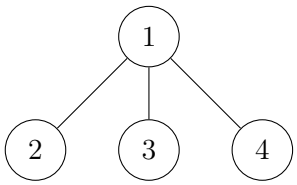
Output $n - 1$ lines, each containing two integers u and v $(1 \le u, v \le n)$, representing an edge of the holly tree.

Examples

standard input	standard output
2 1 3 2	1 2
4 1 3 2 5 2 0 7 3 6 4	1 2 1 3 1 4
6 1 7 4 5 2 3 2 1 6 7 0 3 5 4 3 4 3 2 5 5 6	4 1 2 3 6 4 5 2 4 2

Note

In the second example, the tree in the output is shown in the following figure:



This tree is a holly tree. For example, for the pair of vertices $(2, 4)$, the simple path from 1 to 4 includes vertices $2, 1$, and 4, with a bitwise XOR sum of $2 \oplus 1 \oplus 4 = 7$, which satisfies the constraint $A_{2,4} = 7$ given in the input.