D. Game with Points

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are playing the following game. There are n points on a plane. They are the vertices of a regular n-polygon. Points are labeled with integer numbers from 1 to n. Each pair of distinct points is connected by a diagonal, which is colored in one of 26 colors. Points are denoted by lowercase English letters. There are three stones positioned on three distinct vertices. All stones are the same. With one move you can move the stone to another free vertex along some diagonal. The color of this diagonal must be the same as the color of the diagonal, connecting another two stones.

Your goal is to move stones in such way that the only vertices occupied by stones are 1, 2 and 3. You must achieve such position using minimal number of moves. Write a program which plays this game in an optimal way.

Input

In the first line there is one integer n ($3 \le n \le 70$) — the number of points. In the second line there are three space-separated integer from 1 to n — numbers of vertices, where stones are initially located.

Each of the following n lines contains n symbols — the matrix denoting the colors of the diagonals. Colors are denoted by lowercase English letters. The symbol j of line i denotes the color of diagonal between points i and j. Matrix is symmetric, so j-th symbol of i-th line is equal to i-th symbol of j-th line. Main diagonal is filled with '*' symbols because there is no diagonal, connecting point to itself.

Output

If there is no way to put stones on vertices 1, 2 and 3, print -1 on a single line. Otherwise, on the first line print minimal required number of moves and in the next lines print the description of each move, one move per line. To describe a move print two integers. The point from which to remove the stone, and the point to which move the stone. If there are several optimal solutions, print any of them.

Examples

input
4 2 3 4 *aba a*ab ba*b abb*
output
1 4 1

input
4 2 3 4 *abc a*ab ba*b cbb*
output
-1

Note

In the first example we can move stone from point 4 to point 1 because this points are connected by the diagonal of color '$\texttt{a}$' and the diagonal connection point 2 and 3, where the other stones are located, are connected by the diagonal of the same color. After that stones will be on the points 1, 2 and 3.