Problem Statement

Given a set of N cities and distance between every pair of cities, find the shortest possible route that starts from a city and visits every city exactly once and returns to the starting city.
Write a program to find the shortest tour of N cities.

Input Specification

Your program must read from the standard input:
First line: either EUCLIDEAN or NON-EUCLIDEAN.
Second line: an integer N, number of cities.
Next N lines: each line contains 2D coordinates.
Next N lines: each line contains N distance values,

i.e., distance to kth city from each city.
The coordinates and distances are space separated list of floating-point numbers.
Use the coordinates for display.
Use the distance matrix for computing tour cost.

For example, a Euclidean TSP for N=5:
EUCLIDEAN
5
10.391379 8.405525
14.780237 7.036543
1.511838 6.366090
9.276912 5.418818
11.376465 4.216809
0.0 4.597411 9.110738 3.187861 4.302992
4.597411 0.0 13.285327 5.736168 4.420019
9.110738 13.285327 0.0 7.822640 10.096052
3.187861 5.736168 7.822640 0.0 2.419287
4.302992 4.420019 10.096052 2.419287 0.0

Output Specification

Your program must write the current best tour (in path representation) to standard output, one tour per line.

output line: PATH REPRESENTATION OF TOUR

The city indices in path-representation are zero based in-dices, it goes from 0 to N-1.

For example, for N=5:
1 2 4 0 3
0 1 3 4 2
4 3 2 0 1

Programming Language

Use Python 3.

Evaluation

Each program (process) will be allotted 300 seconds to complete its task and thereafter it will be terminated. The timeout period (300s) may be revised based on how the submissions perform in the trial run. The last valid tour in the standard output will be selected for evaluation. Your program must write at-least one valid tour to the standard output before termination, otherwise evaluation will fail and you will get zero points for that test case.