Problem D

Gold

Source file: gold.{c | cc | java | py2 | py3}

Author: Prof. Rodrigo Bossini Tavares Moreira

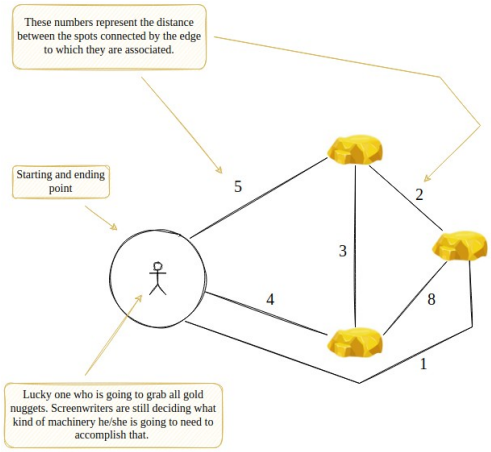
Have you seen a movie called “Gold” recently? No? Well, not that worth, although the main idea is not bad. Two guys find the biggest gold nugget in the world in the desert. Or something alike, we were not paying that much attention. One of them must stay with the nugget while the other leaves in order to get a bulldozer capable of extracting the entire nugget from the sand. No, we won’t tell you who dies at the end, please do not insist. There is a lot of death though. And the movie ends abruptly, you know? All of sudden. Not much explanations. It just ends. Ok, sorry for the spoiler. And for getting you to read stuff not really important to the problem. It is just that some movies really get you disappointed, you know? And sometimes you just need to talk. Well, to the problem. Someone said – not really, this part has just been made up - there will be a follow-up in which there will be many gold nuggets found in the wild. And the lucky ones who find them, at some point of the movie, will need to trace a map to respond the following question.

Given a starting point, what is the length of a smallest path through which one may go to collect all nuggets and return to the starting point?

It is important to know that the map includes the distance between each possible pair of spots.

Figure 1 shows an example.

Figure 1



It looks like in Figure 1 the length of a smallest path is 10. Can you find it? Not that difficult, huh? But make no mistake. This is a truly difficult problem. At least for instances with a significantly high number of spots.

Given a representation of a map, your task is to find the length of a smallest path as described. A map containing $3 \leq n \leq 10$ spots (including the starting/ending point) is represented as a square matrix **M** in which $1 \leq M_{ij} \leq 10$equals the distance between spots **i** and **j**. Map shown in Figure 1, for example, is represented as follows. First line of the matrix represents the distances between the starting/ending point and the other spots.

0 5 4 1
5 0 3 2
4 3 0 8
1 2 8 0

Input

First line of input contains **n**. **n** lines follow. Each of them contains the distances as described, which are all non-negative integers separated by a blank space. Remember the first one represents the starting/ending spot.

Output

Your program must produce a single integer which is the length of a smallest path as described. No break lines and/or blank spaces should be produced.

Example

Input

4
0 1 1 3
1 0 5 1
1 5 0 1
3 1 1 0

Output

4