# Problem B: Pinball machine

Advanced Algorithms for Programming Contests

## Restrictions

Time: 2 seconds
Memory: 512 MB

## Problem description

As you've become more and more obsessed with the pinball machine in your university's social room lately, you decided to finally crack its age-old high score today. For that however, you will need to earn the `blissful billion bonus`, by far the largest point bonus in this highly customized version of the game.

Consequently, it is also the hardest one to earn: There are $N$ objectives the player can complete to earn bonuses, each of which requires the machine's targets to have a certain (unique) activation pattern. To earn the `blissful billion bonus` one needs to complete all of them in the same round and – to make it even harder – all progress in doing so is lost once one completes an objective one had already completed in that same round. At least each objective's activation pattern requires the activation of a target that has to be inactive for all other objectives, so that no objective's activation pattern is a submask of another one's (otherwise one might sometimes have to make extra activation changes in order to avoid recreating the pattern of an already completed objective).

Thus, logically, the effort it takes to complete a desired objective when you just completed another one depends solely on the activation patterns of both those objectives. During the last couple of weeks you meticulously analyzed how much effort each of the transitions between pairs of activation patterns of different objectives takes. Now you want to use this data to find the most efficient way to earn the `blissful billion bonus`, i.e. the order in which you should complete the objectives so that the overall effort this takes you is minimal.

The game starts with all targets inactive. Since the effort it takes to get from there to the activation pattern of an objective is very similar for all of them, you simply don't take it into account. Instead, your calculation of the necessary effort begins at the exact point in time when the first objective is completed (although you may of course choose freely which one this should be).

# Input

The input consists of

- one line containing $N$ $(1 \leq N \leq 13)$ – the number of objectives
- $N$ lines containing $N$ numbers each, where the $i$-th number in row $j$ denotes the effort necessary to transfer between the activation patterns of objectives $i$ and $j$.

# Output

Output a single number – the minimal effort as specified above.

# Sample input and output

| **Input** | **Output** |
|---|---|
| 5<br>0 183 163 173 181<br>183 0 165 172 171<br>163 165 0 189 302<br>173 172 189 0 167<br>181 171 302 167 0 | 666 |