Journey in Matrix

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a square matrix M with side length n. $M_{i,j}$ is the element at row i and column j.

There are two journeys involved, one from $(1,1)$ to (n,n) and the second from (n,n) to $(1,1)$. In the forward journey, you are only allowed to go downwards and rightwards, and in the backward journey you are only allowed to upwards and leftwards. Moreover, in the first journey, you have to ensure that $i \geqslant j$ and in the second journey, $i \leqslant j$, if you are at a position (i,j).

You have to tell the maximum sum of the values you encounter on the two journeys.

Input

You are given an integer n ($1 \leqslant n \leqslant 15$). The next n lines contain n integers each, denoting values of the matrix (-100 $\leqslant M_{i,j} \leqslant$ 100).

Output

Output a single integer containing the maximum sum.

Example

`standard input`	`standard output`

`3`
`5 2 0`
`4 -3 -2`
`5 1 1`
	`22`

Note

In the above example, the optimal path for the forward journey is: $(1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3) = 5 + 4 + 5 + 1 + 1 = 16$. The optimal path for the backward journey is: $(3,3) \rightarrow (2,3) \rightarrow (1,3) \rightarrow (1,2) \rightarrow (1,1) = 1 +$ -2 $+ 0 + 2 + 5 = 6$.