JUMP THE BOARD!

An $n \times n$ game board is populated with integers, one nonnegative integer per square. The goal is to jump along any legitimate path from the upper left corner to the lower right corner of the board. The integer in any one square dictates how large a step away from that location must be. If the step size would advance travel off the game board, then a step in that particular direction is forbidden. All steps must be either to the right or toward the bottom. Note that a 0 is a dead end which prevents any further progress.

Consider the 4 × 4 board shown in Figure 1, where the solid circle identifies the start position and the dashed circle identifies the target. Figure 2 shows the three legitimate paths from the start to the target, with the irrelevant numbers in each removed.

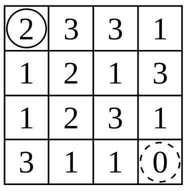

Figure 1

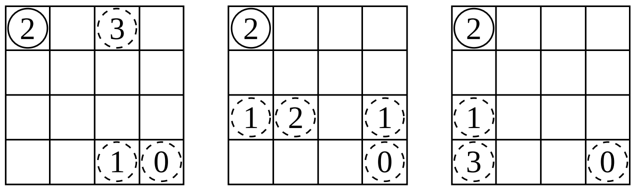

Figure 2

Your task is to write a program that determines the number of legitimate paths from the upper left corner to the lower right corner.

INPUT

The input ~~file **jump.in**~~ contains a first line with a single positive integer n, $4 \le n \le 100$, which is the number of rows in this board. This is followed by n rows of data. Each row contains n integers, each one from the range 0…9.

OUTPUT

The output ~~file **jump.out**~~ should consist of a single line containing a single integer, which is the number of legitimate paths from the upper left corner to the lower right corner.

EXAMPLE

standard input
~~**jump.in**~~

```
4
2 3 3 1
1 2 1 3
1 2 3 1
3 1 1 0
```

standard output
~~**jump.out**~~

```
3
```

GRADING

The number of legitimate paths can be quite big. Only 70% of the score can be achieved using a 64-bit integer variable (`long long int` in C, `Int64` in Pascal). It is guaranteed that all inputs will lead to a number of legitimate paths that can be written with no more than 100 digits.