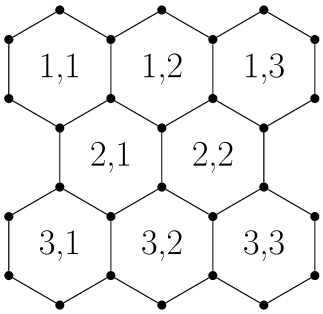
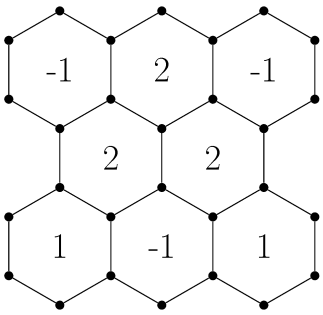
Problem H
Hexagon coloring

You are given a hexagonal grid with n rows, where n is an odd integer. The rows are numbered 1 to n from top to bottom. The odd-numbered rows have exactly n hexagons, and the even-numbered rows have exactly $n-1$ hexagons. Let's denote the j-th hexagon in the i-th row by (i, j).

For example, the below figure shows a hexagonal grid with $n = 3$.



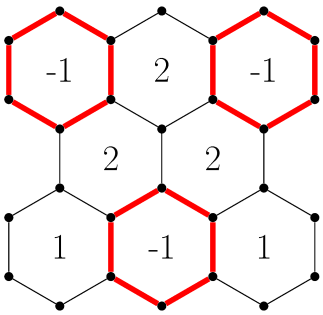
Let's assign an integer between -1 and 6 to each hexagon. Let's $a_{i,j}$ be the integer assigned to the hexagon (i, j). The following figure shows one example assignment:



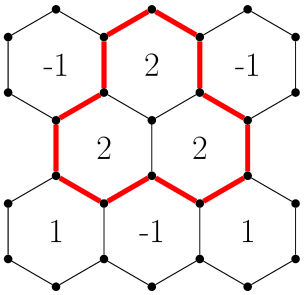
Let's color some edges of some hexagons. A coloring is **valid** iff it satisfies the following conditions:

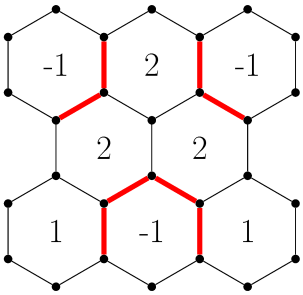
1. For every pair of valid indices i and j, either $a_{i,j} = -1$, or $a_{i,j}$ is equal to the number of colored edges of the hexagon (i, j).
2. The colored edges form one or more loops. Each loop must not self-intersect. Two different loops must not share any vertices or edges.

The following figure shows a valid coloring:



The following two figures show two **invalid** colorings. The one on the left does not satisfy the 1st condition, and the one on the right does not satisfy the 2nd condition.





How many valid colorings are there?

Input

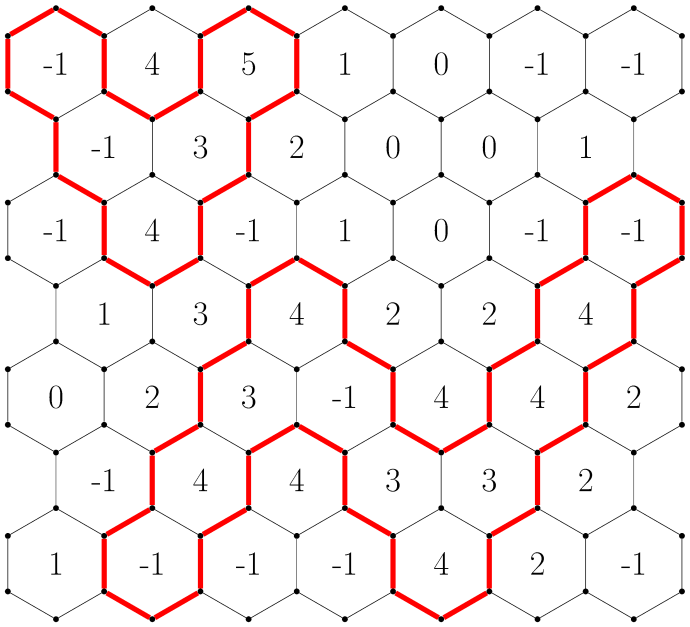
The first line of the input contains a single integer n (n is an odd integer between 1 and 7). The next n lines contain the numbers $a_{i,j}$ ($-1 \leq a_{i,j} \leq 6$). The i-th line contains exactly n integers if i is odd, and $n - 1$ integers otherwise.

Output

Print a single integer — the number of valid colorings.

Explanation of the sample input

The first sample was shown in the above figures.
The second example is shown below:



Sample Input 1

```
3
-1 2 -1
2 2
1 -1 1
```

Sample Output 1

```
1
```

Sample Input 2

```
7
-1 4 5 1 0 -1 -1
-1 3 2 0 0 1
-1 4 -1 1 0 -1 -1
1 3 4 2 2 4
0 2 3 -1 4 4 2
-1 4 4 3 3 2
1 -1 -1 -1 4 2 -1
```

Sample Output 2

```
1
```

Sample Input 3

```
3
-1 2 -1
2 2
-1 -1 -1
```

Sample Output 3

```
4
```