True Square in a Binary Matrix

Consider an $n \times n$ square matrix where each cell contains a binary integer (i.e., a 0 or 1). You can perform the following swap operation *at most* one time:

Choose two *rectangular* submatrices that do not intersect or overlap and *swap* them. Note that both submatrices must have the same exact dimensions and you *cannot* rotate or otherwise change their orientation.

Given an $n \times n$ binary matrix, perform *at most* one swap operation such that the largest $k \times k$ submatrix consisting only of 1's has a maximal value of k. Then print the value of this maximal k as your answer.

Input Format

The first line contains a single integer, n, denoting the length of the matrix's sides.
Each line i of the n subsequent lines contains n space-separated binary integers describing the respective values of each cell in row i of the matrix.

Constraints

- $1 \leq n \leq 300$

Output Format

Print the value of k for the maximal $k \times k$ submatrix consisting only of 1's.

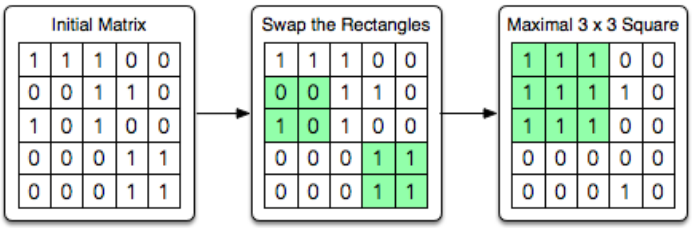
Sample Input

```
5
1 1 1 0 0
0 0 1 1 0
1 0 1 0 0
0 0 0 1 1
0 0 0 1 1
```

Sample Output

```
3
```

Explanation



In the initial matrix, the maximum k value is 2. If we use our swap operation to exchange the contents of two 2×2 *rectangles*, we can get a maximal k value of 3 (as demonstrated in the diagram above). Thus, we print 3 as our answer.