There are n zombies in Seattle, and Liv and Ravi are trying to track them down to find out who is creating new zombies — thus preventing an apocalypse. Other than the patient-zero zombies (who became so by mixing MaxRager and tainted Utopium), new people only become zombies after being scratched by an existing zombie; for this reason, zombiism is transitive. This means that if zombie 0 knows zombie 1 and zombie 1 knows zombie 2, then zombie 0 is connected to zombie 2. A zombie *cluster* is a group of zombies who are directly or indirectly linked through the other zombies they know (such as the one who scratched them or supplies them with brains).

Complete the *zombieCluster* function in your editor. It has 1 parameter: an array of binary strings (i.e., composed of 0s and 1s) named *zombies* that describes an $n \times n$ matrix of known connected zombies; if $zombies[i][j] = 0$, then the i^{th} and j^{th}zombies *do not* know one another (otherwise, the cell contains a 1 and they *do* know one another). Your function must return an integer denoting the number of zombie clusters Liv and Ravi have identified in Seattle.

Note: Method signatures may vary depending on the requirements of your chosen language.

Input Format

The locked stub code in your editor reads the following input from stdin and passes it to your function:
The first line contains an integer, n, describing the base size of your zombie association matrix. Each of the n subsequent lines contains a binary string of length n describing a row in the matrix.

Constraints

- $1 \leq n \leq 300$
- $0 \leq i < n$
- $|zombies| = n$
- Each $zombies[i]$ contains a binary string of n zeroes and ones.
- $zombies[i][i] = 1$, where $0 \leq i < n$.
- $zombies[i][j] = zombies[j][i]$, where $0 \leq i < j < n$.

Output Format

Your function must return a single integer denoting the number of different zombie clusters in Seattle. This is printed to stdout by the locked stub code in your editor.

Sample Input 0

```
4
1100
1110
0110
0001
```

Sample Output 0

```
2
```

Sample Input 1

```
5
10000
01000
00100
00010
00001
```

Sample Output 1

```
5
```

Explanation

In the diagrams below, the squares highlighting a known connection between two different zombies are highlighted in green. Because each zombie is already aware that they are personally a zombie, those are highlighted in grey.

Sample Case 0:

Sample Case 0

	Z_0	Z_1	Z_2	Z_3
Z_0	1	1	0	0
Z_1	1	1	1	0
Z_2	0	1	1	0
Z_3	0	0	0	1

We have $n = 4$ zombies numbered Z_0 through Z_3. There are 2 pairs of zombies who directly know each another: (Z_0, Z_1) and (Z_1, Z_2). Because of zombiism's transitive property, the set of zombies $\{Z_0, Z_1, Z_2\}$ is considered to be a single zombie cluster. The remaining zombie, Z_3, doesn't know any other zombies and is considered to be his own, separate zombie cluster ($\{Z_3\}$). This gives us a total of 2 zombie clusters, so we print 2 on a new line.

Sample Case 1:

Sample Case 1

	Z_0	Z_1	Z_2	Z_3	Z_4
Z_0	1	0	0	0	0
Z_1	0	1	0	0	0
Z_2	0	0	1	0	0
Z_3	0	0	0	1	0
Z_4	0	0	0	0	1

No zombie knows who any other zombie is, so they each form their own zombie clusters: $\{Z_0\}$, $\{Z_1\}$, $\{Z_2\}$, $\{Z_3\}$, *and* $\{Z_4\}$. This means we have 5 zombie clusters, so we print 5 on a new line.