fter IOI Ilya decided to make a business. He found a social network called "TheScorpyBook.com". It currently has **N** registered users. As in any social network two users can be friends. Ilya wants the world to be as connected as possible, so he wants to suggest friendship to some pairs of users. He will suggest user **u** to have a friendship with user **v** if they are not friends yet and there is a user **w** who is friends of both of them. Note that **u**, **v** and **w** are different users. Ilya is too busy with IPO these days, so he asks you to count how many friendship suggestions he has to send over his social network.

Input

The first line contains an integer number **N** — the number of users in the network. Next **N** lines contain **N** characters each denoting friendship relations. **j**th character if the **i**th lines equals one, if users **i** and **j** are friends and equals to zero otherwise. This relation is symmetric, i.e. if user **a** is friend of **b** then **b** is also a friend of **a**.

Output

Output a single integer — number of friendship suggestions Ilya has to send.

Constraints

- $1 \leq N \leq 2000$

Sample Input

4

0111

1000

1000

1000

Sample Output

6

```
import java.io.*;
import java.text.*;
import java.util.*;
import java.math.*;

class Set {
	static Scanner sc = new Scanner(System.in);

	static String i() {
		return sc.next();
	}

	public static void main(String[] args) throws IOException {

		int n = sc.nextInt();
		BitSet[] ar = new BitSet[n];
		for (int i = 0; i < n; i++) {
			String s = i();
			ar[i] = new BitSet(n);
			for (int j = 0; j < n; j++) {
				if (s.charAt(j) == '1')
					ar[i].set(j);
			}
		}

		long c = 0;
		for (int i = 0; i < n; i++) {
			for (int j = i + 1; j < n; j++) {
				if (!ar[i].get(j) && ar[i].intersects(ar[j])) {
					c += 2;
				}
			}
		}
		System.out.print(c);

	}
}
```