

Riding in cycles

There are N cities, numbered from 1 to N , in the country you are living. **Each pair** of the cities is connected with exactly one road. However, **each road is a one-way road**, so it is either possible to go directly from A to B or from B to A for each pair of cities (A, B) .

You are living in city #1 and you are practicing for upcoming cycling marathon, so you want to construct the following training plan:

First day you have to ride over 3 roads starting from and finishing in city #1.

Second day you have to ride over 4 roads in the same manner.

Third day you have to ride over 5 roads.

...

The last, $(N-2)$ -th, day you have to ride over N roads starting from and finishing in city #1.

You don't like to visit the same city more than once per day, so you have to find a training route for each day that passes through **each city at most once**. City #1 should appear only at the start and at the end of each route.

Write the program that, given the layout of the network, outputs training route for each day, or writes "impossible" if such training plan is not achievable.

Input

The first line of input contains the integer N ($3 \leq N \leq 1000$), number of cities.

Each of the next N lines contains exactly N characters that describes network layout. j -th character in i -th of these lines is '1' if it is possible to ride from city number i to city number j , or '0' otherwise.

Output

You should output training route for each day in a separate line. Training route consists of space separated integers - numbers of the cities in order they should be visited. Each training route starts and ends with 1.

If there is no achievable training plan, output word 'impossible' in a single line, instead.

Example

Input:

```
5
01000
00011
11001
10100
10010
```

Output:

1 2 5 1

1 2 4 3 1

1 2 4 3 5 1