

# Conflict

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            3 seconds  
Memory limit:         1024 megabytes

*In a desperate conflict, with a ruthless enemy...*

---

*This is an interactive problem.*

As an elite spy fighting against a great evil empire, you have been tasked with  $T$  separate reconnaissance missions. Each mission takes place in a different city of the empire, and your success is crucial to the resistance.

Every city is represented by  $N$  key buildings, numbered from 1 to  $N$ , connected by a network of roads. Each road connects two different buildings, and there may be multiple roads between the same pair of buildings. Your objective for each mission is to fully reconstruct the city's road network.

To achieve this, you are equipped with a special device capable of performing a **sequential shutdown** of the city's power grid. You can define a shutdown sequence  $p_1, p_2, \dots, p_N$  — a permutation of all  $N$  buildings. At the exact moment when the power to building  $p_i$  is cut, the device reports the number of roads that connect  $p_i$  to other buildings that **still have power**.

You are allowed to use this device at most  $N - 1$  times during each mission.

## Interaction Protocol

The first line of the input contains a single integer  $T$ , the number of test cases. Then,  $T$  test cases follow.

For each test case, the interaction begins with a single integer  $N$ , the number of buildings, given to your program via standard input.

Your program can then perform sequential shutdowns to explore the city's road network. To do this, you may issue queries of the form:

- ?  $p_1$   $p_2$   $\dots$   $p_N$  — Initiate a shutdown using your chosen **shutdown sequence**  $p_1, \dots, p_N$ .

In response to such a query, a single line containing  $N$  space-separated integers  $c_1, c_2, \dots, c_N$  will be given to your program via standard input. Here  $c_i$  is the measurement taken at the moment building  $p_i$ 's power is cut: the number of roads connecting it to other buildings that **still have power**.

You can perform at most  $N - 1$  sequential shutdowns per test case.

Once you have completed your mission and mapped out the entire road network, you must report your findings. First, print a line in the following format:

- !  $M$

where  $M$  is the total number of roads you have discovered.

Then, for the next  $M$  lines, print two space-separated integers  $u$  and  $v$ , representing a discovered road between buildings  $u$  and  $v$ . If multiple roads exist between the same pair, you must print that pair multiple times. The roads may be printed in any order; any order is acceptable.

The interactor is **non-adaptive**: the road network of each city is fixed in advance and does not depend on your queries.

## Scoring

- $1 \leq T \leq 1000$
- $2 \leq N \leq 1000$
- $0 \leq M \leq 10^4$
- The sum of  $N$  over all test cases does not exceed 2000.
- The sum of  $M$  over all test cases does not exceed  $10^4$ .

## Example

standard input	standard output
2	
3	
	? 1 2 3
2 1 0	
	? 3 2 1
2 1 0	
	! 3
	1 2
	2 3
	3 1
5	
	? 1 2 3 4 5
5 1 0 0 0	
	? 5 1 2 3 4
2 3 1 0 0	
	! 6
	1 2
	1 3
	2 3
	1 4
	1 5
	1 5

## Note

After printing each query, you must flush the output buffer to ensure the interactor receives your output. Failing to do so can result in an unexpected verdict. You can flush the output by using the following methods:

- In C++, call `fflush(stdout)` or `cout.flush()`.
- In Java, call `System.out.flush()`.
- In Python, call `sys.stdout.flush()`.
- In Kotlin, call `System.out.flush()`.

For other languages, you should refer to the official documentation for your language.