Is this a problem?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Of course, this is a problem.

This is an interactive problem. You are given an unknown undirected graph with n vertices and m edges, with no multiple edges or self-loops. Each time, you may ask the interaction library the following question:

- Given a vertex set $S \subseteq \{1, 2, \ldots, n\}$, ask whether this vertex set is an independent set.

Here, an independent set means that there is no edge between any two distinct vertices in the set S.

Your goal is to determine the entire edge set of the graph using no more than $M = n + m(2 + \lceil \log_2 n \rceil)$ queries.

Note that you do not know the value of m in advance, and the interaction library is **adaptive**, i.e. the graph is not fixed from the beginning, but will be adjusted dynamically according to your queries.

Input

The first line contains an integer T ($1 \leq T \leq 100$), denoting the number of test cases.

For each test case, one line contains an integer n ($1 \leq n \leq 500$), denoting the number of vertices of the unknown undirected graph.

It is guaranteed that the sum of n over all test cases does not exceed 500, $0 \leq m \leq 500$, and the sum of m does not exceed 500.

Interaction Protocol

When you need to ask a query, output one line: $?kv_1v_2...v_k$

where k denotes the size of the vertex set, and $v_1, v_2, \ldots, v_k$ are the indices of the vertices in the set you are querying, all pairwise distinct.

The interaction library will return an integer: `1` if the vertex set is an independent set; `0` otherwise.

When you have determined all edges of the graph, output one line: $!mu_1v_1u_2v_2...u_mv_m$

where m is the number of edges in the undirected graph, and (u_i, v_i) represents an edge in the graph. You may output the edges in any order.

After that, if there are no remaining test cases, your program must terminate immediately; otherwise, it should proceed to the next test case.

Example

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

Note

The sample demonstrates an interactive process of querying whether each edge exists individually.