Problem E. Eulerian?

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This problem is interactive.

We have hidden from you an undirected graph G on n vertices. It is guaranteed to be connected and to not contain multiple edges or self-loops.

You can ask **up to** 60 **queries** of the following form:

- Consider a subset S of all vertices of G. How many edges are there in the subgraph induced by S? In other words, how many edges in G have both their endpoints in S?

Your goal is to determine whether there exists an Eulerian cycle in this graph. An Eulerian cycle is a path in the graph that goes through every edge exactly once, and it starts and ends in the same vertex.

Note that **graph G is fixed before the start of interaction**. In other words, the interactor is not adaptive.

Input

The first line contains a single integer n ($3 \le n \le 10^4$), the number of vertices in G. It is guaranteed that G has no more than 10^5 edges, is connected, and does not contain multiple edges or self-loops.

Interaction Protocol

You start the interaction by reading a line with the integer n.

To find the number of edges in the subgraph of G on k vertices $x_1, x_2, \ldots, x_k$, print a line formatted as "`?` k x_1 x_2 ... x_k" ($0 \le k \le n$, $1 \le x_i \le n$, all x_i are distinct).

In response, the jury program will print a line with a single integer m: the number of such edges.

In case your query is invalid, or if you asked more than 60 queries, the jury program will print -1 and will finish interaction. You will receive "Wrong answer" outcome. Make sure to terminate your solution immediately to avoid getting other outcomes.

When you have determined whether the graph contains an Eulerian cycle, print a single line: "`! YES`" if such a cycle exists, and "`! NO`" if it doesn't.

After printing each line, do not forget to output the end-of-line and to flush the output. Otherwise, you will receive "Idleness limit exceeded" outcome.

Example

standard input	standard output
3 1 0	 ? 2 1 2 ? 2 1 3 ! NO

Note

The hidden graph in the example is the graph with 3 vertices and edges $(2, 1)$ and $(2, 3)$.