

## Problem M. Put Happiness in your Pocket

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

*This is an interactive problem.*

*Sea, you again. Wish you happiness next summer holiday!*

Long, long ago, Una and Kamome were already good friends.



Picture 1: Una and Kamome travelling in Qingdao

They liked playing guessing games. Last time, the game Una invented was quickly cracked by Kamome's excellent strategy. This time, Una brought a new game.

Una has a tree in mind with  $n$  nodes. Kamome can make no more than  $n$  queries:

- Kamome can tell Una a permutation  $p_1, p_2, \dots, p_n$  of  $0 \sim n - 1$ , Una will tell Kamome the length of the weighted diameter of the tree if node  $i$  is assigned a weight of  $2^{p_i}$ .

But sometimes, Kamome finds that Una is being a troublemaker. More specifically, sometimes no matter how many times she asks, she still can't determine the tree with certainty. In such cases, Kamome can report that there is no solution.

### Interaction Protocol

This is an interactive problem. Remember to flush the output buffer after every print. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java and Kotlin;
- `sys.stdout.flush()` in Python.

First, you should read an integer  $T$  ( $1 \leq T \leq 10^4$ ) indicating the number of test cases.

For each test cases, you should read an integer  $n$  ( $2 \leq n \leq 100$ ,  $\sum n^2 \leq 10^6$ ), indicating the size of the tree.

To make a query, you should output one line formatted as “?  $p_1 p_2 \dots p_n$ ” ( $0 \leq p_i \leq n - 1$ ,  $p_i \neq p_j$  for  $1 \leq i < j \leq n$ ).

Then, read a string that contains  $n$  characters indicating the weight of diameter of the tree in binary representation. If your query is invalid, or you make more than  $n$  queries in one test case, jury program will output  $-1$ . If your program reading  $-1$ , you should exit immediately to avoid undefined behavior.

If you determine that it is impossible to uniquely determine the tree, output “! NO”.

Otherwise, you should output one line formatted as “! YES  $u_1 v_1 u_2 v_2 \dots u_{n-1} v_{n-1}$ ” indicating this tree contains edges  $(u_i, v_i)$ . You can output them in any order.

Then, read one word OK or WA indicating whether your answer is correct. If your program reading WA, you should exit immediately to avoid undefined behavior.

If the tree is impossible to determine it and you output YES, even though you output the correct answer, your answer will also be considered **Wrong Answer**, because Una thinks you are cheating.

Note that the grader is **not adaptive**, that is the answer won't change after your queries.

## Examples

standard input	standard output
2	
4	? 0 1 2 3
1110	? 2 1 0 3
1110	! YES 1 4 2 4 3 4
OK	
3	? 0 1 2
111	! NO
OK	

## Note

A testing tool is provided to help contestants develop and test their solutions. You can download this tool from the attachments. Executing the tool with a “-h” option should describe how to use the tool. The testing tool will only implement some test scenarios and only some functionality of the real judge program.