Mouse

Maximum time of execution: 3 seconds/test.
Maximum available memory: 128 MB

Once upon a time, a special mouse tried to discover a secret permutation p[1] … p[N]. However, he wasn't able to without using a special device called the "permutation discoverer". Given some permutation q[1] … q[N], this device tells him the number of positions i for which p[i] = q[i]. He cannot use the device more than a certain number of times though.

More formally, there exists a secret permutation p[1] … p[N]. You can use an operation `query(q[1] … q[N])` that returns the number of positions i for which p[i] = q[i].

TASK

Given N, using a small enough number of queries, find p.

INTERACTION

This is an interactive problem. The contestant must implement a function `void solve(int N)` that eventually finds the hidden permutation p. To do this, include the header `grader.h`, and use the function `int query(vector<int> q)`. If given a permutation q, this function will implement the behavior described earlier. To answer, do a query with a q equal to what you think p is. If you are correct, the return value will, of course, be N. **You must terminate the solve function after receiving a result from query equal to N.**

CONSTRAINTS:

Subtask	Score	Restrictions
Subtask 1	13	N <= 7
Subtask 2	38	N <= 50
Subtask 3	49	N <= 256

SCORING:

Let Q be the number of queries used in a test. Then the scoring is as follows:

- **For Subtask 1:**

- **if Q <= 50, 13 points;**
- **if 50 < Q <= 200, 9 points;**
- **if 200 < Q <= 5040, 6 points;**
- **if Q > 5040, 0 points.**

- **For Subtask 2: let Q' = (floor(Q / 100) + 1) * 100**
  - **if Q' <= 400, then 38 points;**
  - **if 400 < Q' <= 700, then (38-29) * (700-Q') / (700-400) + 29 points;**
  - **if 700 < Q' <= 1300 then (29-21) * (1300-Q') / (1300-700) + 21 points;**
  - **if 1300 < Q' <= 10000 then (21-4) * (10000-Q') / (10000-1300) + 4 points;**
  - **If 10000 < Q', then 0 points.**
- **For Subtask 3: defin Q' as in subtask 2**
  - **if Q' <= 2400, then 49 points.**
  - **if 2400 < Q' <= 5000, then (49 - 29) * (5000 - Q') / (5000 - 2400) + 29**
  - **if 5000 < Q', then 0**

NOTE: If the correct permutation is found after too many queries, you will get the verdict "Correct", with detail "Too many queries", and 0 points.