Problem A. Impossible to Guess

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Andrew thought up a permutation $p_1, \ldots, p_n$ of n first positive integers and said it to Pavel so that he wouldn't forget it. This was not a very wise decision because Pavel suffers from memory disorders and can't remember the exact permutation, but for each its subsegment $[l, r]$ Pavel can quickly say a set of numbers $p_l, \ldots, p_r$ in ascending order.

Alex is very interested in the permutation thought up by Andrew so he wants to find it out from Pavel in secret from Andrew. Alex hasn't much time so he can ask only $\lceil \frac{n}{2} \rceil$ questions (where $\lceil x \rceil$ means the number x rounded up). Help him to find Andrew's permutation out.

Input

This is the interactive problem. Here your program must in the process of the solution exchange information with the jury's program. Please note that after outputting each message your program must flush the buffer, so that your output reaches the jury's program: for instance, calls of «`fflush(stdout)`» or «`cout.flush()`» do it in C++, «`System.out.flush()`» in Java, «`flush(output)`» in Pascal.

At the beginning the single integer n ($1 \le n \le 100$) — the length of the permutation — is sent your program. Then you can ask at most $\lceil \frac{n}{2} \rceil$ questions and after that give the final answer.

To ask a question you should output a letter «`Q`» first, and then two integers, separated by a space: l and r ($1 \le l \le r \le n$) — the indices of the first and the last elements of the subsegment you want to ask.

Jury's program will answer each of such questions, sending $(r - l + 1)$ integers separated by spaces: the elements of the permutation $p_l, \ldots, p_r$, sorted in the ascending order.

To give the final answer, you should output a letter «`A`» first, then n elements of the permutation separated by spaces — $p_1, \ldots, p_n$. After that you program must be terminated.

Examples

stdin	stdout
3 3 1 3	 Q 1 1 Q 1 2 A 3 1 2

Note

Emply lines in the sample are given only for convenience, to make it clear in which order the messages are written. When solving the problem you must not output empty lines and jury's program won't output empty lines too.