Problem I. Imprecise Permutation Sort

Time limit: 10 seconds
Memory limit: 512 megabytes

This is an interactive problem.

A permutation $a[1], a[2], \ldots, a[n]$ of integers from 1 to n is hidden from you.

Your task is to sort it in ascending order by comparing and swapping pairs of elements. This problem could be pretty easy, but the jury member responsible for the problem was too concentrated on floating-point arithmetic in problems G and J and implemented an "imprecise" comparator:

- if $\frac{|a[i]-a[j]|}{max(a[i],a[j])} \le 0.01$, then return 0;
- otherwise, if $a[i] < a[j]$, then return -1;
- otherwise, return 1.

Your program can make queries to compare any two elements with this comparator, or to swap any two elements. After each swap, it will be told whether the permutation became sorted.

Sort a permutation of size up to 16 384 using no more than 300 000 queries.

Interaction Protocol

Receive an integer n from the jury's program — the size of the permutation ($1 \le n \le 16\,384$). Then print queries and receive responses from the jury's program. After each query the output should be flushed and then a single integer should be read — the response to that query.

Comparison queries have a format "`C i j`", and swap queries have a format "`S i j`", where i and j are indices of two elements ($1 \le i, j \le n$). Making queries with $i = j$ is allowed.

The response to a comparison query is 0 if $a[i]$ "approximately equals" $a[j]$, otherwise: -1 if $a[i] < a[j]$, and 1 if $a[i] > a[j]$.

A swap query swaps values in $a[i]$ and $a[j]$, and the response to a swap query is 1 if after this swap the array became sorted in ascending order, and 0 otherwise.

Your program should exit as soon as it receives 1 as a response to a swap query.

The program should make at least one swap query. For example, if the hidden permutation is already sorted, it can make a query "`S 1 1`", receive a 1, and exit.

Examples

standard input	standard output
4 -1 -1 1 1	 C 1 2 C 2 3 C 3 4 S 3 4
1 1	 S 1 1