Intersegment Activation

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

This is an interactive problem.

There is an array of n cells, numbered from 1 to n. For each pair of integers (i, j), where $1 \le i \le j \le n$, there is a barrier covering all cells from i to j, inclusive. Each barrier is either *active* or *inactive*. A cell is *visible* if there are no active barriers that cover it. Otherwise, the cell is *invisible*.

The state of each barrier is unknown to you. All you can observe is the number of visible cells. But you can flip the state of any barrier: if it's active, it turns inactive, and the other way around. Your task is to make all barriers inactive, so that all cells become visible.

Interaction Protocol

First, read an integer n, denoting the number of cells ($1 \le n \le 10$).

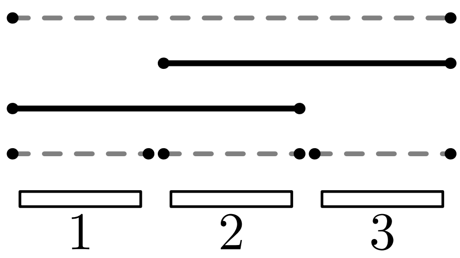
The following interaction will proceed in rounds. Your program should start each round by reading an integer k, denoting the number of currently visible cells ($0 \le k \le n$).

- If $k = n$, then the task is done and your program must exit.
- If $k < n$, you can flip the state of any barrier. On a separate line, print two integers i and j to flip the state of the (i, j) barrier ($1 \le i \le j \le n$). After your query, the next round begins, and your program should read a new value of k.

Your solution must make all cells visible using at most 2500 flips. In the beginning, not all cells are visible ($k < n$ in the first round).

The interactor is not adaptive: in every test, the state of all barriers is chosen before the program execution.

Example

standard input	standard output	Initial state
3		
0		
	2 2	
0		
	2 3	
1		
	1 2	
2		
	2 2	
3		

Note

In the example, initially, only two barriers, $(1, 2)$ and $(2, 3)$, are active. These two barriers cover all three cells, so k is equal to 0 in the first round.

- After flipping the $(2, 2)$ barrier, there are now three active barriers, and still $k = 0$ visible cells.
- After flipping the $(1, 2)$ barrier, cell 1 becomes visible, so now there is $k = 1$ visible cell.
- After flipping the $(2, 3)$ barrier, cell 3 also becomes visible. The only invisible cell now is 2, covered by the only active barrier, $(2, 2)$, and there are $k = 2$ visible cells.
- After flipping the $(2, 2)$ barrier, all barriers are now inactive, and all cells are visible. After reading $k = 3$, the program terminates.