COI '13 #4 Nizovi

Time limit: 1.0s **Memory limit:** 256M

Mergesort is a quick sorting algorithm invented by John Von Neumann in 1945. In the heart of the algorithm lies a procedure that combines two already sorted sequences into a new sorted sequence. In this task you need to write a program which does a similar thing.

Let A and B be sequences of **integers sorted in ascending order**, the sequence A is called *small sequence* and B *large sequence*. As it can be derived from these names, the sequence A contains less or equal number of elements than sequence B, and often the sequence A is **a lot shorter**. The sequence C is obtained by combining sequences A and B in such a way that first we sequentially take all the numbers from sequence A and then, after them, sequentially take all the numbers from sequence B. We use the notation $C[X]$ to denote the X^{th} element of sequence C where X is an integer in the interval from 1 to total number of elements in sequence C.

In the beginning, your program knows only the lengths of sequences A and B, but not the sequence elements themselves. The program must **sort the sequence C in ascending order** using only the following interactive commands:

- `cmp X Y` is a command which compares $C[X]$ and $C[Y]$ returns -1 if $C[X]$ is less than $C[Y]$, 1 if $C[X]$ is greater than $C[Y]$ and 0 if they are equal.
- `reverse X Y` is a command which reverses a subsequence of sequence C between the X^{th} and Y^{th} element (inclusive). For example, if sequence C is equal to $(2, 5, 2, 3)$ then the `reverse 2 4` command will alter it so it becomes $(2, 3, 2, 5)$.
- `end` is a command that denotes the end of interaction and your program should call it when the sequence C is sorted in ascending order.

The cost of command `reverse X Y` is equal to the length of the reversed subsequence (in other words, $Y - X + 1$), whereas the rest of the commands do not have a cost. Write a program that will sort the sequence C obtained by combining sorted sequences A and B as described above using **at most $100\,000$ commands in total** (including the command `end`) and additionally, so that the total cost of all `reverse` commands is **at most $3\,000\,000$**).

Interaction

Before interacting, you must read two integers from the standard input, N_A and N_B $(1 \le N_A \le 1\,000, N_A \le N_B \le 1\,000\,000)$ - lengths of sequences A and B.

Afterwards, your program can give commands by outputting using standard output. Each command must be output in its own line and has to be in the exact form of one of the three commands described in the task. Regarding commands `cmp` and `reverse`, X and Y have to be integers less than or equal to $N_A + N_B$, and regarding the command reverse, it additionally must hold that X is less than or equal to Y.

After command `cmp`, your program must input one integer using the standard input - the command result. After other commands, your program must not input anything. After outputting the command `end`, your program needs to flush the standard output and finish executing.

Please note that you need to flush stdout after each command, or interaction may halt. In C++, this can be done with `fflush(stdout)` or `cout << flush` (depending on whether you use `printf` or `cout`). In Java, this can be done with `System.out.flush()`. In Python, you can use `sys.stdout.flush()`.

Constraints

Subtask	Points	Constraints
1	30	$1 \leq N_A, N_B \leq 50$
2	30	$1 \leq N_A, N_B \leq 500$
3	40	No additional constraints.

Sample Interaction

`>>>` denotes your output. Do not print this out.

```
2 3
>>> cmp 1 4
-1
>>> reverse 2 5
>>> cmp 2 3
0
>>> reverse 4 5
>>> reverse 2 5
>>> end
```

Sample Explanation

Output	Input	Sequence C	Total Cost
	$2\ 3$	$-1, 3, 2, 5, 5$	0
`cmp 2 3`	-1	$-1, 3, 2, 5, 5$	0
`reverse 2 5`		$-1, 5, 5, 2, 3$	4
`cmp 2 3`	0	$-1, 5, 5, 2, 3$	4
`reverse 4 5`		$-1, 5, 5, 3, 2$	6
`reverse 2 5`		$-1, 2, 3, 5, 5$	10

end			