**Assignment on Red Black Tree**

A super-computer named TEUB has many running processes. Each process has a priority x. Two active processes in TEUB cannot have equal priorities. When a process finishes its task, it gets terminated. Sometimes, the programmers of TEUB want to know how many running processes have priorities less than y. The programmers hired you to help them using Red-Black tree.

The input has four types of commands.

* Initiation of a program
* Termination of a program
* Searching for a program
* Find the programs with less priority

**Input**

First line of input shows the total number of commands (N).

Each of the following N commands, has two integers ei and xi.

|  |  |
| --- | --- |
| ei | Meaning |
| 0 | Terminate the program with priority xi. |
| 1 | Initiate a program with priority xi. |
| 2 | Search the program with priority xi. |
| 3 | Find the number of programs with priority less than xi. |

**Output**

First line of input shows the number of output lines.

For each command, you have to print three integers ei ,xi and ri.

ri signifies the result of the corresponding command.

|  |  |
| --- | --- |
| ei | ri |
| 0 | 1 if successful termination.  0 if there is no program with priority xi |
| 1 | 1 if successful initiation  0 if there is already a program with priority xi |
| 2 | 1 if found  0 if not found |
| 3 | The number of programs with priority <xi. |

**Sample I/O**

|  |  |  |
| --- | --- | --- |
| Sample Input | Sample Output | Explanation |
| 11  1 1  1 2  1 3  1 1  0 1  0 4  2 3  2 5  1 1  3 3  3 6 | 11  1 1 1  1 2 1  1 3 1  1 1 0  0 1 1  0 4 0  2 3 1  2 5 0  1 1 1  3 3 2  3 6 3 | Line count  Successful initiation 1  Successful initiation 2  Successful initiation 3  Same priority (1) exists  Successful termination 1  No priority (4) exists  Priority 3 found  Priority 5 not found  Successful initiation 1  2 programs having priority < 3  3 programs having priority < 6 |

**Constraints**

1N105

1xi106

0ei3

Each of the commands has to be answered in logarithmic time.

**More instructions**

* Write the program in such a way to accept input from file
* Write Red-Black tree codes in such a way that it can be reused for other tasks during online evaluation.

**Submission**

* Include only source files
* Do not include executable binaries, input/output files
* Place your files in a folder named 1905XXX
* Zip the folder
* Submit to Moodle after renaming it to 1905XXX.zip