✷ Corona Tracking App ✷

Programming Assignment 3

The experts of the government found out that the virus spreads only if two people have exactly a certain distance *d*; larger or smaller distances are safe they claim. Thus, as a new corona measure, the government declared it illegal to meet at distance exactly *d*.

To control the people and strengten its power, the government introduced an app that tracks the positions of all its inhabitants and sends them every few seconds to a central server.

The server controls a huge signal light in the city center that signalizes the level of danger. If there are two people at distance *d*, the light should be red; otherwise, if there are no people at distance *d*, it should be yellow (there is no green).

Since the whole world view is now one dimensional, every position is described by a single coordinate. Also given that the experts change their mind rather frequently, upon initialization, the server receives the currently forbidden distance *d*—if *d* changes, the server is just restarted.

It is up to you to save the world!

Problem Definition

See below for a formal definition of the input/output format.

Your program receives a positive distance *d* followed by a sequence of *n* events: An event can be either

- that a person appeared at some position, or
- that a person disappeared from some position.

After each event, your program should output

- "red" if there are currently at least two persons at distance exactly *d*, and
- "yellow" otherwise.

Note that initially (before the first event) all positions are empty.

The events are given as a 2D array event[0..n-1][0..1], where event[i][0] is

- "1" if a person appeared at position event[i][1] (on the x-axis), and
- "-1" if a person disappeared from that position.

The output is an array light[0..n-1] of strings, where light[i] is the color directly after event[i].

Send Feedback

Example

Input:

```
d = 4,

event =
[[ 1, 2 ],
 [ 1, 10],
 [ 1, 6 ],
 [-1, 2 ],
 [-1, 6 ],
 [ 1, 9 ],
 [ 1, 14]]
```

Positions:

```
2
2, 10
2, 6, 10
6, 10
10
9, 10
9, 10, 14
```

Pairs of distance 4:

```
no pairs
no pairs
(2, 6) and (6, 10)
(6, 10)
no pairs
no pairs
(10, 14)
```

Output:

```
light =
[ yellow,
  yellow,
  red,
  red,
  yellow,
  yellow,
  red ]
```

Details

You can assume that

- $1 \le n \le 300\,000$,
- $1 \le d \le 10^9$,
- event[i][0] is either -1 or 1 for each i,
- $0 \le$ event[i][1] $\le 10^9$ for each i, and
- at any time, there is at most one person on a single position.

Formally, the input of your program is given via the standard input and your output has to be written into the standard output. The format of the input and the output is described at the end of this page. However, you don't need to worry about it. The following code (one in Python and one in C++) automatically reads the input and writes the output. You can use it as a base of your program! (But you are free to write everything from scratch yourself.)

[Example Code in Python] (material/base.py) [Example Code in C++] (material/base.cpp)

Rules

- The **deadline is on Friday, 28 May** 1 minute before midnight (23:59 Polish time zone).
- Hint: use an appropriate **dictionary** and a clever idea! You will need to use among others the remove/delete operation of your dictionary. The dictionaries from the standard libraries (Python/C++) suffice.
- Discussing the algorithm and the implementation with other people is not allowed.
- We will compare all solutions and take measures if we suspect plagiarism.
- Only C++ and Python are allowed.
- Using standard libraries is allowed—including numpy for Python (anything that works on the Szkopuł webserver is allowed).
- Upload your program on the Szkopuł webserver anytime before the deadline and how often you want. Shortly after each upload, you will see which tests you passed or failed and what your

Send Feedback

total score is.

- **Your last submission counts! Hence, please take care that your last submission is the one with the most points.**
- The initial tests (Test0a and Test0b) do not give you any points. Their input and output is provided below. Test0a corresponds to the example instance above.
- The inputs of all the other tests are secret.
- You get 0 points for a test if your program crashes, has the wrong output, is too slow or takes too much memory.
- To get the full score (100 points), your program has to pass all the tests within the respective time and memory limits. (For all the tests, the time limit is 10s and the memory limit is 64MB.) Note that this is only possible if the underlying algorithm has a sufficiently small time and memory complexity.
  - Tests 1-3: focus on correctness; can be passed easily
  - Tests 11-14: focus on running time
  - Tests 21-27: focus on memory (and time)

  There are two time limits: If you are below the first time limit, you get full points. If you are between the two time limits, you get less points. If you are above the second time limit, you get 0 points. The second time limit is twice as large as the first one. In Szkopuł, only the value of the second time limit is displayed.
- **Activity points**: You can earn up to two activity points by submitting a test on Moodle. The test should contain an input and a corresponding output in the format as specified below. It will be published to help other students.
- In the code above, you find in the comments some code by which you can read the input from a given file (input.txt) and write the output to another file (your-output.txt) and compare it to the intended output (output.txt). By this, you can use the test data submitted by other students to test your code.

Format

Input:

- first row: n d
- next n rows: two integers; the first one is either -1 or 1, the second one is nonnegative (these n rows correspond to the array "event")

Output: n rows each consisting of either the string "red" or "yellow".

Tests 0a and 0b have the following input and output. (Note that Test 0a corresponds to the example above.)

- [Input Test 0a] (material/input0a.txt) [Output Test 0a] (material/output0a.txt)
- [Input Test 0b] (material/input0b.txt) [Output Test 0b] (material/output0b.txt)