Ball Clock Assignment

Movement has long been used to measure time. For example, the ball clock is a simple device which keeps track of the passing minutes by moving ball-bearings. Each minute, a rotating arm removes a ball bearing from the queue at the bottom, raises it to the top of the clock and deposits it on a track leading to indicators displaying minutes, five-minutes, and hours. These indicators display the time between 1:00 and 12:59, but without 'a.m.' or 'p.m.' indicators. Thus 2 balls in the minute indicator, 6 balls in the five-minute indicator and 5 balls in the hour indicator displays the time 5:32.

Unfortunately, most commercially available ball clocks do not incorporate a date indication, although this would be simple to do with the addition of further carry and indicator tracks. However, all is not lost! As the balls migrate through the mechanism of the clock, the order they are found in can be used to determine the time elapsed since the clock had some specific ordering. The length of time which can be measured is limited because the orderings of the balls eventually begin to repeat. Your program must compute the time before repetition, which varies according to the total number of balls present.

Operation of the Ball Clock

Every minute, the least recently used ball is removed from the queue of balls at the bottom of the clock, elevated, then deposited on the minute indicator track, which is able to hold four balls. When a fifth ball rolls on to the minute indicator track, its weight causes the track to tilt. The four balls already on the track run back down to join the queue of balls waiting at the bottom in reverse order of their original addition to the minutes track. The fifth ball, which caused the tilt, rolls down to the five-minute indicator track. This track holds eleven balls. The twelfth ball carried over from the minutes causes the five-minute track to tilt, returning the eleven balls to the queue, again in reverse order of their addition. The twelfth ball rolls down to the hour indicator. The hour indicator also holds eleven balls, but has one extra fixed ball which is always present so that counting the balls in the hour indicator will yield an hour in the range one to twelve. The twelfth ball carried over from the five-minute indicator causes the hour indicator to tilt, returning the eleven free balls to the queue, in reverse order, before the twelfth ball itself also returns to the queue.

Guidelines

The exercise may be completed in Java, Go, Python, or JavaScript. You are welcome to do it in more than one of the languages.

Note: If you are doing this assignment as part of an application for employment, check with your recruiter to determine what language(s) to use.

No permutation or LCM algorithms are allowed. A full simulation is required. Please ensure that your code moves each ball.

Last Updated: 10/5/17 12:59:00 PM

Implementation

Valid numbers of balls are in the range 27 to 127. Ball inputs outside of this range should produce an error message. Clocks must support two modes of computation. Each mode is described below, along with sample inputs and outputs. For each mode, a final line of output should indicate the amount of time (in both milliseconds and seconds) it took for the clock to complete that run.

Mode 1 (Cycle Days)

The first mode takes a single parameter specifying the number of balls and reports the number of balls given in the input and the number of days (24-hour periods) which elapse before the clock returns to its initial ordering.

An input of `30` yields the following output:

```
30 balls cycle after 15 days.
Completed in x milliseconds (y.yyy seconds)
```

An input of `45` yields the following output:

```
45 balls cycle after 378 days.
Completed in x milliseconds (y.yyy seconds)
```

Mode 2 (Clock State)

The second mode takes two parameters, the number of balls and the number of minutes to run for. If the number of minutes is specified, the clock must run to the number of minutes and report the state of the tracks at that point in a JSON format.

An input of `30 325` yields the following output:

```
{"Min":[],"FiveMin":[22,13,25,3,7],"Hour":[6,12,17,4,15],"Main"
  [11,5,26,18,2,30,19,8,24,10,29,20,16,21,28,1,23,14,27,9]}
```