POMODORO INTERVALS

8DCC

1. The Pomodoro Technique

The Pomodoro Technique is a time management method developed by Francesco Cirillo in the late 1980s. It uses a kitchen timer to break work into intervals (*pomodoros*), typically 25 minutes in length, separated by short breaks. After 4 pomodoros are done, a bigger break is taken.

For this example, these will be the specifics timings:

(1) Work for 25 minutes.
(2) Rest for 5 minutes.
(3) Go to step one, until 4 pomodoros have passed.
(4) Once the 4 pomodoros are done, rest for 30 minutes.

With this, we know that the first short break will start at minute 25 and end at minute 30, and the first big break will start at minute 115 ($30 \cdot 4 - 5$) and end at minute 140.

2. The problem

Given a number of minutes, how do we know if we should be resting or working when that time has passed?

You can calculate if a minute is in a short or long break by performing a modulus with the end of the break, and checking if that value is greater or equal than the start of the break.

$$\text{inShortBreak}(m) = m \bmod 30 \geq 25$$
$$\text{inLongBreak}(m) = m \bmod 140 \geq 115$$

However, the fifth pomodoro, the one after the first long break doesn't start aligned to the 30 minute boundary ($140 \bmod 30 \neq 0$).

At first I thought about calculating the Least Common Multiple (LCM) of 140 and 30 (which is 420), but that won't work here.

3. The solution

The solution was actually quite simple. You can think of the break cycle as a big 145-minute cycle, and calculate the minutes relative to that, just like we did before.

$$\text{inShortBreak}(m) = (m \bmod 145) \bmod 30 \geq 25$$
$$\text{inLongBreak}(m) = (m \bmod 145) \bmod 140 \geq 115$$

Or, in C:

```
bool inPomodoro(int minutes) {
    int minutes_in_clycle = minutes % ((25 + 5) * 3 + 25 + 30);
    bool in_long_rest = (minutes_in_clycle % 140 >= 115);
    bool in_short_rest = (minutes_in_clycle % 30 >= 25);
    return !in_long_rest && !in_short_rest;
}
```