

Analysis: FreeCell Statistics

The first thing to notice in this problem is that both P_D and P_G can range from 0 to 100, so let's start by taking care of some easy cases. If our unnamed player has won 100% of his total games ($P_G = 100$) but not won 100% of his games today ($P_D < 100$), then something has clearly gone wrong with the calculator. Similarly, if $P_G = 0$ but $P_D > 0$, something is wonky.

The cases where $P_D = P_G = 0$ or $P_D = P_G = 100$ are also easy, and the answer for both of them is "Possible" — they mean that all the games ever ended with the same result (loss or win, respectively).

The trick now is that once we ruled out the case of P_G being zero or a hundred, we do not need to worry about it any more! Indeed, assume we have a solution that gives the correct daily percentage of wins, and it consists of W games won and L games lost today. To get a global win percentage of P_G , we can, for instance, assume we won a total of $(W + L) * P_G$ and lost a total of $(W + L) * (100 - P_G)$. As $1 \leq P_G \leq 99$, the numbers are greater than W and L , respectively, so they are possible to achieve.

The small data set can now be solved by brute force by simply trying all possible values of D from 1 to N and all possible number of games won from 0 to D , and seeing if any of them results in exactly P_D percentage of games being won.

Solving the large data set can require some maths. One way to solve the problem is to directly solve for the minimum number of games we would need to play in order to get a win percentage of P_D and simply verify that this number is $\leq N$. If we let W be the number of games we have won today, then we want to solve $W / D = P_D / 100$ for the minimum value of D such that W is integral.

From here, it is easy to see $D = 100 * W / P_D$. Thus, if we want to minimize D , then we need to find the smallest value W such that the right hand side is integral. In order to do this, we divide 100 and P_D by their [greatest common divisor](#) (let's call this value G) so that they are relatively prime and W is minimal when it is P_D / G . Plugging this in and cancelling terms tells us that $D = 100 / G$ is the fewest number of games we must be play.

A simpler way to solve this problem is by brute force. We can just try all possible values of D from 1 to N and check if any of them could result in exactly P_D percentage of games being won by checking that $D * P_D = 0 \pmod{100}$, stopping the loop the first time we find a candidate value of D or we exceed N games. While at first this solution appears to be $O(N)$ and would time out for the large data set, this loop will in fact run *at most* 100 times, regardless of the value of N , so this solution will be plenty fast enough. Coders who noticed this simpler solution early were rewarded with very fast submission times.