

Solutions

TOMO

Keeping track of the entire number displayed isn't very feasible, but we can get away with keeping only the rightmost K digits of the number, where K is the number of digits in the given number C .

We can easily retrieve the rightmost K digits of a number by calculating the remainder of dividing that number by pot10 , where pot10 is the smallest power of 10 larger than C . The first part of the solution calculates pot10 .

We then simulate operations on the calculator (modulo pot10), waiting for the number C to appear. Note that at most pot10 iterations are needed here, since, according to the pigeonhole principle, some number will have appeared at least twice after so many iterations, meaning we got stuck in a loop and there is no solution.

ZMIJA

A direct simulation of the rules in the problem description will do. We process events in order (an event is the snake changing its direction) and simulate all moves between two events as follows:

- if the snake's head leaves the board, the game is over
- if the snake's head is on a field with an apple, eat the apple and lengthen the snake
- if the snake's body is still on the field the head has entered, the game is over

After the last event, we let the snake bite itself or fall off the board.

AUTOPUT

Consider the function $f(k, x)$, the smallest possible amount of time in which we can reach position x , having till now built exactly k tunnels or viaducts.

Let $\text{right}(x)$ correspond to the first position to the right of x where the height is the same as the height of position x .

We can now use $f(k, x)$ to calculate the following values:

$f(k, x+1)$ - if we proceed to position $x+1$ without building a tunnel or viaduct

$f(k+1, \text{right}(x))$ - if we build a tunnel or viaduct starting at position x and ending at position $\text{right}(x)$

We can use dynamic programming to calculate $f(k, x)$ for all values of k and x . Note that we don't have enough memory available to hold the entire table, but we can get away with storing only two rows (for k and $k+1$) at all times.