

# Loop Detection

## Problem description

We define the following transformation on an integer number: If you have a number

1525856328535326023410029108,

you look at the last two digits, which form in this case the number 08 (or here just 8). Next you cut the number after its 8th digit and transpose both parts getting

2853532602341002910815258563.

Finally, you increase this number by one resulting in

2853532602341002910815258564.

The numbers will always have exactly 28 digits and—if necessary— are padded with zeros on the left or have the first digit chopped off should the number have 29 digits (which can only happen if the number consists solely of 9s). If the last two digits indicate that you should cut outside the number, the transposition is doing nothing because you cut off an empty piece. Even in this case the number does not stay the same as we are adding one to it.

If you start with a number  $k$  and transform it repeatedly you will eventually enter a cycle that repeats itself indefinitely. Your task is to find out how long that cycle is.

Your program reads a single number  $k$  as its input and should output a single number, i.e., the length of the cycle into which you run when repeatedly transforming  $k$ .

## Sample input/output

Input	Output
1525856328535326023410029108	2220930