F. Palindrometer

Source file: **`pali.{c, java, cpp}`**
Input file: **`{stdin, System.in, cin}`**
Output: **`{stdout, System.out, cout}`**

While driving the other day, John looked down at his odometer, and it read **100000**. John was pretty excited about that. But, just one mile further, the odometer read **100001**, and John was REALLY excited! You see, John loves palindromes — things that read the same way forwards and backwards. So, given any odometer reading, what is the lest number of miles John must drive before the odometer reading is a palindrome? For John, every odometer digit counts. If the odometer read was **000121**, he would not consider that a palindrome.

Input

There will be several test cases in the input. Each test case will consist of an odometer reading on its own line. Each odometer reading will be from **2** to **9** digits long. The odometer in question has the number of digits given in the input — so, if the input is **00456**, the odometer has 5 digits. There will be no space sin the input, and no blank lines between input sets. The input will end with a line with a single **0**.

Output

For each test case, output the minimum number of miles John must drive before the odometer reading is a palindrome. This may be **0** if the number is already a palindrome. Output each integer on its own line, with no extra spaces and no blank lines between outputs.

Sample Input	**Sample Output**

`100000`
`100001`
`000121`
`00456`
`0`
	`1`
	`0`
	`979`
	`44`