

# Runaround Numbers

## Problem Description

Runaround numbers are integers with unique digits, none of which is zero (e.g., 81362) that also have an interesting property, exemplified by this demonstration:

- If you start at the left digit (8 in our number) and count that number of digits to the right (wrapping back to the first digit when no digits on the right are available), you'll end up at a new digit (a number which does not end up at a new digit is not a Runaround Number). Consider: 8 1 3 6 2 which cycles through eight digits: 1 3 6 2 8 1 3 6 so the next digit is 6.
- Repeat this cycle (this time for the six counts designed by the '6') and you should end on a new digit: 2 8 1 3 6 2, namely 2.
- Repeat again (two digits this time): 8 1
- Continue again (one digit this time): 3
- One more time: 6 2 8 and you have ended up back where you started, after touching each digit once. If you don't end up back where you started after touching each digit once, your number is not a Runaround number.

Given a number **M** (that has anywhere from 1 through 9 digits), find and print the next runaround number higher than **M**, which will not exceed 10,000,000 for the given test data.

## Input

A single line with a single integer, **M**.

## Output

A single line containing the next runaround number higher than the input value, **M**.

## Sample Input

81361

## Sample Output

81362

## Algorithm Template

1. How to check whether all digits are used?
2. How to keep track on current position (which digit are you processing now)?
3. How to repeat the cycle?