

UCF Local Contest (Qualifying Round) — September 3, 2022

Decimal XOR

filename: dexor

Difficulty Level: Easy

Time Limit: 5 seconds

The binary operation XOR accepts two binary digits as input and outputs a binary digit: if both input digits are 0 (or both are 1), the output is 0; otherwise the output is 1. We can look at this as: if both input values are low (or both are high), the output is 0; otherwise the output is 1.

Decimal numbers have several digits and each digit can be one of 10 values (0-9). We define the operation DEXOR (XOR of two decimal numbers) as follows: we DEXOR two decimal digits at a time; the two decimal digits at 1st position are DEXOR'ed, the two decimal digits at 10th position are DEXOR'ed, the two digits at 100th position are DEXOR'ed, etc. When DEXOR'ing two decimal digits, the result digit is 0 if both digits are too small (≤ 2) or both digits are too large (≥ 7); the result digit is 9 otherwise.

The Problem:

Given two decimal numbers, compute their DEXOR.

The Input:

There are two input lines, each line providing a decimal number between 0 and 999,999 (inclusive). Assume that there will not be extra leading zeroes in an input number, i.e., there will not be extra zeroes at the beginning of a number in the input.

The Output:

Print the DEXOR of the two decimal numbers. When DEXOR'ing two decimal numbers, if one has fewer digits, it should be considered as having zeros on the left to make both numbers having the same number of digits. The result should have as many digits as the larger number.

Sample Input

Sample Output

22776 15954	09099
29 18908	09900

Note that, in the second Sample, 29 should be treated as 00029 so that it will have the same number of digits as the second number (so that they can be DEXOR'ed digit-by-digit).