

Euler 0033

The Problem:

The fraction $49/98$ is a curious fraction, as an inexperienced mathematician in attempting to simplify it may incorrectly believe that $49/98 = 4/8$, which is correct, is obtained by cancelling the 9s.

We shall consider fractions like, $30/50=3/5$, to be trivial examples.

There are exactly four non-trivial examples of this type of fraction, less than one in value, and containing two digits in the numerator and denominator.

If the product of these four fractions is given in its lowest common terms, find the value of the denominator.

Considerations and Approach:

Naively... we can iterate through the denominator up to 100, and then iterate through the numerator up to the denominator.

To find the solution we can check every number such that the denominator and numerator have an equal digit, and that the result of the fraction is the same as the digit reduced fractions.

We skip every multiple of 10 over 10 since numerator would make the reduced single digit fraction = 0, which won't ever be the case of the base fraction, or it will be undefined, which is problematic. We can start from 11 every time since we can't digit reduce a single digit number.

The Code:

```

import math

upper_limit = 100
cycles = 0
non_trivial_den = []
non_trivial_num = []

#denominator and numerator
for denominator in range(11,100):
    for numerator in range(11,denominator):
        if denominator % 10 == 0 or numerator % 10 == 0:
            pass
        else:
            #iterate through the denominator to find numbers that are equivalent
            counter = 0
            for den_index in range(len(str(denominator))):
                for num_index in range(len(str(numerator))):
                    cycles += 1
                    if int(str(numerator)[num_index]) == int(str(denominator)[den_index]):
                        new_num = int(str(numerator)[:num_index] +
str(numerator)[(num_index+1):])
                        new_den = int(str(denominator)[:den_index] +
str(denominator)[(den_index+1):])
                        if numerator/denominator == new_num/new_den:
                            print(denominator, numerator, new_num, new_den)
                            non_trivial_den += [new_den]
                            non_trivial_num += [new_num]

print(cycles, math.prod(non_trivial_den), math.prod(non_trivial_num))

```

Revision #2

Created 2026-03-02 20:44:39 UTC by Maxwell

Updated 2026-03-02 20:46:44 UTC by Maxwell