Decimals to fractions

For the integers between 2 and 10, finding decimal equivalents for the reciprocals, fractions like 1/2 and 1/5 and so on, is pretty simple (0.5 and 0.2, respectively).

Even 1/4 and 1/8 (0.25 and 0.125) aren't bad.

1/3 and its multiples have the issue of repeating 3's.

```
     0.333333
 3 / 1.000000
```

and we are left to ponder how to interpret the result that $3 \cdot 1/3 = 1$, but $3 \cdot 0.3333 = 0.9999$ repeating, if they are the same. Of course the answer gets to the problem of how to define real numbers, not just rationals but also π or $\sqrt{2}$.

The only moderately complicated result is for 7. Computing 1/7 by long division:

```
     0.142857
 7 / 1.000000
       7
     ---
       30
       28
       --
        20
        14
        --
         60
         56
         --
          40
          35
          --
           50
           49
           --
           10
```

For the next step, we have 10 as the dividend so the result is just 0.142857 repeating.

How to convert that decimal back to 1/7?

The trick is to multiply by 10 to whatever power is needed to bring the multiple into alignment with the original.

```
          f =      0.142857
1000000 . f = 142857.142857
 999999 . f = 142857
```

The repeat length has 6 places and starts right after the decimal point so we need 10^6.

```
f = 142857/999999
```

The last step is to check our work. We factor both the numerator and the denominator. I get

```
142857 = 3.3.3. .11.13.37
------   -----------------
```

```
999999   3.3.3.7.11.13.37
```

which works out correctly. It is equal to 1/7 as expected.

Here is a short Python function I will use to find factors (not very efficient since it tests more than primes):

```
def f(n):
    for i in range(2,n+1):
        while n % i == 0:
            print(i)
            n = n/i
        if n == 1:
            return
```

When I do `f(999999)` I get those factors.

An interesting thing about this problem is that the successive dividends are: $10, 30, 20, 60, 40, 50$ followed by 10 again. So if we compute say, decimal (3/7), we obtain a cyclically permuted version of the same pattern, just starting with `428571` .

All the possible numerators are there, 1 through 6.

If we have a multiple of 7 in the denominator, like

```
>>> 1/14
0.07142857142857142
```

we have the same pattern with something extra in front. Using the same trick with many fewer zeros

```
      f = 0.07142857
100 . f = 7.142857 = 7 + 1/7
```

No subtraction, just do $7 + 1/7 = 50/7$, then multiply by 1/100 to obtain $f = 1/(2 \cdot 7) = 1/14$.

Some simple fractions have quite complicated decimals. Consider 1/23. I use the Python decimal module to get enough precision:

```
import decimal
decimal.getcontext().prec = 50
d = 1/decimal.Decimal(23)
d
```

output:

```
Decimal('0.043478260869565217391304347826086956521739130434783')
```

22 digits in the repeat. Can you see it?

```
>>> (int(1e22) * d) - d
Decimal('434782608695652173913.00000000000000000000000000000')
>>> int(_)
434782608695652173913
```

So then we just check the result:

```
>>> x = 434782608695652173913
>>> y = 9999999999999999999999
>>> y/x
23.0
>>> y % x
0
```

These two integers must have all the same factors except for an extra factor of 23 on the bottom. But those are two pretty large numbers to factor.

Based on what we saw above, we're going to have at least two factors of 3 and then one of 11, and of course there'll be at least one 23 if we work with the denominator, so dividing I get

```
>>> n = decimal.Decimal(9999999999999999999999)
>>> 3*3*11*23
2277
>>> n/_
Decimal('43917435221783047877')
```

Working with this, I find another factor of 11 and what's left is 399249411107118617.

However, then it gets hard! I use the script from above to find that the next factor is 4093.

All the rest will be larger than that. The next is 8779 and then what's left is 11111111111. The next factor is 21649, and it leaves a remainder of 513239, which is the last prime. That's fairly exhausting.

To recap

```
>>> 3*3*11*11*23*4093*8779*21649*513239
9999999999999999999999
```

The numerator is the same except it's missing the factor of 23.

To be sure there's no mistake, I look at a table of primes. They're all there.

`https://primes.utm.edu/lists/small/millions/`

It's interesting that the repeat length for 1/7 is 6 and that for 1/23 is 22.

The length of the repeat for 1/17 is also 16, that for 1/19 is 18 and so on. Yet the repeat for 1/11 is only 2 and for 1/13 it's only 6.

There's an interesting discussion in the second answer here

`https://math.stackexchange.com/questions/377683/length-of-period-of-de`

In general, the repeat length of $1/n$ cannot be longer than $n-1$. Often it is equal to that but sometimes not. The reason is related to Euler's totient function.