Analysis – Ian Rosner

The output of the marbles program is a result of **factors**. In order for a cup to end up with a marble upon completion of the full sequence, it has to be hit an odd amount of times. Let’s take a look at cup 36.

1. First pass – filled (every cup is filled). 1 x 36
2. Second pass – emptied (every other cup is hit). 2 x 18
3. Third pass – filled (every third cup). 3 x 12
4. Fourth pass – emptied (every 4th cup). 4 x 9
5. Sixth pass – filled (every 6th cup – root factor). 6 x 6
   1. This is the half-way point. Now comes the reverse of the previous factors.
6. Ninth pass – 9 x 4 – empty
7. 12th pass – 12 x 3 – fill
8. 18th pass – 18 x 2 – empty
9. 36th pass – 36 x 1 – fill

Now let’s take a quick glance at a number which did not appear in the program output, the number **6**.

1. 1st pass – filled. 1 x 6
2. 2nd pass – emptied. 2 x 3

(Half way – no root factor separating the two halves).

1. 3rd pass – filled. 3 x 2
2. 6th pass – emptied. 6 x 1

The key here is that root factor. The root factor only occurs once – (x \* x) will only ever be hit once, whereas (x \* y) will also be hit on pass (y \* x). The root factor adds the odd offset that is needed for a cup to end with a marble.

For the primes program, a similar concept is in place. A prime number can only be factored as 1 \* itself (and itself \* 1). Therefore, we can count on a prime number only ever being hit twice in the marble simulation. Instead of toggling the marble status, we add a marble to keep count of hits. Any cup with 2 marbles is a prime.

Quick demonstration for the number **7**:

1. 1st pass – marble added. 1 x 7
2. 7th pass – marble added. 7 x 1

… as opposed to a number such as **4**:

1. 1st pass – marble added. 1 x 4
2. 2nd pass – marble added. 2 x 2
3. 4th pass – number added. 4 x 1