Starred exercises are optional.

1. Read the description of public key cryptography and Sections 1–3.5, 4.1, 5.1, and 5.2 of RSA, on wikipedia. Describe the role of 3 of the following algorithms/results from elementary number theory, in RSA:
   a) Euclid's gcd algorithm;
   b) Bézout's lemma;
   c) Fast exponentiation;
   d) Chinese remainder theorem; and
   e) Fermat's little theorem.

   (∗) if you describe all 5 of them.

2. Compute the following:
   - the gcd of $\frac{2^3 \cdot 3^1 \cdot 2^5 \cdot 5^4}{5^3}$ and $2 \cdot 3^7 \cdot 5^2$;
   - integers u and v such that $2 = u \cdot 60 + v \cdot 14$;
   - the inverse of 9 modulo 11; and
   - a natural number $0 \leq x < 9 \cdot 11$ such that $x \equiv 4 \pmod 9$ and $x \equiv 5 \pmod{11}$.

3. For each of the following, describe how to compute it efficiently, and compute it.
   - the inverse of 2016 modulo 2017;
   - 2015^{2016};
   - 2015^{2016} mod 2017; and
   - $2014^{(2015^{2016})}$ mod 2017.

   You may use that 2017 is a prime number, and you may make use of the following Haskell implementation `fe` of fast exponentiation:

```
fe :: Integer -> Integer -> Integer
fe a n = if n > 1 then high * low else low where
  high = (fe a (n `div` 2))^2
  low = if (n `mod` 2) == 1 then a else 1
```

   Hint: Think for each 'modulo'-item whether/how FLT can be used.

4∗) Let $\sim_1$ and $\sim_2$ be equivalence relations on a set A. Is $(\sim_1)^{-1}$ an equivalence relation? If so, show this. If not, give a counterexample. Same question for $\sim_1 \cup \sim_2$, for $\sim_1 \cap \sim_2$, and for $\sim_1^*$.