1. The RSA algorithm

1. Let \( p = 3, \ q = 5 \) and \( n = pq = 15 \). Suppose \( c = 3 \). Find the smallest positive integer \( c' \) such that \( c c' \equiv 1 \mod \phi(n) \).

2. Suppose that you tell people to use the parameters \( n = pq \) and \( c > 0 \) in the RSA cryptosystem. Thus they are supposed to first convert each message they want to send you into a sequence of residue classes \( [a] \in (\mathbb{Z}/n)^* \). For each such \( [a] \), they should broadcast \( [b] = [a]^c = a^c \mod n \).

Here one can use the binary expansion \( c = c_0 + c_1 \cdot 2 + \cdots + c_m \cdot 2^m \) with \( c_i \in \{0, 1\} \) to compute

\[
[b] = [a]^c = [a]^c_0 \cdot ([a]^2)^{c_1} \cdots ([a]^{2^m})^{c_m}
\]

efficiently by finding the \( [a]^{2^j} = ([a]^{2^{j-1}})^2 \) by repeated squarings and reductions mod \( n \). You use your knowledge of \( p \) and \( q \) in \( n = pq \) to find a positive integer \( c' \) such that \( cc' \equiv 1 \mod \phi(n) \), where \( \phi(n) = (p - 1)(q - 1) \). Then \( cc' = 1 + m\phi(n) \) for some \( m > 0 \). On hearing \( [b] = [a]^c \), you recover \( [a] \) using

\[
[a]^{cc'} = [a]^{1 + m\phi(n)} = [a]([a]^{\phi(n)})^t = [a]
\]

where \( [a]^{\phi(n)} = 1 \) because of Euler’s Theorem.

When \( n = 15 \) and \( c = 3 \), write down a table with the following rows:

i. The first row is a list of the residues \( [b] \in (\mathbb{Z}/15)^* \).

ii. The second row is the corresponding decoded message \( [b]^{c'} \).

3. Suppose now that \( p \) and \( q \) are arbitrary distinct odd primes. Let \( c \) be any positive integer which is relatively prime to \( \phi(pq) = (p - 1)(q - 1) \). The object of this exercise is to show that there are at least 4 residue classes \( [a] \in (\mathbb{Z}/pq)^* \) such that \( [a]^c = [a] \). (These \( [a] \) would not be changed by the RSA cryptosystem, and thus should probably not be used as messages!)

i. Show that \( c \) must be odd.

ii. Suppose \( [a] \in (\mathbb{Z}/pq)^* \) and that \( a \equiv \pm 1 \mod p \) and \( a \equiv \pm 1 \mod q \). Show that \( [a]^c = [a] \) in \( (\mathbb{Z}/pq)^* \).

iii. Suppose \( \epsilon_1, \epsilon_2 \in \{1, -1\} \). Show that there is an integer of the form \( \epsilon_1 + pm \) for some \( m \in \mathbb{Z} \) such that \( \epsilon_1 + pm \equiv \epsilon_2 \mod q \). Then use (ii) to show that there are at least four \( [a] \in (\mathbb{Z}/pq)^* \) such that \( [a]^c = [a] \).

iv. Are there \( p, q \) and \( c \) as above such that there are exactly 4 residue classes \( [a] \in (\mathbb{Z}/pq)^* \) such that \( [a]^c = [a] \)?
2. Continued fractions and leap years

The number of days in a year is 365.242199 to six decimal places. This problem has to do with using continued fractions to devise a more accurate system of leap years than the current one.

4. The current leap year system starts with a calendar of 365 days. A leap day is added every 4 years, but every 100 years one does not add a leap day. Every 400 years, one puts back in the leap day. Calculate the total number $N$ of days this system counts every 400 years. The number

$$
\frac{N}{400} - (365.242199)
$$

is then the average fractional number of days by which the current system is incorrect every year (to 6 decimal places).

5. Calculate convergent $p_4/q_4$ of the continued fraction of 365.242199. Here the 0th convergent is $p_0/q_0 = 365/1$. (You should end up with $q_4 = 128$). You can use Maple to do this. Look up “number theory” with the help menu of Maple and then look up the information connected with the “cfrac” function.

6. Write $p_4/q_4$ from part 5 in the form

$$
\frac{p_4}{q_4} = 365 + \frac{n}{q_4}
$$

for some $n$. Devise a system of leap years which will produce $365 \cdot q_4 + n = p_4$ days every $q_4$ years. The number

$$
\frac{p_4}{q_4} - (365.242199)
$$

is the average fractional number of days by which this system will be incorrect every year. Compare this number to the result of part (4).

3. Infinite continued fractions

Suppose $n \geq 1$ is an integer. Find the infinite continued fraction for $\sqrt{n^2 + 1}$.

Hint:

$$
\sqrt{n^2 + 1} = n + (\sqrt{n^2 + 1} - n) = n + \frac{1}{\sqrt{n^2 + 1} - n}
$$