1. **DIVMOD algorithm for any n**

   We saw in class a DIVMOD algorithm which computed \((n \text{ DIV } d, n \text{ MOD } d)\) whenever \(n, d > 0\). Write a revised algorithm that works for any integer \(n\) (both positive and negative) and \(d > 0\). Use pseudo-code.

2. **Paying a price using only 3 and 5 cent coins**

   (a) Implement the algorithm we learned in class to pay price \(p\) using only 3 cent and 5 cent coins (in your favorite programming language). Attach a printout of your code.
   
   (b) What is the time complexity of the algorithm, as a function of \(p\)?
   
   (c) Can you find a faster algorithm? Or do you think that this is the fastest possible? In any case, justify your answer.

3. **Modular arithmetics:**

   Let \(n \geq 2\) and \(x \neq 0 \mod n\). The inverse of \(x \mod n\) is a number \(y\) such that \(x \cdot y = 1 \mod n\). An inverse may not always exist; when it exists, we denote it by \(x^{-1} \mod n\).

   (a) Compute the inverses of 1,...,6 mod 7.
   
   (b) Prove that if \(n\) is prime, then any \(x \neq 0 \mod n\) has an inverse. Hint:

      (i) Show that the function \(f: \{0,1,...,n-1\} \rightarrow \{0,1,...,n-1\}\) defined by \(f(a) = ax \mod n\) is injective.

      (ii) Show that then, as the domain and range have the same size, it then must also be surjective; in particular, 1 is in its image.

   (c) Prove that if \(n\) is not prime, then there exists some \(x \neq 0 \mod n\) which has no inverse. Hint:

      show that there exist \(x, y \neq 0 \mod n\) such that \(x \cdot y = 0 \mod n\); and that this implies that \(x\) has no inverse mod \(n\).

4. **Prime factorization:**

   Let \(n \geq 2\) and let \(n = p_1 p_2 \ldots p_k\) be its prime factorization, where the primes are not necessarily distinct. Prove that \(k \leq \log_2 n\) (hint: prove the equivalent statement \(n \geq 2^k\) by induction on \(k\)).