1. Problem 31.2-6.
Recall $F_1 = 0$, $F_2 = 1$, and $F_{k+1} = F_k + F_{k-1}$ for $k > 2$. By the discussion in the book, $\textsc{Extended-Euclid}(F_{k+1}, F_k)$ returns (d, x, y), where $d = xF_{k+1} + yF_k$ and $d = \gcd(F_{k+1}, F_k)$. The book also shows $\gcd(F_{k+1}, F_k) = \gcd(F_k, F_{k+1} \mod F_k) = \gcd(F_k, F_{k+1} - F_k) = \gcd(F_k, F_{k-1}) = \ldots = \gcd(2, 1) = 1$. We will show by induction that for $k \geq 2$, $x = \pm F_{k-1}$ and $y = \mp F_k$. The signs are determined by this: $x > 0$ and $y < 0$ iff k is odd. Clearly $\textsc{Extended-Euclid}(F_3, F_2) = \textsc{Extended-Euclid}(1, 1) = (1, 0, 1) = (1, -F_1, F_2)$. Now let $(d, x, y) = \textsc{Extended-Euclid}(F_{k+1}, F_k)$ and $(d', x', y') = \textsc{Extended-Euclid}(F_k, F_{k-1})$. By induction assume $d' = 1$, $x' = \mp F_{k-2}$, $y' = \pm F_{k-1}$. By the algorithm, $d = d'$, $x = y'$, and $y = x' - y'$. This implies $d = 1$, $x = \pm F_{k-1}$, and $y = \mp F_{k-2} - \pm F_{k-1} = \mp(F_{k-2} + F_{k-1}) = \mp F_k$. This completes the induction since $k - 1$ is even iff k is odd.

2. Problem 31.2-8.
Note $\operatorname{lcm}(a, b) = ab/\gcd(a, b)$, so we can use Euclid's algorithm to compute the greatest common divisor to compute the least common multiple of a pair of integers. To compute the least common multiple of a set of integers, we recursively decompose it into pairs: $\operatorname{lcm}(a_1, \operatorname{lcm}(a_2, \operatorname{lcm}(\ldots \operatorname{lcm}(a_{n-1}, a_n) \ldots)))$. To be sure this works, we have to prove that $\operatorname{lcm}(a_1, a_2, \ldots, a_n) = \operatorname{lcm}(a_1, \operatorname{lcm}(a_2, \ldots, a_n))$. Let $m' = \operatorname{lcm}(a_1, a_2, \ldots, a_n)$ and $m = \operatorname{lcm}(a_1, \operatorname{lcm}(a_2, \ldots, a_n))$. Then we know $a_i | m'$ for all i, and $a_1 | m$ and $\operatorname{lcm}(a_2, \ldots, a_n) | m$. The latter implies that all of a_2 through a_n also divides m. Thus m is a common multiple of $a_1, a_2, \ldots, a_n$ and therefore greater than or equal to the least common multiple, m'. So $m \geq m'$. Conversely, we know that m' is a multiple of a_1 through a_n. Therefore it is a common multiple of a_2 through a_n, which means $\operatorname{lcm}(a_2, \ldots, a_n)$ divides m' because the least common multiple divides every common multiple. Thus m' is a common multiple of a_1 and $\operatorname{lcm}(a_2, \ldots, a_n)$, and therefore $m' \geq m$. Together we have $m = m'$, so we may break the setwise least common multiple computation into pairwise least common multiple computations. Thus there are a total of $n - 1$ multiplications and divisions plus $n - 1$ calls to Euclid's algorithm.

3. Problem 31.4-1.

$$
\begin{aligned}
&35x \equiv 10 \pmod{50}. \\
&a = 35,\ b = 10,\ n = 50. \\
&d = \gcd(35, 50) = 5.\ d = 35x' + 50y' \Rightarrow x' = 3,\ y' = -2. \\
&x_0 = x'(b/d) \bmod n = 3(10/5) \bmod 50 = 6. \\
&x_i = x_0 + i(n/d).\ i(n/d) = i(50/5) = 10i.
\end{aligned}
$$

Solutions are 6, 16, 26, 36, 46.

4. Problem 31.5-4.
By Corollary 31.29, $f(x) \equiv 0 \pmod{n}$ iff $f(x) \equiv 0 \pmod{n_i}$ for each i. Say $f(x) \equiv 0$

(mod n_i) has r_i roots. Since there are r_i possibilities for each component to be 0, there must be $\prod r_i$ possible ways for $f(x)$ to be $0 \pmod{n}$.

5. Problem 31.6-1.
   $\mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

x	$\text{ord}_{11}(x)$	$\text{ind}_{11,2}\, x$
1	1	0
2	10	1
3	5	8
4	5	2
5	5	4
6	10	9
7	10	7
8	10	3
9	5	6
10	2	5

6. Problem 31.7-1.
   $p = 11$, $q = 29$, $n = pq = 319$, $e = 3$. Note e does not divide n. $\phi(n) = 10 \cdot 28 = 280$.
   $d \equiv e^{-1} \pmod{280} \Leftarrow 3d \equiv 1 \pmod{280} \Leftarrow 3d = 1 + 280x$. $x = 3 \Rightarrow d = 187$.
   $M = 100 \Rightarrow P(M) = P(100) = 100^3 \bmod 319 = 254$.

7. Problem 31.7-3.
   $P_A(M_1)P_A(M_2) = (M_1^e \bmod n)(M_2^e \bmod n) = M_1^e M_2^e \bmod n = (M_1M_2)^e \bmod n = P_A(M_1M_2)$.

   Input: $P_A(M)$, the encrypted message; $P_A = (e, n)$, the public key
   Output: M, the decrypted message

   repeat
       Pick a random message M'.
       Encrypt M' to form $P_A(M')$.
       Calculate $P_A(MM') = P_A(M)P_A(M')$.
       Decrypt this efficiently with probability 0.01 to get $MM' \pmod{n}$.
       if efficient decryption succeeded **then**
           Compute $M'^{-1} \pmod{n}$.
           return $(MM')(M'^{-1}) = M$
       end if
   until k iterations
   return no solution found

   Note that $M'^{-1} \pmod{n}$ exists as long as M' does not divide n, but n's only factors are 1, p, q, and $n = pq$, so M' will almost always have an inverse. The probability of success of this algorithm is $1 - (0.99)^k$, so $k = 459$ implies there is a greater than 99% chance of success.