CSI2101-2009 - ASSIGNMENT#4

Due date: Thursday April 9 at 12:30 (up to max 24hs late with 10% off)
Hand in method: You may hand in to the TA at the tutorial immediately BEFORE the due date (i.e. tutorial of April 6); or otherwise at the dropoff box at SITE 1st floor.

---

(1) (25 points) Number Theory exercises:
  (a) (5 points) Section 3.4: 12 (proof involving arithmetic mod m)
  (b) (5 points) Section 3.4: 22 (proof involving arithmetic mod m)
  (c) (4 points) Section 3.5: 4-a,b,c,d (finding prime factorization)
  (d) (11 points) Section 3.7: 28 (using Fermat's Little/Chinese Remainder Thm)

(2) (20 points) Number Theory Applications:
Consider the RSA Cryptosystem. Bob's public keys are $n = 4757$ and $e = 299$. Alice uses these keys and sends Bob a message M encoded as $C = 1080$. However, since Bob used n too small, a malicious eavesdropper, Eve, is able to factor n as a product of two prime numbers: $n = 4757 = 71 \times 67$.
Show how Eve can use this information to decode the message C in order to discover the original message M; show your work and give the original message M.
Requirements:
- In order to compute the inverse of $a \pmod{m}$, when $gcd(a, m) = 1$, use the Euclidean algorithm to find the gcd and then work backwards in order to determine s and t such that $1 = gcd(a, m) = s \times a + t \times m$. The inverse of a mod m is s. Show your work.
- In order to compute $b^a \pmod{m}$ you may use some fast exponentiation algorithm available over the internet, such as the one found at: `http://www.math.umn.edu/~garrett/crypto/a01/FastPow.html`

(3) (45 points) Recurrence relations:
  (a) (7 points) Section 7.1: 30 (modeling with recurrence relations)
  (b) (5+5 points) Section 7.2: 4-b,d (solving recurrence relations)
  (c) (5+5+5+5 points) Section 7.2: 46 (modeling population growth and solving recurrence relations)
  (d) (8 points) Section 7.3: 20 (analysing divide-and-conquer algorithms)

(4) Intro to graphs (10 points)
  (a) (5 points) Section 9.2: 12, 16 (understanding graph models)
  (b) (5 points) Section 9.3: 36, 38 (graph isomorphism check)