ECE 321 HW 1 (Due Monday 9/29/08)

P8 (exercise 1.1) 5
P18 (exercise 1.2) 5
P38 (exercise 1.4) 1, 3

1.

a. $\gcd(31415, 14142) = \gcd(14142, 3131) = \gcd(3131, 1618) = \gcd(1618, 1513) = \gcd(1513, 105) = \gcd(1513, 105) = \gcd(105, 43) = \gcd(43, 19) = \gcd(19, 5) = \gcd(5, 4) = \gcd(4, 1) = \gcd(1, 0) = 1.$

b. To answer the question, we need to compare the number of divisions the algorithms make on the input given. The number of divisions made by Euclid's algorithm is 11 (see part a). The number of divisions made by the consecutive integer checking algorithm on each of its 14142 iterations is either 1 and 2; hence the total number of multiplications is between $1 \cdot 14142$ and $2 \cdot 14142$. Therefore, Euclid's algorithm will be between $1 \cdot 14142/11 \approx 1300$ and $2 \cdot 14142/11 \approx 2600$ times faster.

2.

a. Divide the given number n by 2: the remainder r_n (0 or 1) will be the next (from right to left) digit of the binary representation in question. Replace n by the quotient of the last division and repeat this operation until n becomes 0.

b. **Algorithm** *Binary*(n)
//The algorithm implements the standard method for finding
//the binary expansion of a positive decimal integer
//Input: A positive decimal integer n
//Output: The list $b_k\ b_{k-1} \ldots b_1\ b_0$ of n's binary digits
$k \leftarrow 0$
while $n \neq 0$
    $b_k \leftarrow n \bmod 2$
    $n \leftarrow \lfloor n/2 \rfloor$
    $k \leftarrow k + 1$

3.

a. Replace the ith element with the last element and decrease the array size by 1.

b. Replace the ith element with a special symbol that cannot be a value of the array's element (e.g., 0 for an array of positive numbers) to mark the ith position as empty. (This method is sometimes called the "lazy deletion".)

4.

a.

$push(a)$	$push(b)$	pop	$push(c)$	$push(d)$	pop
a	b a	a	c a	d c a	c a

b.

$enqueue(a)$	$enqueue(b)$	$dequeue$	$enqueue(c)$	$enqueue(d)$	$dequeue$
a	ab	b	bc	bcd	cd