The Extended Euclidean Algorithm

Background
Bézout's Identity: There exist integers x, y such that $ax + by = \gcd(a, b)$.
Let $a = E$ and $b = (p-1)(q-1)$. In the RSA cryptosystem, E must be relatively prime to $(p-1)(q-1)$. Thus, there exist integers x, y such that $Ex + (p-1)(q-1)y = 1$. In $\mod (p-1)(q-1)$: $Ex + (p-1)(q-1)y \equiv 1 \mod (p-1)(q-1)$, so $Ex \equiv 1 \mod (p-1)(q-1)$, so $x = E^{-1} = D$.

Theorem 6.2: If a and b are integers such that $a > b > 0$, and q and r are integers such that $a = qb + r$ where $0 \leq r < b$ (Euclidean Division), then $\gcd(a, b) = \gcd(b, r)$. i.e. $\gcd(a, b) = \gcd(a - b, b)$.
The Euclidean Algorithm applies Euclidean Division repeatedly to find the gcd by reducing the problem to smaller pieces.

The Extended Euclidean Algorithm finds the values of x and y specified in Bézout's Identity by reversing the steps of the Euclidean Algorithm.
Example: Finding $gcd(97, 20)$, and then x, y such that $97x + 20y = gcd(97, 20)$.

The Euclidean Algorithm		**The Extended Euclidean Algorithm**	
$97 = 4 \cdot 20 + 17$	(Step 1)	$1 = 3 - 1(2) = 1(3) - 1(2)$	(Rearrange step 4)
$20 = 1 \cdot 17 + 3$	(Step 2)	$= 1(3) - 1(17 - 5 \cdot 3) = -1(17) + 6(3)$	(Sub from step 3)
$17 = 5 \cdot 3 + 2$	(Step 3)	$= -1(17) + 6(20 - 1 \cdot 17) = 6(20) - 7(17)$	(Sub from step 2)
$3 = 1 \cdot 2 + 1$	(Step 4)	$= 6(20) - 7(97 - 4 \cdot 20) = -7(97) + 34(20)$	(Sub from step 1)
$2 = 2 \cdot 1 + 0$	(Step 5)		

Thus, $\gcd(97, 20) = 1$ and $97(-7) + 20(34) = 1$.

Problem Set

1. Use the Euclidean Algorithm to find the greatest common divisor for each of the following pairs of integers without using a computer.

   (a) 51, 89
   $89 = 1 \cdot 51 + 38 \to 51 = 1 \cdot 38 + 13 \to 38 = 2 \cdot 13 + 12 \to 13 = 1 \cdot 12 + 1 \to 12 = 1 \cdot 12 + 0$.
   Thus, $\gcd(51, 89) = \gcd(89, 51) = \gcd(51, 38) = \gcd(38, 13) = \gcd(13, 12) = \gcd(12, 1) = 1$.

   (b) 102, 202
   $202 = 1 \cdot 102 + 100 \to 102 = 1 \cdot 100 + 2 \to 100 = 50 \cdot 2 + 0$. Thus, $\gcd(102, 202) = 2$.

   (c) 666, 1414
   $1414 = 2 \cdot 666 + 82 \to 666 = 8 \cdot 82 + 10 \to 82 = 8 \cdot 10 + 2 \to 10 = 5 \cdot 2 + 0$. Thus, $\gcd(666, 1414) = 2$.

2. Use the Extended Euclidean Algorithm to express the greatest common divisor of each of the following pairs of integers as a linear combination of these integers.

   (a) 51, 89
   $1 = 1(13) - 1(12) = 1(13) - 1(38 - 2 \cdot 13) = -1(38) + 3(13) = -1(38) + 3(51 - 1 \cdot 38) = 3(51) - 4(38) = 3(51) - 4(89 - 1 \cdot 51) = -4(89) + 7(51)$.

   (b) 102, 202
   $2 = 1(102) - 1(100) = 1(102) - 1(202 - 1 \cdot 102) = -1(202) + 2(102)$.

   (c) 666, 1414
   $2 = 1(82) - 8(10) = 1(82) - 8(666 - 8 \cdot 82) = -8(666) + 65(82) = -8(666) + 65(1414 - 2 \cdot 666) = 65(1414) - 138(666)$.

3. Find $50^{-1} \mod 127$ without using a computer.
   By Bézout's Identity, if $\gcd(50, 127) = 1$, then $50x + 127y = 1$, so $x = 50^{-1} \mod 127$.
   First, we use the Euclidean algorithm to check $gcd(50, 127)$. $127 = 2 \cdot 50 + 27 \to 50 = 1 \cdot 27 + 23 \to 27 = 1 \cdot 23 + 4 \to 23 = 5 \cdot 4 + 3 \to 4 = 1 \cdot 3 + 1 \to 3 = 3 \cdot 1 + 0$. Thus, $gcd(50, 127) = 1$.
   Then, we use the Extended Euclidean Algorithm to solve for x and y. $1 = 1(4) - 1(3) = 1(4) - 1(23 - 5 \cdot 4) = -1(23) + 6(4) = -1(23) + 6(27 - 1 \cdot 23) = 6(27) - 7(23) = 6(27) - 7(50 - 1 \cdot 27) = -7(50) + 13(27) = -7(50) + 13(127 - 2 \cdot 50) = 13(127) - 33(50)$. Thus, $50^{-1} \mod 127 = -33$.

4. In the RSA cryptosystem, find D given $p = 13$, $q = 17$, and $E = 5$.
   $(p-1)(q-1) = 12 \cdot 16 = 192$. Now, we apply the Extended Euclidean Algorithm on $192, 5$ to find $D = 5{-}1 \mod 192$.
   The Euclidean Algorithm: $192 = 38 \cdot 5 + 2 \to 5 = 2 \cdot 2 + 1 \to 2 = 2 \cdot 1 + 0$.
   The extension: $1 = 1(5) - 2(2) = 1(5) - 2(192 - 38 \cdot 5) = -2(192) + 77(5)$. Thus, $D = 77$.

5. Write a Python function that uses the Euclidean Algorithm to find and return the greatest common divisor of two positive integers. Use your function to find the greatest common divisor for each of the following pairs of integers.

```
def euclideanGCD(a, b):
    equations = []
    if(a < b):
        a, b = b, a
    r = -1
    while(r!=0):
        q = a // b
        r = a - q * b
        equations.append({'a':a, 'q':q, 'b':b, 'r':r})
        a, b = b, r
    print("gcd(" + str(equations[0]['a']) + ", " +
        str(equations[0]['b']) + ") = " + str(equations[-1]['b']))
    return equations
```

   (a) $\gcd(9876543210, 123456789) = 9$
   (b) $\gcd(11111111111, 1000000001) = 1$
   (c) $\gcd(73433510078091009, 45666020043321) = 3$

6. Write a Python function that uses the Extended Euclidean Algorithm to write the greatest common divisor of each of the following pairs of positive integers as a linear combination of those integers.

```
def extendedAlgorithm(a, b):
    equations = euclideanGCD(a, b)
    g = equations[-1]['b']
    if len(equations) < 2:
        print(str(g) + " = " + str(equations[0]['a']) + "*0 + "
            + str(equations[0]['b']) + "*1")
        return {'gcd':g, 'a':equations[0]['a'], 'x':0, 'b':equations[0]['b'], 'y':1}
    equations=equations[-2::-1]
    a = 1
    x = equations[0]['a']
    b = -equations[0]['q']
    y = equations[0]['b']
    for i in range(1, len(equations)):
        newA = b
        newX = equations[i]['a']
        newB = a + b * -equations[i]['q']
        newY = x
        a, x, b, y = newA, newX, newB, newY
    print(str(g) + " = " + str(a) + "*" + str(x) + " + " + str(b) + "*" + str(y))
    return {'gcd':g, 'a':a, 'x':x, 'b':b, 'y':y}
```

   (a) $9 = -1371742 \cdot 9876543210 + 109739361 \cdot 123456789$
   (b) $1 = 99009901 \cdot 11111111111 + -1100110010 \cdot 1000000001$
   (c) $3 = -6495686882417 \cdot 73433510078091009 + 10445427205865236 \cdot 45666020043321$

7. In the previous section, we used the Extended Euclidean Algorithm to find that $34 \equiv 20^{-1} \mod 97$.

(a) Find $20^{-1} \mod 97$ by making clever use of Euler's Theorem.
By Euler's theorem, $20^{\phi(97)} \equiv 1 \mod 97$, so $20^{\phi(97)-1} \equiv 20^{-1} \mod 97$. 97 is prime, so $\phi(97) = 96$. Thus, $20^{-1} \equiv 20^(96-1) = 20^{95} \mod 97$. $20^{95} = (20^2)^{47} \times 20 = 400^{47} \times 20 \equiv 12^{47} \times 20 = (12^3)^{15} \times 144 \times 20 = 1728^{15} \times 2880 \equiv (-18)^{15} \times -30 = ((-18)^2)^7 \times -18 \times -30 = 324^7 \times 540 \equiv 33^7 \times 55 = (33^2)^3 \times 33 \times 55 = 1089^3 \times 1815 \equiv 22^3 \times 69 = 10648 \times 69 \equiv 75 \times 69 = 5175 \equiv 34$.

(b) Explain why using Euler's Theorem is be a much more computationally difficult strategy than using the Extended Euclidean Algorithm.
Euler's theorem first requires finding the Euler-Totient function of the modulus (let's call it m), which in turn requires calculating $m-1$ GCDs. It then involves raising the number you're trying to find the inverse of (E) to the power of m, which involves either many multiplications and modulo operations, or multiplying very large numbers together.
The Extended Euclidean Algorithm, on the other hand, only needs to calculate a GCD once (and is a calculation that would be performed in calculating $\phi(m)$ for Euler's theorem anyway, since $E < m$). The program uses a little bit more memory to drastically cut down on the number of operations it has to perform.