Contents

Chapter 1

Greatest Common Divisor

The *greatest common divisor* (GCD) of two integers, a and b, which are not both 0, is the largest integer that divides both a and b. We write this $\gcd(a, b)$ and define $\gcd(0, 0)$ to be 0 because otherwise it would be annoying.

We also have a name for two integers that share no factors. If $\gcd(a, b) = 1$, we say that a and b are *relatively prime*.

1.1 Problem Set

1. Find the greatest common divisor for each of the following pairs of integers.
   (a) 15, 35
   (b) 0, 111
   (c) -12, 18
   (d) 99, 100
   (e) 11, 121
   (f) 100, 102
2. Let a be a positive integer. What is $\gcd(a, 2a)$?
3. Let a be a positive integer. What is $\gcd(a, a^2)$?
4. Let a be a positive integer. What is $\gcd(a, a + 1)$?
5. Let a be a positive integer. What is $\gcd(a, a + 2)$?
6. Find the greatest common divisor for each of the following sets of integers.
   (a) 8, 10, 12
   (b) 6, 15, 21
   (c) -7, 28, -35

7. Find a set of three integers that are mutually relatively prime, but any two of which are not relatively prime.

8. Find four integers that are mutually relatively prime such that any three of these integers are not mutually relatively prime.

Chapter 2

Modular Numbers

Two integers, m and n, are said to be congruent modulo d if they have the same remainder when divided by d. This is written $m \equiv n \mod d$.

Consider integers mod 4. Regardless of which integer you pick, that integer must be congruent to 0, 1, 2, or 3 modulo 4 because those are the only possibilities for remainders after dividing by 4. So, for example, we say that all of the numbers in $\{\ldots, -3, 1, 5, 9, \ldots\}$ belong to the same *congruence class* mod 4. It would be nice to pick just one representative of that congruence class, and nicest of all to make that representative the smallest non-negative member of the equivalence class. So, we would say that the *canonical complete residue system* modulo 4 is $\{0, 1, 2, 3\}$ or $\mathbb{Z}_4$.

Henceforth, $\mathbb{Z}_n$ will refer to the set of integers from 0 to $n-1$, which represents all of the congruence classes modulo n.

2.1 Problem Set

1. Consider the integers modulo 6.
   (a) Construct a table for addition modulo 6.
   (b) Construct a table for subtraction modulo 6.
   (c) Construct a table for multiplication modulo 6.
2. Which decimal digits occur as the final digit of a fourth power of an integer?
3. Compute the number k in $\mathbb{Z}_{12}$ such that $37^{453} \equiv k \mod 12$. Explain how you did it.
4. Compute the number k in $\mathbb{Z}_7$ such that $2^{50} \equiv k \mod 7$ without using a computer.
5. Compute the number k in $\mathbb{Z}_{12}$ such that $39^{453} \equiv k \mod 12$ without using a computer.
6. Find the numbers in $\mathbb{Z}_{47}$ that are congruent to each of the following without using a computer.
   (a) 2^{32}
   (b) 2^{47}
   (c) 2^{200}

7. Find the canonical residue congruent to each of the following without using a computer.
   (a) $3^{10} \mod 11$
   (b) $2^{12} \mod 13$
   (c) $5^{16} \mod 17$
   (d) $3^{22} \mod 23$
   (e) Make a conjecture based on the congruences in this problem.

Chapter 3

Character Ciphers

First, let's develop a standard for translating English letters into numbers with the table below.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

If I want to encode the message "GOJAGS" using a character cipher, the first thing I need to do is find the numerical equivalents for the characters in my original plaintext. In the case of "GOJAGS", that would be 6-14-9-0-6-18.

Next, we'll encode the message using cipher. Let's use the **Caesar Cipher**[1]:

$$C \equiv P + 3 \mod 26$$

This takes my original plaintext, P, and converts it to cyphertext, C. Numerically, that produces the string 9-17-12-3-9-21. So my encoded message is:

JRMDJV

The Caesar Cipher is a type of *shift transformation*, where

$$C \equiv P + k \mod 26$$

and $k \in \mathbb{Z}_{26}$. More generally (and slightly less trivial to cryptanalyze) are *affine transformations*, where

$$C \equiv aP + b \mod 26$$

and $a, b \in \mathbb{Z}_{26}$ such that $\gcd(a, 26) = 1$. It is necessary for $\gcd(a, 26) = 1$ so that as P runs through a complete system of residues modulo 26, C also does.

There are, however, several problems with the ciphers explained above. First of all, encryption systems like this are actually pretty easy to break (that is, it is easy to infer the encryption method from just intercepting a message). In fact, using data about the way letters are used in English (see

[1] Yes, this is really a cipher attributed to Julius Caesar.

table below), one can often break a code like this by hand in few minutes (assuming the message is long enough).

The approximate frequencies of occurrence of the letters of the alphabet.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Frequency (in %)	7	1	3	4	13	3	2	3	8	< 1	< 1	4	3

Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency (in %)	8	7	3	< 1	8	6	9	3	1	1	< 1	2	< 1

Affine transformation ciphers are usually easily broken (especially when the messages are long enough) by assigning the most frequently used character to E and the second most to T. Then, you have enough information to solve for a and b, and you have broken the cipher. There are algorithms that allow laptop computers to break character ciphers in a fraction of a second.

Moreover, there is a second problem. In order for us to converse using this decoding method, you and I have to both know the code ahead of time. If I want to send you a new code, I cannot do it without, say, using the old code.

3.1 Problem Set

1. Using the Caesar Cipher, encrypt the message ATTACK AT DAWN.
2. Decrypt the ciphertext message LFDPH LVDZL FRQTX HUHG, which has been encrypted using the Caesar Cipher.
3. Encrypt the message SURRENDER IMMEDIATELY using the affine transformation $C \equiv 11P + 18 \mod 26$.
4. Decrypt the message YLFQX PCRIT, which has been encrypted using the affine transformation $C \equiv 21P + 5 \mod 26$.
5. If the most common letter in a long ciphertext, encrypted by a shift transformation $C \equiv P+k \mod 26$ is Q, then what is the most likely value of k?
6. The message IVQLM IQATQ SMIKP QTLVW VMQAJ MBBMZ BPIVG WCZWE VNZWU KPQVM AMNWZ BCVMK WWSQM was encrypted by a shift transformation $C \equiv P + k \mod 26$. Use frequencies of letters to determine the value of k. What is the plaintext message?
7. If the two most common letters in a long ciphertext, encrypted by an affine transformation $C \equiv aP + b \mod 26$, are W and B, respectively, then what are the most likely values for a and b?
8. The message WEZBF TBBNJ THNBT ADZQE TGTYR BZAJN ANOOZ ATWGN ABOVG FNWZV A was encrypted by an affine transformation $C \equiv aP+b \mod 26$. The most common letters in the plaintext are A, E, N, and S. What is the plaintext message.
9. Given two ciphers, plaintext may be encrypted by first using one cipher, and then using another cipher on the result. This procedure produces a *product cipher*. Find the product cipher

obtained by using the transformation $C \equiv 5P + 13 \mod 26$ followed by the transformation $C \equiv 17P + 3 \mod 26$.

Chapter 4

Euler's Theorem

First, we'll define a new function called the Euler phi-function or the Euler totient-function, $\phi(n)$. If n is a positive integer, then $\phi(n)$ represents the number of positive integers not exceeding n that are relatively prime to n. You should verify that $\phi(8) = 4$.

Theorem 4.1 (Euler's Theorem)**.** If m is a positive integer and a is a positive integer with $\gcd(a, m) = 1$, then

$$a^{\phi(m)} \equiv 1 \mod m.$$

4.1 Problem Set

1. Find $\phi(n)$ for the following integers.
   (a) 7
   (b) 10
   (c) 11
   (d) 16
2. Find the last digit in the decimal expansion of 3^{1000}.
3. Find the last digit in the decimal expansion of $7^{999,999}$.
4. Find the number in $\mathbb{Z}_{35}$ congruent to $3^{100,000}$.
5. Use Euler's Theorem to find the multiplicative inverse of 2 modulo 9.
6. Solve each of the following linear congruences using Euler's Theorem.
   (a) $5x \equiv 3 \mod 14$
   (b) $4x \equiv 7 \mod 15$
   (c) $3x \equiv 5 \mod 16$
7. If p and q are distinct primes, what is $\phi(pq)$? It is safe to assume that ϕ is a multiplicative function (i.e., $\phi(pq) = \phi(p) \cdot \phi(q)$) if p and q are distinct primes.

Chapter 5

The RSA Cryptosystem

5.1 The Theoretical Basis for RSA Encryption

The RSA algorithm involves 5 numbers: p, q, E, D, and M. As a brief introduction, they are:

- two different prime numbers, p and q
- two numbers in $\mathbb{Z}_{(p-1)(q-1)}$:
  - the "encoding number", E
  - the "decoding number", D
- a number in $\mathbb{Z}_{pq}$ known as the "message number", M

E can be any number in $\mathbb{Z}_{(p-1)(q-1)}$ relatively prime to $(p-1)(q-1)$. D is then defined to be the multiplicative inverse of E in $\mathbb{Z}_{(p-1)(q-1)}$.

1. At this point, you should stop and do the Google Sheets activity, "Modular Inverses". Try to answer this question: Why must E be relatively prime to $(p-1)(q-1)$?

Answer: if E has factors in common with $(p-1)(q-1)$, then it will not have a multiplicative inverse in $\mathbb{Z}_{(p-1)(q-1)}$. That is, we wouldn't be able to find the corresponding decoding number, D.

There is one key mathematical fact (which I'll refer to in this packet as the RSA Cryptosystem Crux Theorem) that provides the theoretical basis for RSA encryption. Before formally stating the RSA Cryptosystem Crux Theorem, we will consider some specific examples of its truth.

2. Let $p = 3$ and let $q = 5$. Then $(p-1)(q-1) = 8$ and $pq = 15$. Suppose that we choose E to be 3. (We could choose any number that didn't have a factor of 2 since 2 is the only factor in 8). Find D. Remember that

$$\mathbb{Z}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}.$$

3. Sticking with the same p, q, and E (and therefore the same D), complete the table below using the rules of $\mathbb{Z}_{15}$. Remember that you don't simplify exponents according to the rules of mod numbers: exponents are regular integers. What do you notice about the entries of the last row?

$M \mod pq$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M^E \mod pq$														
$M^{ED} \mod pq$														

4. At this point, you should stop and do the Google Sheets activity, "RSA Cryptosystem Crux Theorem". State a conjecture based on your observations from the activity.

The pattern that you discovered above actually holds all the time:

Theorem 5.1 (RSA Cryptosystem Crux)**.** Suppose that p and q are two distinct prime numbers. Let E be relatively prime to $(p-1)(q-1)$. Then if D is the multiplicative inverse of E in $\mathbb{Z}_{(p-1)(q-1)}$ and if M is any number in $\mathbb{Z}_{pq}$, it will always be that

$$M^{ED} \equiv M \mod pq.$$

The previous theorem, which you have seen demonstrated, is actually not so hard to prove. You'll be able to do so shortly. It relies on three results which lead up to it.

Theorem 5.2. If p and q are distinct prime numbers and M is a positive integer with $\gcd(M, pq) = 1$, then

$$M^{(p-1)(q-1)} \equiv 1 \mod pq.$$

Theorem 5.3. Let p and q be distinct prime numbers, k be a positive integer, and M be a number in $\mathbb{Z}_{pq}$ with $\gcd(M, pq) = 1$. Then

$$M^{1+k(p-1)(q-1)} \equiv M \mod pq.$$

Theorem 5.4. Let p and q be distinct primes and E be a number in $\mathbb{Z}_{(p-1)(q-1)}$ such that E is relatively prime to $(p-1)(q-1)$. Then E has a multiplicative inverse in $\mathbb{Z}_{(p-1)(q-1)}$. That is, there exists some D in $\mathbb{Z}_{(p-1)(q-1)}$ such that

$$D \equiv E^{-1} \mod (p-1)(q-1).$$

Notice that Theorem 5.4 is the formal statement of the conclusion you reached in problem #1. I won't ask you to prove this one, but hopefully you have become confident in its truth after exploring some specific examples.

5. Find the primes p and q if $pq = 14,647$ and $\phi(pq) = 14,400$.
6. Prove Theorem 5.2 based on what you have already learned (perhaps in a previous section).
7. Prove Theorem 5.3 based on what you have already learned.
8. Prove Theorem 5.1 (RSA Cryptosystem Crux) based on what you have already learned.

5.2 The RSA Encryption Algorithm

Here is the idea. Suppose that there are two people, Alice and Bob, and Alice wants to send Bob a secret message. (Believe it or not, whenever cryptographers discuss a situation like this one involving two hypothetical people, they almost always use the names Alice and Bob.) Using some simple encryption method like the one above, Alice converts her message to a number, which we call M (M might be a really long number).

Here is how Bob and Alice communicate. First, Bob picks two prime numbers p and q. We will assume $p = 3$ and $q = 97$, though in practice the primes will be much larger (more on this later). Bob also picks an encoding number, E, from $\mathbb{Z}_{(p-1)(q-1)} = \mathbb{Z}_{192}$, making sure that E has no prime factors in common with $(p-1)(q-1) = 192$. We will choose $E = 5$. Bob then **tells** Alice the number $pq = 291$ and the encoding number E. The crazy part is that it won't matter whether this message is intercepted. Bob could publish the two numbers on his website!

Here is how Alice encodes her message: she simply raises her numerical message M to the power of E and computes the answer in $\mathbb{Z}_{pq} = \mathbb{Z}_{291}$. She sends this number to Bob.

1. Suppose that Alice's secret message is the number $M = 2$. What number does she send Bob? Describe in your own words how you found this number.

2. Suppose that Alice's secret message is the number $M = 150$. What number does she send Bob? If you try to do this directly, your calculator might overflow, but there is a way to avoid this problem (and these sorts of computational shortcuts are important in practical implementations of RSA). Describe how you found the number in your own words.

Notice that we must know ahead of time that M is less than 291 (so that we can appropriately interpret it as a number in $\mathbb{Z}_{291}$) which doesn't give us too many options. In practice, however, p and q are extremely large and so this won't cause us a problem. (Alice could also compensate by sending her message in small chunks.)

Once Bob receives the encoded message, he needs to decode it. First, he computes the multiplicative inverse of $E = 5$ in $\mathbb{Z}_{(p-1)(q-1)} = \mathbb{Z}_{192}$. Notice you could easily find this using Euler's Theorem and a computer. By hand, this method is admittedly tedious when working in $\mathbb{Z}_{192}$. Therefore I will just tell you that the inverse of 5 is $D = 77$ and point out that, in practice, there is an algorithm, called the Euclidean algorithm, by which it is relatively easy to compute inverses of modular numbers (and extremely easy for a computer to make such a computation even for very large numbers). You may learn more about this later.

3. Bob can now decode Alice's message:

   (a) Verify that 77 is indeed the multiplicative inverse of 5 in $\mathbb{Z}_{192}$. Explain in your own words how you know that you are correct.

   (b) Using the RSA Cryptosystem Crux Theorem, explain how Bob can use the number D to decode Alice's encoded message M^E and recover her original message M.

5.3 Why RSA Encryption is Useful

Let's talk about the advantages of this system.

1. Suppose that you are listening in on Bob's public transmission and you hear that $pq = 21$ and $E = 5$. Recover $(p-1)(q-1)$ and D. You have broken Bob's code!

2. You are a secret agent. An evil spy with shallow number theory skills uses the RSA Public Key Coding System in which the public modulus is $n = 1537$, and the encoding exponent is $E = 47$. You intercept one of the encoded secret messages being sent to the evil spy, namely the number 570. Using your superior number theory skills, decode this message, thereby saving countless people from the fiendish plot of the evil spy.

At this point you might be thinking that this encryption method is even easier to break than character ciphers are, and, if Bob used the numbers of the previous example, you would be absolutely right. There is, however, a catch: in order to break Bob's code you had to factor pq. In practice, the primes chosen are much larger, perhaps hundreds of digits long. Factoring huge numbers is difficult even for a computer. In fact, one can chose primes that are so large that a computer effectively cannot break the code (though making your primes too large can also make the encoding and decoding processes take too long).

We have thus solved the problem of having a code that is too easy to break. We have also solved our second problem: Bob can give Alice a new code any time he wants and it doesn't matter who overhears. Alice (and anybody else overhearing) will know how to encode messages but only Bob will know how to decode them. By the way, an encryption algorithm like this is called a *public key* encryption algorithm because Bob can make his encoding key available to the public.

3. Summarize RSA encryption and the advantages it offers in your own words.

4. Can you see any weaknesses in RSA? There are a couple and one has to make a few simple modifications to the messages being sent and the algorithm to avoid them. For the sake of simplicity, the method discussed here is the simplest version and in practice more sophisticated methods (based on the same idea) are used.

When we first started studying mod numbers, a couple of you were curious as to their uses in real life. I responded that if you were able to magically understand everything about mod numbers (and if you played your cards correctly), you could probably rule the world. That is because mod numbers are used in RSA encryption (and in tons of other things) and RSA encryption is used millions of times a day. When you log onto a website, it often sends your computer a public encryption key (like the one Bob sent to Alice). Your computer can then easily send information to the website (your credit card number, Social Security number, etc.) without the risk of hackers being able to decode the messages. If you could somehow break RSA, you could get all of this information and use it to serve your own evil purposes (though my understanding is that mathematicians have proven that, properly implemented, RSA is basically unbreakable). Again, I am simplifying things a bit, but the situation I described is the basic idea.

Chapter 6

The Euclidean Algorithm

6.1 Motivation

Suppose you want to break an RSA encryption. You have two tasks ahead of you. Recall that pq and E are public.

- You first need to factor pq; i.e., you need to figure out what p and q are, given only their product.
- You need to find $E^{-1} \mod (p-1)(q-1)$ because that number is D, the decoding number.

The first task, factoring pq, is difficult. That's the whole reason the RSA cryptosystem is so effective. Mercifully, the second task, finding $E^{-1} \mod (p-1)(q-1)$, is significantly easier. Finding $E^{-1} \mod (p-1)(q-1)$ will be the focus of this chapter.

An overview of the relevant number theory

There is a theorem in number theory known as **Bézout's Identity**, which states that if a and b are non-zero integers, then there exist integers x and y such that

$$ax + by = \gcd(a, b).$$

For our purposes, we will only need a special case of this fact.

Theorem 6.1 (A special case of Bézout's Identity)**.** If a and b are non-zero integers such that $\gcd(a, b) = 1$, then there exist integers x and y such that

$$ax + by = 1.$$

The **Extended Euclidean Algorithm** is an algorithm that computes the solution to Bézout's identity. That is, given $ax + by = \gcd(a, b)$, the Extended Euclidean Algorithm will tell you what x and y have to be.

Relating this theory to the RSA cryptosystem

Remember E, our encoding number from the RSA cryptosystem? Let's let $a = E$ and $b = (p-1)(q-1)$. Bézout's Identity gurantees that there are integers x and y such that

$$Ex + (p-1)(q-1)y = \gcd(E, (p-1)(q-1)).$$

But what is $\gcd(E, (p-1)(q-1))$? If you'll recall, E and $(p-1)(q-1)$ have to be relatively prime. Therefore, $\gcd(E, (p-1)(q-1)) = 1$, and so Bézout's Identity in this case looks like

$$Ex + (p-1)(q-1)y = 1.$$

Why this matters to you

One could, and you should, write a program that executes the Extended Euclidean Algorithm to solve this equation for x and y. When you do so, you'll have an incredibly valuable piece of information. Consider this equation modulo $(p-1)(q-1)$.

$$\begin{aligned}
Ex + (p-1)(q-1)y &= 1 \\
Ex + (p-1)(q-1)y &\equiv 1 \mod (p-1)(q-1) \\
Ex &\equiv 1 \mod (p-1)(q-1) \qquad \text{since } (p-1)(q-1) \equiv 0 \mod (p-1)(q-1) \\
x &\equiv E^{-1} \mod (p-1)(q-1)
\end{aligned}$$

In other words, x is D, the decoding number!

6.2 The Euclidean Algorithm

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 $0 \leq r < b$, then

$$\gcd(a, b) = \gcd(b, r).$$

This is another way of stating the same fact: If a and b are integers such that $a > b > 0$, then $\gcd(a, b) = \gcd(a - b, b)$. I find this statement a little more difficult to relate to the steps of the Euclidean Algorithm, but it is easier to program.

The **Euclidean Algorithm** is a procedure by which the preceding theorem is repeatedly applied to find the greatest common divisor of two integers. I'll demonstrate using an example wherein I'd like to find the $\gcd(51, 15)$.

a	$=$	$q \cdot b$	$+$	r	
51	$=$	$3 \cdot 15$	$+$	6	(Step 1)
15	$=$	$2 \cdot 6$	$+$	3	(Step 2)
6	$=$	$2 \cdot 3$	$+$	0	(Step 3)

In Step 1, we used the division algorithm to write 51 as $3 \cdot 15 + 6$. Using the notation from our theorem, we have $a = 51$, $b = 15$, $q = 3$, and $r = 6$. The theorem tells us that $\gcd(51, 15)$, whatever that number might be, is the same as $\gcd(15, 6)$.

We have reduced the problem of finding $\gcd(51, 15)$ to simply finding $\gcd(15, 6)$. Here comes Step 2. Since 15 can be written as $2 \cdot 6 + 3$, our theorem—this time using $a = 15$, $b = 6$, $q = 2$, and $r = 3$—tells us that $\gcd(15, 6) = \gcd(6, 3)$.

We now know that $\gcd(51, 15) = \gcd(15, 6) = \gcd(6, 3)$. In Step 3, we find that $\gcd(6, 3) = \gcd(3, 0)$ (this time by using $a = 6$, $b = 3$, $q = 2$, and $r = 0$). When we reach a remainder of zero, the Euclidean Algorithm halts. This is because for any positive integer n, $\gcd(n, 0) = n$. So the b-value from our final step (or, equivalently, the r-value from our second-to-last step) is our greatest common divisor. In this case, the Euclidean Algorithm concluded (correctly) that $\gcd(51, 15) = 3$.

6.3 The Extended Euclidean Algorithm

Let's look at the integers 97 and 20. First, we'll use the Euclidean algorithm to determine $\gcd(97, 20)$. Then, we'll demonstrate the implementation of the Extended Euclidean Algorithm on the same two numbers.

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

The Euclidean Algorithm tells us that $\gcd(97, 20) = 1$. Recall that Bézout's Identity guarantees that there exist integers x and y such that

$$97x + 20y = 1.$$

The Extended Euclidean Algorithm finds those values of x and y for us simply by reversing the steps of the Euclidean Algorithm! Since, we now know that $97(-7) + 20(34) = 1$, we also know that $-7 \equiv 97^{-1} \mod 20$ and $34 \equiv 20^{-1} \mod 97$.

6.4 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
   (b) 102, 202
   (c) 666, 1414
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
   (b) 102, 202

(c) 666, 1414

3. Find $50^{-1} \mod 127$ without using a computer.

4. In the RSA cryptosystem, find D given $p = 13$, $q = 17$, and $E = 5$.

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.

   (a) 9876543210, 123456789

   (b) 11111111111, 1000000001

   (c) 45666020043321, 73433510078091009

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.

   (a) 9876543210, 123456789

   (b) 11111111111, 1000000001

   (c) 45666020043321, 73433510078091009

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.

   (b) Explain why using Euler's Theorem is be a much more computationally difficult strategy than using the Extended Euclidean Algorithm.