Problems from the Book:

Chapter 2

2. Changing the plaintext to numbers yields 7, 14, 22, 0, 17, 4, 24, 14, 20. Applying \(5x + 7\) to each yields \(5 \cdot 7 + 7 = 42 \equiv 16 \pmod{26}\), \(5 \cdot 14 + 7 = 77 \equiv 25\), etc. Changing back to letters yields ZNIHOBXZD as the ciphertext.

4. Let the decryption function be \(x = ay + b\). The first letters tell us that \(7 \equiv a \cdot 2 + b \pmod{26}\). The second letters tell us that \(0 \equiv a \cdot 17 + b\). Subtracting yields \(7 \equiv a \cdot (-15) \equiv 11a\). Since \(11^{-1} \equiv 19 \pmod{26}\), we have \(a \equiv 19 \cdot 7 \equiv 3 \pmod{26}\). The first congruence now tells us that \(7 \equiv 3 \cdot 2 + b\), so \(b = 1\). The decryption function is therefore \(x \equiv 3y + 1\). Applying this to CRWWZ yields happy for the plaintext.

5. Let \(mx + n\) be one affine function and \(ax + b\) be another. Applying the first then the second yields the function \(a(mx + n) + b = (am)x + (an + b)\), which is an affine function. Therefore, successively encrypting with two affine functions is the same as encrypting with a single affine function. There is therefore no advantage of doing double encryption in this case. (Technical point: Since \(\gcd(a, 26) = 1\) and \(\gcd(m, 26) = 1\), it follows that \(\gcd(am, 26) = 1\), so the affine function we obtained is still of the required form.)

15. The number of seconds in 120 years is

\[
60 \times 60 \times 24 \times 365 \times 120 \approx 3.8 \times 10^9.
\]

Therefore you need to count \(10^{100}/(3.8 \times 10^9) \approx 2.6 \times 10^0\) numbers per second!

Chapter 2 Computer Problems

2. Use \(\text{fr} = \text{frequency(lcll)}\) to get a frequency count. Observe that the most common common letter is \(l\), which is 7 places after \(e\). Try shifting back by 7 using \(\text{shift(lcll}, -7)\) to get the answer \(\text{ans = eveexpectseggsforbreakfast}.\)

6. a) The message can be found in the file ciphertexts.m under the variable gaat. The following code performs the conversion to 0, 1, 2, 3, and performs the shifting.

```matlab
ind0=find(gaat==A);
ind1=find(gaat==C);
ind2=find(gaat==G);
ind3=find(gaat==T);
vec=gaat; vec(ind0)=0; vec(ind1)=1; vec(ind2)=2; vec(ind3)=3;
vec=mod(vec+1,4);
ind0=find(vec==0);
ind1=find(vec==1);
ind2=find(vec==2);
ind3=find(vec==3);
output=vec;
output(ind0)=A;
output(ind1)=C;
output(ind2)=G;
output(ind3)=T;
output=char(output);
The answer is TCCAAGTGTGTGGCAACCGGACGACCCCTTCAGAGACTCCGA.
```

b) The following code assumes that the affine cipher is of the form \(y = ax + b \pmod{4}\). The parameters \(a\) and \(b\) should be entered in. The restrictions are that \(a\) is relatively prime to 4 which means that \(a\) is either 1 or 3.

```matlab
ind0=find(gaat==A); ind1=find(gaat==C); ind2=find(gaat==G); ind3=find(gaat==T);
vec=gaat;
vec(ind0)=0; vec(ind1)=1; vec(ind2)=2; vec(ind3)=3;
```
vec=mod(a*vec+b,4);
ind0=find(vec==0);
ind1=find(vec==1);
ind2=find(vec==2);
ind3=find(vec==3);
output=vec;
output(ind0)=A;
output(ind1)=C;
output(ind2)=G;
output(ind3)=T;
output=char(output);

Chapter 3

1. (a) Apply the Euclidean algorithm to 17 and 101:

\[ 101 = 5 \cdot 17 + 16 \]

\[ 17 = 1 \cdot 16 + 1. \]

Working back yields \( 1 = 17 - 16 = 17 - (101 - 5 \cdot 17) = (-1) \cdot 101 + 6 \cdot 17. \)

(b) Since \(-101 + 6 \cdot 17 = 1\), we have \(6 \cdot 17 \equiv 1 \pmod{101}\). Therefore \(17^{-1} \equiv 6 \pmod{101}\).

Supplemental

1. Notice that if \(n = 7 \cdot 11\) then the numbers \(a\) that are relatively prime to \(n\) are precisely those that do not contain either 7 or 11 as their factors. The numbers that have either 7 or 11 as factors are those that are multiples of 7 or multiples of 11. Thus, the numbers \(a\) with \(\gcd(a, n) = 1\) are

\[ 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 15, \ldots. \]

If we generalize to \(n = pq\) where \(p\) and \(q\) are primes, then we must remove all multiples of \(p\) or \(q\). Hence, we remove

\[ p, 2p, 3p, \ldots \]

and

\[ q, 2q, 3q, \ldots \]