CS 105 Review Questions #2

In addition to these questions, remember to look over your labs, homework assignments, readings in the book and your notes.

1. Explain how writing functions inside a Pascal program can potentially make the program more concise (i.e. take up fewer total lines of code).

2. Indicate whether each statement is true or false.
   a. In ASCII code, the values corresponding to letters in the alphabet are consecutive.

   b. ASCII code uses the same values for lowercase letters as it does for capital letters.

   c. In ASCII code, non-printing, invisible characters all have value 0

3. Properties of ASCII representation:
   a. Each character is stored as how much information?

   b. If the ASCII code for 'A' is 65, then the ASCII code for 'M' should be _____.

   c. Suppose that the hexadecimal ASCII code for the lowercase letter 'a' is 0x61. Which lowercase letter would have the hexadecimal ASCII code 0x71 ?

   d. If we represent the sentence "The cat slept." in ASCII code (not including the quotation marks), how many bytes does this require?

   e. How does Unicode differ from ASCII?

4. Estimate the size of the following file. It contains a 500-page book, where a typical page has 2000 characters of ASCII text and one indexed-color image measuring 200x300 pixels. "Indexed color" means that each pixel of an image takes up one byte.

5. Suppose that c is a variable that contains a character value. We wish to determine if it is a letter, and we know that capital letters have lower ASCII values than lowercase letters.
   a. What is wrong with this if-statement?
      if (c >= 'A') and (c <= 'z') then

   b. How would you fix it?

6. The ASCII codes for 'a' and '0' (zero) are 97 and 48, respectively. What is the ASCII code for '3' ? What is the ASCII code for '6' ? The value '3' + '3' is the ASCII code for what letter? When you ask the computer for 3+3, how come it knows to give you 6 as the answer instead of a letter?

7. The process of converting cyphertext into plaintext is called _____.

8. If we encrypt the word YOU using a Caesar cipher with the key = 3, what is the encrypted form?

9. Suppose someone sent a message in English that was encrypted with a Caesar cipher. One of the words in the ciphertext message is LEWLYPLUJL.
   a. How would you begin to guess the original plaintext word if you don't know the key?

   b. Explain why it is not necessary to decode each letter when testing a possible key.

   c. What was the original word?

   d. What was the key value?

10. Besides trial and error on the various key values, what is another way to attack a Caesar cipher? Under what circumstances would you use one method over the other?

11. Suppose a Caesar-like encryption system uses this cipher function for each letter to encrypt: $f(x) = (7x + 3) \bmod 26$. (This type of cipher is called an affine cipher.) What is the encrypted value of the letter K? Assume that the letters A-Z are numbered 1-26 for this purpose.

12. Encode the message "steganography" using a transposition cipher with key = 3.

13. In steganography, a message is inserted into a larger text file.
   a. Explain how to accomplish this insertion without changing the text file's size.

   b. Explain how to extract the message.

   c. How much larger should the text file be compared to the message? Why?

   d. How was steganography accomplished in ancient times?

14. How large of a text message can be hidden inside a 200x200 RGB image? Assume the image depicts something busy with much detail such as a tropical rain forest.

15. In cryptography, what is the difference between a cipher and a code?

16. In order to convert plaintext into ciphertext, you need two pieces of information. What are they?

17. Besides determining a secret message's key, what else can a cryptanalyst do to glean the meaning of a message?

18. Which types of ciphers are sensitive to the statistical distribution of letters?

19. The two classic types of cryptography are transposition and substitution. Explain what each one means.

20. What features did the Enigma machine have that made it more than just a Vigenère cipher?

21. Which encryption scheme is harder to crack: a cryptogram, or a one-time pad? Why?

22. In the simple substitution cipher, how many possible keys are there? Why is this not a secure method of encryption, despite the number of keys?

23. When using a homophonic cipher, why is each letter replaced with a two-digit number instead of a one-digit number or letter?

24. Give two examples of a polyalphabetic cipher system.

25. Use the book cipher to encrypt this phrase: "we did the test." Use the following text as the key.

Jakaranda strate roep my vanaand.
Ek luister na die liedjie op die wind,
En ek volg my hart waar hy gaan.
Jare gelede vaar ek na die vreemde
Met drome wat sweef op die wind.
My hart se begeerte om lewensgeheime
In onbekende stede te vind.

26. What was the weakness of the Vigenère cipher that Charles Babbage was able to exploit?

27. Why is the one-time pad considered the best form of a Vigenère cipher?

28. When the Germans used Enigma, why did they have both a day key and a message key?

29. What did Enigma's plugboard do?

30. A certain cipher system will take 50 years to break using current technology. Moore's Law states that the speed of the latest computer technology doubles every 18 months. Assuming Moore's Law is still valid into the future, how long can we expect the cipher system to still be secure?

31. Define the following terms, as they relate to the subject of secret communication:
    a. steganography

    b. transposition

    c. null

32. Briefly explain how the Great Cipher of Louis XIV worked. What did the ciphertext look like?

33. Suppose you received a message that was encrypted using a book cipher. You have the book that was used as the key. The first number in your ciphertext message is 3. How do you determine the plaintext word that is represented by this number? Be specific.

34. Suppose the country is at war, and you manage to intercept an enemy message. You know that the enemy always uses a one-time pad to encrypt its message. Your cryptanalysis team tells you that they are almost done deciphering the text. So far, they have deduced that the plaintext message should read as follows.

    CEASE ZQRE SURRENDER TO US FORCES

    What advice would you give to the cryptanalysis team as they continue their work?

35. For each of the following cipher systems, encrypt the word "australia".
    a. Vigenère cipher with key = "surf"

    b. Caesar cipher with key = 24

36. The purpose of a homophonic cipher is to encrypt common letters with one of several possible numbers. Rare letters would still be represented by a single value. On average, the letters J, Q, and Z appear only 0.1% of the time in English text. With this in mind, explain how using a 2-digit number in a homophonic cipher would make these unusual plaintext letters conspicuous in ciphertext.

37. What two properties should a Vigenère cipher key have in order to maximize its security?

38. What does the phrase "exhaustive key search" mean? What is its purpose?

39. Explain why a homophonic cipher is more difficult to break than an ordinary substitution cipher.

40. Compare the Vigenère cipher with the Caesar cipher.

41. Compare a one-time pad with a Vigenère cipher.

42. Is the pinprick method an example of cryptography or steganography? How does it work?

43. What is wrong with the following Pascal code that attempts to find the smallest number in array A? Show how you would fix it.

```
min := 0;
for i := 1 to length(A) do
  if A[i] < min then
    A[i] := min;
```

44. What is the output as the following Pascal code executes? Assume that A is an array that contains 3 integers: 4, 7 and 2.

```
for i := 1 to 3 do
  begin
    write(A[i]);
    write('A');
  end;
writeln('B');
```

45. Suppose array A contains the 5 integer values 2, 3, 0, 4 and 1. What is the value in the variable "location" when this code has completed?

```
target := 3;
for i := 1 to 5 do
  if target = A[i] then
    location := i;
```

46. The following algorithm attempts to solve the following problem. We have a list of 10 playing cards, and we want to count how many are face cards (jack/queen/king), and print out this number. But there is one mistake. What is it? Assume that the cards have already been initialized into the array "list".

(1) facecard := 0
(2) for count := 1 to 10
(3) if list[count] is jack, queen or king
(4) facecard := facecard + 1
(5) count := count + 1
(6) Print the value of count.

a. On line 1, facecard should be set to 1 instead of 0.
b. On line 2, the 10 should be changed to 11.
c. On line 2, the 1 should be changed to 0.
d. Lines 4 and 5 should be reversed.
e. On line 6, "count" should be replaced with "facecard".
f. On line 6, the statement should appear inside the loop, not after it.

47. The following code attempts to determine if a list contains an odd number. However, it contains a mistake. Show how you would fix the error. Assume that array A contains these values: 3, 4, 7, 9, 12, 8, 5, 6.

```
for i := 1 to length(A) do
  if A[i] mod 2 = 1 then
    found := true
  else
    found := false
```

Answers to review questions for test 2.

1. If there is a block of code that needs to be performed more than once in different parts of a program, it would probably be beneficial to use a function. The function would be written at the beginning of the program. Then, every time its actions need to be performed, you could simply call the function instead of retyping all of the steps.

   As a food analogy, think about a cookbook for making cookies. In every recipe it is necessary to start with cookie dough. This procedure requires several steps. But it would be wasteful to include the details on how to make cookie dough for every cookie recipe in the book. Instead, at the beginning of the book there would be a special recipe (function) just for making cookie dough. Then, all the other recipes can refer to this initial step (calling the function).

2. True/false:
   a. True
   b. False
   c. False

3. Properties of ASCII representation:
   a. Each character has 8 bits or 1 byte.
   b. M is the 13^{th} letter of the alphabet. Thus, M = A+12, so M = 65+12 = 77.
   c. 0x71 – 0x61 = 16. Sixteen letters after ‘a’ is the 17^{th} letter which is ‘q’.
   d. The words have lengths of 3, 3, and 5. There are two spaces and one punctuation symbol. Each symbol is one byte, for a total of 3+3+5+2+1 = 14 bytes.
   e. In Unicode, characters are allocated 2 bytes instead of 1, allowing for many more possible symbols, such as many foreign alphabets.

4. Each page will have 2000 characters, and an image containing 60,000 pixels. In this case, each character and pixel is one byte. Thus, each page has 62,000 bytes of information, which is roughly 60 KB. If there are 500 pages, the total document is about 30 MB.

5. Checking to see if a character is a letter:
   a. The problem is that there are nonletter characters between the capital and lowercase values. In other words, there are several values between ‘Z’ and ‘a’.

b. We could be more explicit in the if-condition, as follows:
if (('A' <= c) and (c <= 'Z')) or (('a' <= c) and (c <= 'z')) then

6. The ASCII code for '3' is 3 more than the ASCII code for '0'. This means '3' = '0' + 3 = 48 + 3 = 51.
Similarly, the ASCII code for '6' is '0' + 6 = 48 + 6 = 54.
'3' + '3' works out to 102, which is 5 more than 97, so 102 is 'f'.
The computer can distinguish between 3 + 3 and '3' + '3' because it understands the difference between the integer (number) type and the string (text) type.

7. Decryption or decrypting

8. Let's assume that the alphabet wraps around. In other words Z+1 = A. In this case, adding 3 to the letters of YOU gives us (Y + 3)(O + 3)(U + 3) = BRX.

9. More practice with the Caesar cipher:
   a. We would need to decrypt the message with every possible key from 1 to 25. One of the results is the correct plaintext.
   b. For each of the 25 cases, it's not necessary to decode the entire message, because after looking at the first few letters, we can immediately tell if the letters are not forming an intelligible word, and we can give up on that case.
   c. Now the question asks for us to crack the message!
   Let's try subtracting 1 from each letter: KDV… – we conclude no word can start this way.
   Let's try subtracting 2 from each letter: JCU... – we conclude no word can start this way.
   Let's try subtracting 3 from each letter: IBTIVM… – we conclude no word can start this way.
   Let's try subtracting 4 from each letter: HASHUKH… - we conclude no word can start this way.
   Let's try subtracting 5 from each letter: GZR… - we conclude no word can start this way.
   Let's try subtracting 6 from each letter: FYQF… - we conclude no word can start this way.
   Let's try subtracting 7 from each letter: EXPERIENCE – looks like a good answer!
   d. Evidently the key is 7.

10. We could do a frequency analysis, as we would for a general substitution cipher (cryptogram). Or we could look for the existence of very short words. These techniques are useful when the amount of text we have to work with is large. For short messages, such as the previous question, we lack these context cues, but trial and error is still reliable.

11. K is letter number 11. f(11) = (7*11 + 3) mod 26 = 80 mod 26 = 2. The enciphered letter is B.

12. We insert the letters of our word into a table having only 3 columns. Unused cells are set to Z. To obtain the ciphertext, we read down the columns.

S	T	E
G	A	N
O	G	R
A	P	H
Y	Z	Z

Therefore, the ciphertext is SGOAY TAGPZ ENRHZ. I've put in spaces for readability. In reality, there would not be spaces, or you could group the letters differently (e.g. in fours).

13. Steganography questions.
    a. We insert characters by replacing existing letters.
    b. Create a string variable for the recovered plaintext. Initialize it to be the empty string. Find the places where the characters were inserted (e.g. every 100th letter). For each letter you encounter, concatenate it into the recovered plaintext string.
    c. The container message should be much larger so that the inserted message cannot be detected. In our lab, we experimented with container messages being 100 or 1000 times larger, at least.
    d. In ancient times, a message could be hidden inside food, on the scalp of someone's head, or by subtly marking letters of an existing message. Many other low-tech examples are possible.

14. Let's suppose we try to put 1 bit of our message inside each pixel of an image. There are 40,000 pixels in the image. Therefore, we can hide 40,000 bits, which equals about 5,000 bytes or 5,000 characters. This is a little more than one typed page.

15. A code performs encryption at the level of entire words instead of individual letters as a cipher does.

16. You need to know the method of encipherment and the key.

17. Look for context cues, such as the time of day of the message, where the message is from/to, the length of the message, detect if the message was made leisurely or in haste.

18. Frequency analysis can be exploited for substitution ciphers.

19. Transposition means moving the letters around, but not changing them. Substitution means shuffling the alphabet so that each letter is replaced by some other letter, as in a cryptogram.

20. Enigma had a plugboard that swapped pairs of letters.

21. A one-time-pad is much more difficult. In fact, if properly implemented it is unbreakable. The key is long and random. When trying all possible keys to crack the message, all possible plaintext messages will come out.

22. The number of keys is 26! (twenty-six factorial) All a cryptanalyst has to do is apply frequency analysis and look for common short words and digraphs.

23. With a homophonic cipher, common letters are replaced with one of several possible symbols. The number of possible ciphertext symbols is therefore much larger than 26, so a 2-digit number is certainly needed at the very least. A one-digit number would be unsuitable because that would only allow for 10 distinct symbols. Using letters is no good because we need far more than 26 symbols. The example in the book used 100 symbols, from 00 to 99.

24. A simple polyalphabetic cipher would be to use two Caesar ciphers at once. For example, adding 5 to the odd numbered letters, and adding 7 to the even numbered letters. We could get even more fancy and use 3 Caesar ciphers, by adding 5, 7 and 18 to each letter, and repeating the sequence. In general, we can use a Vigenere cipher, which is the most famous kind of polyalphabetic cipher, and is the basis for one-time pad encryption and Enigma.

25. We need to number each of the words of the book. Here it goes!

    (1)Jakaranda (2)strate (3)roep (4)my (5)vanaand
    (6)Ek (7)luister (8)na (9)die (10)liedjie (11)op (12)die (13)wind
    (14)En (15)ek (16)volg (17)my (18)hart (19)waar (20)hy (21)gaan
    (22)Jare (23)gelede (24)vaar (25)ek (26)na (27)die (28)vreemde
    (29)Met (30)drome (31)wat (32)sweef (33)op (34)die (35)wind
    (36)My (37)hart (38)se (39)begeerte (40)om (41)lewensgeheime
    (42)In (43)onbekende (44)stede (45)te (46)vind

    Each letter in the plaintext needs to be replaced with a number based on the numbering of these words. A letter may correspond to more than one possible number, so there is more than one possible ciphertext. Here is a possible answer: 13-6 9-42-9 45-18-6 45-6-2-45

26. Babbage looked for repetitions of short letter sequences. For example, seeing many instances of JQX could be a common three-letter word like THE. The locations of these sequences implies the length of the key. The individual letters of the key can be inferred by doing frequency analysis on each interleaved portion of letters in the message, knowing that each distribution is a shifted Caesar cipher.

27. A one-time pad ensures the two things we need to perfect a Vigenere cipher: we need the key to be long and random. It consists of a random stream of numbers. It is a "pad" in the sense that we try not to use the same key for two different message. If it's a real pad, we are supposed to tear off one page after use and not use it again.

28. The day key was the key to be used on a particular day. However, it was only used to encode the message key. Each message would have a unique message key. The Germans did not want to use the day key to encode entire messages because it would give an eavesdropper too much text (all the messages of the day) using the same key.

29. The purpose of the plugboard was to swap letters before transmission.

30. Let's see how long it will take to complete the decipherment if we decide to wait until a better computer comes along.

If we wait this long to start	The computation time will take	Total time
0 years	50 years	50 years
1.5 years	25 years	26.5 years
3 years	12.5 years	15.5 years
4.5 years	6.2 years	10.7 years
6 years	3.1 years	9.1 years
7.5 years	1.6 years	9.1 years
9 years	0.8 years	9.8 years

    The shortest amount of time is approximately 9 years.

31. Definitions
    a. Hiding a message inside a larger document.
    b. A cipher system that moves letters around instead of changing them.
    c. A ciphertext symbol that stands for nothing in the plaintext. (Literally nothing at all)

32. The Great Cipher encoded syllables instead of letters. Each ciphertext symbol was a 3-digit number.

33. The ciphertext number 3 refers to the first letter of the 3rd word in the book. Note that each number in the ciphertext represents just one letter of plaintext. (However, if the book is so long (like a real book) that every conceivable word is contained in it, we could revise our definition of book cipher to allow for each number to stand for the entire word, instead of just the first letter of the word in the book.)

34. I would tell the team to give up, because they are only going to generate all possible messages.

35. Two ways to encipher "Australia":
    a. A U S T R A L I A
       S U R F S U R F S
       As numbers, we have
        1 21 19 20 18 1 12 9 1
       19 21 18 6 19 21 18 6 19
       ---
       20 16 11 26 11 22 4 15 20
       TPKZKVDOT

    b. Adding 24 to each letter (equivalently, subtracting 2):
       YSQRPYJGY

36. We can still look for common letter combinations such as digraphs. For example, TH or ER are fairly common in English. We can also look for letters that are only followed by a small number of choices. For example, if we notice that the number 16 is always followed by either the number 45 or 57, maybe 16 is the Q and 45 and 57 are the U.

37. It should be long and random.

38. It means to try all possible keys to decipher a message. Another term for this is brute force.

39. When using a substitution cipher, the most common letters are still conspicuous. In a homophonic cipher, each symbol is used approximately with the same frequency, so frequency analysis is practically negated.

40. A Caesar cipher adds the same number to each letter of a message. A Vigenere cipher will add different values to each letter, and so is more secure.

41. A one-time pad is a type of Vigenere cipher in which the key is at least as long as the message, and is completely random. In general, a Vigenere cipher could allow for a key that is short and easy to remember, like someone's name.

42. Pinprick is a type of steganography. We are given an existing document, such as a newspaper. We use a needle to prick little holes near certain letters in the articles of the newspaper. The concatenation of these marked letters is our hidden message. We might even prick holes inside pictures and margins to make the pattern of holes appear more random.

43. There are two mistakes. The last assignment statement is written backwards. And the initial assignment to min assumes that the smallest number in the array is negative. The correct code should be as follows. Note that you could begin the loop at i = 2 instead of 1.

```
min := A[1];
for i := 1 to length(A) do
  if A[i] < min then
    min := A[i];
```

44. 4A7A2AB

45. 2

46. E

47. First, we should initialize found to false before the loop. We should not use a for-loop, because we don't necessarily need to visit all of the elements, and we don't know the number of iterations in advance. Instead, use a while loop. The loop should continue as long as there are still numbers in the array to examine, and we have not yet found any odd number.

```
found := false;
i := 1;
while (not found) and (i <= length(A)) do
  begin
    if A[i] mod 2 = 1 then
      found := true;
    i := i + 1;
  end;
```