CHAPTER 1

1) 10 - a:
   If the Germans knew their enigma was broken, it might not do well to stop all communication with the machines. Perhaps they could purposely obstruct the allies with faulty info using the enigma. Or another approach could be they go with the scorch earth method and cease all communications with the machine since the main point of the machine was defunct.

   b:
   The Nazis may have continued to use the Enigma because they did not know it had been cracked. The allies were very secretive about the whole project and great care was taken into not letting the Germans know that they had broken the enigma. Also, the Germans may have known it that it was broken but perhaps not to what extent. They maybe thought that changing a few pads or code books was sufficient. Another reason why they would continue to use it is they didn't have another cryptography solution for their communications. I think it would be worse to have no replacement.

2) a:
   Depending on what your organization is and what it's doing, then certain aspects might be more important. If for example you were a part of a top-secret project where the information is very important, then confidentiality would be the most important thing, and something like "availability" would take a backseat. However, on the flip side if you are doing something that is supposed to save life's like emergency broadcasts or other things like that, it would be best to take availability a little bit more seriously. Integrity would be most important whenever you're dealing with transactions like in banking software.
   b:
   Some situations where strengthening one thing might weaken another could be if you make your users have very complex passwords, you're strengthening the integrity of the system but weakening the availability or vice versa. Another situation could be a system where no one except a select few knows how a program or system works, something like an online banking website. The confidentiality of it would be very strong since not a lot of people know how the system works but could you as a user trust the site to keep your banking information secure just based on the banks word? I think this lowers the integrity since you must blindly trust the system for it to work, open source software/systems could be an answer to that problem.

CHAPTER 2

3) 8 - a:
   The terms confusion in cryptography means basically the amount of "obscuring" happening in between the plaintext and the cipher text basically making it harder and more complex to see a relationship between the two. While the term diffusion means "spreading" the stats of the plaintexts "through" the cipher which means for example if you change one character in the plaintext, then good diffusion will have multiple characters in the cipher text be effected.
   b:

The classic cipher which employs only confusion is a simple substitution cipher and a one-time pad.

c:

The classic cipher which employs only diffusion is a double transposition.

d:

The cipher which employs both is DES

4) 19 -

a.

The key for thrill is

	T	H	R	I	L	L
plain	111	001	101	010	100	100
	L	**K**	**I**	**S**	**T**	**L**
key	**100**	**011**	**010**	**110**	**111**	**100**
cipher	011	010	111	100	011	000
	K	I	T	L	K	E

b.

The key for tiller is

	T	I	L	L	E	R
plain	111	010	100	100	000	101
	L	**E**	**K**	**E**	**K**	**R**
key	**100**	**000**	**011**	**000**	**011**	**101**
cipher	011	010	111	100	011	000
	K	I	T	L	K	E

29 –

5) a:

On average, it'll take (2^40)/2 or 2^39 tries.

b:

Since she is doing an exhaustive search method, she's testing every combination of the keys. So, if the plaintext is completely correct with no errors, then she's got it. Much harder to do this than simple substitution. But with modern hardware, it shouldn't take her that long. To automate this I would compare the permutated cipher text to a dictionary, and if she has a certain amount of words, ping her and let her look. If the plaintext is correct, then you're done. If not, then keep going. Since this isn't simple substitution, she can't keep the key and simply substitute it until she gets closer, she must throw it out each time it doesn't give results since this is an exhaustive search attack.

c:

the amount of work required to do this with my automated way may be very large depending on the size of the dictionary. You must compare every permutation with every word in the dictionary.

d: I believe there will be a low but not insignificant amount of false alarms since you're either completely correct, or completely wrong in an exhaustive search attack, however there may be a few depending on the amount of words you want to match before an alarm is raised, also some keys might give plaintext which could be just a bunch of the letter a in a row. This would be a correct word in the dictionary but is still not correct.

6) a:

The keyspace is 62 to the 10^{th} power

b: 10ln(62)/ln(2) = 59.54 so about 60 bits 0r 2^60

c: 2^(60-1) / 2^40 = 524288

So, it would take on average about 524288 seconds or 6 days