1. Given the security levels TOPSECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED and categories A, B, and C, specify what type of access (read, write, both, or neither) is allowed in each of the following situations:

   Reading with categories has the following logic: Subject s can read object o iff L(s) dom L(o) and s has permission to read o. So 'Reading Down' is allowed

   Writing with categories has a similiar logic: Subject s can write object o iff L(o) dom L(s) and s has permission to write o. So 'Writing up' is allowed

   (a) Paul cleared for TOPSECRET, [A,C] and a document classified SECRET, [B,C]

   Neither since [A,C] is not a subset of [B,C] and TS > S.

   (b) Anna, cleared for CONFIDENTIAL,[C] and a document classified CONFIDENTIAL,[B]

   Neither since [C] $\not\subset [B] and [B] \not\subset [C]$

   (c) Jessie cleared for SECRET,[C] and a document classified CONFIDENTIAL,[C]

   Read only since Secret > Confidential and $[C] \subset$**[C]**

   (d) Sammie cleared for TOPSECRET,[A,C] and a document classified CONFIDENTIAL,[A]

   Read only since Top Secret > Confidential and $[A] \subset$**[A,C]**

   (e) Robin UNCLASSIFIED and a document classified CONFIDENTIAL,[B]

   Neither since the empty set [ ] can't be a subset of $[B]$

From the Book

2.1 Suppose someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and then send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check, and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?

The problem with this is that an eavesdropper will see K xor R being sent. Then the eavesdropper sees R sent back. Then the eavesdropper could do (K xor R) xor R which will equal the key K since the R's cancel each other out.

2.2.a This problem uses a real-world example of a symmetric cipher, from a old U.S. Special Forces manual. The document, filename *Special Forces.pdf*, is available at box.com/CompSec4e. Using the two keys (memory words) *cryptographic* and *network security*, encrypt the following message:

"Be at the third pillar from the left outside the lyceum theatre tonight at seven. If you are distrustful bring two friends."

Make reasonable assumptions about how to treat redundant letters and excess letters in the memory words and how to treat spaces and puncuation. Indicate what your assumptions are.

Assumptions are that redundant letters and spaces / puncations are going to be removed.
Doing so, our first key becomes *cryptogahi*. And we can write the message

in the following matrix format:

b	e	a	t	t	h	e	t	h	i
r	d	p	i	l	l	a	r	f	r
o	m	t	h	e	l	e	f	t	o
u	t	s	i	d	e	t	h	e	l
y	c	e	u	m	t	h	e	a	t
r	e	t	o	n	i	g	h	t	a
t	s	e	v	e	n	i	f	y	o
u	a	r	e	d	i	s	t	r	u
s	t	f	u	l	b	r	i	n	g
t	w	o	f	r	i	e	n	d	s

The first key has alphabectical order "a, c, g, h, i, o, p ,r, t, y" and each will have a value assigned according to this order from 1 - 10. Since "a" has original position of 8, the 8th column will go first. Then c has position 1 so then column 1 will go next. G has position 7, so 7th goes next and so on: h → 9th column, i → 10th column, o → 6th column, p → 4th column, r → 2nd column, t → 5th column, y → 3rd column.
The encyption that yields is the following:
trfhehftinbrouyrtusteaethgisrehfteatyrndiroltaougshlletinibitihiuoveufedmtcesatwtledmnedlraptseterfo

Doing the same thing for the second key gives: *networkscu*

And the matrix becomes:

t	r	f	h	e	h	f	t	i	n
b	r	o	u	y	r	t	u	s	t
e	a	e	t	h	g	i	s	r	e
h	f	t	e	a	t	y	r	n	d
i	r	o	l	t	a	o	u	g	s
h	l	l	e	t	i	n	i	b	i
t	i	h	i	u	o	v	e	u	f
e	d	m	t	c	e	s	a	t	w
t	l	e	d	m	n	e	d	l	r
a	p	t	s	e	t	e	r	f	o

The alphabectical order of the second key is: "c,e,k,n,o,r,s,t,u,w"
The columns read according to the values associated with the alphabectical order of the key gives: c → 9th column, e → 2nd column, k → 7th column, n → 1st column, o → 5th column, r → 6th column, s → 8th column, t → 3rd, u → 10th, w → 4th column.
The final encrypted message is again the columns read in order which gives:
isrngbutlfrrafrlidlpftiyonvseetbehihtetaeyhattucmehrgtaioenttusruieadrfoetolhmetntedsifwrohuteleitds

2.3 Consider a very simple symmetric block encryption algorithm, in which 64-bits blocks of plaintext are encrypted using a 128-bit key. Encryption is defined as

$$C = (P \oplus K_0) \boxplus K_1$$

Where C = ciphertext; K = secret key; K_0 = leftmost 64 bits of K; K_1 = rightmost bits of K, $\oplus$ bitwise exclusive and $\boxplus$ = addition mod 2^{64}

(a) Show the decryption equation. That is, show the equation P as a function of C, K_1 and K_2

$$\boxed{(C \boxminus K_1) \oplus K_0 = P}$$

(b) Suppose an adversary has access to two sets of the plaintexts and their corresponding cichertexts and wishes to determine K. We have the two equations:

$$C = (P \oplus K_0) \boxplus K_1; C^{'} = (P^{'} \oplus K_0) \boxplus K_1$$

First, derive an equation in one unknown. Is it possible to proceed further to solve for K_0?

$$K_0 = (C \boxminus K_1) \oplus P$$

By plugging the above into the second equation for K_0, one could simplify the algebra and solve for K_1. Then take that value and plug back into the first equation to solve and get K_0