Question 1

(2 marks) Let Σ = {0, 1} and consider languages A = {01, 00, 1}, B = {10, 11, 0}.
(a) Write down all strings in the set A · B. How many strings there are in A · B?
(b) Write down all strings in the set B · A. How many string there are in B · A?

part (a)

{0110,0111,010, 0010,0011,000, 110,111,10}
there are 9 strings in A·B

part (b)

{1001,1000,101, 1101,1100,111, 001,000,01}
there are 9 strings in B·A

Question 2

(3 marks) In this question the alphabet is Σ = {0, 1}. Let R = (00 + 10*1)* and S = (10*1 + 0*10*)*.
(a) Give two examples of a string z that is both in R and in S (that is, z ∈ R ∩ S).

(b) If possible, give two examples of a string x that is in R and is not in S (that is, x ∈ R ∩ S where S is the complement of S). If no such strings exist, write “R ∩ S does not have two strings”.

(c) If possible, give two examples of a string y that is in S and is not in R (that is, y ∈ R ∩ S). If no such strings exist, write “R ∩ S does not have two strings”.

In each case briefly explain (using natural language) why your example strings have the required property.

part (a)

z_1 = 101
z_2 = 11

both of these are in R and S because the concatenation within allows us to [ignore 00] and then create a string that will match [(10*1 + 0*10*)*, ignoring 0*10*].

part (b)

x_1 = 00
x_2 = 0000

these 2 strings are only in R as it is impossible to make a string solely from 0's in S as 1 is contained in both parts of the concatenation and none of the 1's have closure, meaning they cannot be removed.

part (c)
----

y_1 = 010
y_2 = 0001000

these 2 strings are only in S because it is impossible to create a string with a
a string in R with a substring of '1' with a suffix and prefix of [0*] as only
one part of the concatenation in R has a substr '1', and there are 2 of them as the prefix
and suffix with a substring of [0*]. none of the 1's in R have closure thus it is
impossible to make the above strings using said set.

Question 3
==========

(5 marks) Show how to define the following languages over Σ = {0, 1} using only ε, the
alphabet symbols 0 and 1, and the operations of union, concatenation, and closure.
Note: Your answer cannot use the intersection or complementation operation.
Below “or” always means “inclusive or”.
(a) All strings that have both 000 and 111 as a substring.

(b) All strings that have 0000 or 1111 as a substring.

(c) All strings that both begin and end with 0110. (Note that the prefix 0110 and the
suffix 0110 may overlap.)

(d) All strings that do not have 111 as a substring.

(e) All strings that have even length and, at the same time, have 010 as a substring.

part (a)

A = (0+1)*(111)(0+1)*(000)(0+1)* + 111000 + 000111

part (b)

B = (0+1)*(1111+0000)(0+1)* + 1111 + 0000

part (c)

C = 0110(0+1)*0110 + 0110110 + 0110

part (d)

D = (0+01+10+101+010)*(1+ε)

part (e)

E = (00+01+10+11)*(010)(1+0)

Question 4
==========

(2 marks) Let Σ = {a, b} and consider the state-transition diagram given in Figure 1.
Figure 1: State-transition diagram for Question 4.

(a) Give examples of three strings that are accepted by the state diagram and examples of three strings that are not accepted by the state diagram.

(b) Write out explicitly the transition table (or transition function) that defines the state transitions of the diagram.

part (a)

```
        accepted = {ab,abbab,abbba}
       ¬accepted = {abb,abba,a}
```

part (b)
----

state \ input	a	b
A	A	B
B	B	C
C	A	B

Question 5
==========

(3 marks) Let Σ = {a, b, c, d} and consider the nondeterministic state diagram with ε-transitions given in Figure 2.
Using the systematic method described in the lectures (and in the text), convert the state diagram into an equivalent (non)deterministic state diagram without ε-transitions.
You should not further modify/simplify the resulting state diagram.

in order to transform this ε-NFA into an NFA we must first remove all ε-transitions as well as any states that only recieve ε-transitions.

```
copy                                                  NOTE: [↓x↑] is a self loop over x
diagram      />----------ε----------→\
        → (E) ←ε<//>a→ ((F)) ←b<//>c→ (G)
          ↓b↑                         ↓d↑

remove  → (E) -a→ ((F)) ←b<//>c→ (G)
ε-trans   ↓b↑                    ↓d↑

                             />——d--→\
relink  → (E) ←b<//>a→ ((F)) ←b<//>c→ (G)
e-trans   ↓b↑           ↓b↑           ↓d↑
states
```

make final ε-trans final states final states in copy. there is nothing to change for this step.

thus the final NFA sans ε-transitions:

```
                             />——d--→\
        → (E) ←b<//>a→ ((F)) ←b<//>c→ (G)
          ↓b↑           ↓b↑           ↓d↑
```

Question 6

(5 marks) Let Σ = {a, b}. Using the systematic method described in the lectures (the subset construction), convert the nondeterministic state diagram given in Figure 3 into a deterministic state diagram. Your answer should indicate how the deterministic state diagram is obtained from the nondeterministic one: the states of the deterministic diagram should be labeled by sets of states of the nondeterministic diagram.

to use the state subset algorithm we must first create a state transition diagram using subsets of the states.

	input	a	b
state	A	BC	A
	BC	A	BD
	BD	BA	BD
	BA	BA	BA

now that we have the transition table, we can make the a DFA for this NFA

```
                                          ↓b↑
        → (A) ←a<//>b→ (BC) -b→ (BD) -a→ (BA)
          ↓a↑                    ↓b↑      ↓a↑
```

thus, this is the DFA derived from the given NFA using the subset algorithm