1) Design a DFA for the set of all the strings over the alphabet {a, b} such that all the strings start with zero or more of { bbb , bab } and ends with bbb.

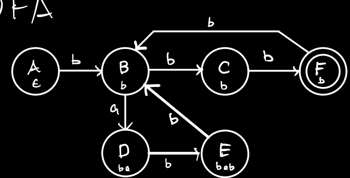
1) What is the NFA for the language
2) What is the transition table for the NFA
3) What is the transition table for the DFA
4) Design the DFA based on the step d

I think the question means:

$(bbb+bab)^* bbb$

$L = \{ \epsilon, bbb, bbbbbb, babbbb, babbbbbbb, \dots \}$

① NFA



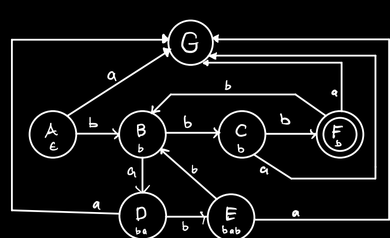
② NFA TT

	a	b
A	∅	B
B	D	C
C	∅	F
D	∅	E
E	∅	B
F	∅	B

③ DFA TT

	a	b
A	G	B
B	D	C
C	G	F
D	G	E
E	G	B
F	G	B
G	∅	∅

④ DFA



2) Write the regular expression for the following over {a,b} or {0,1}

1) Strings with exactly three occurrences of ba.

$L = \{ \square\, ba\, \square\, ba\, \square\, ba\, \square \}$

$\square \to b^*,\ a^*,\ (ab)^*$ — covered by a^*b

$R.E = [(a^* b^*)\, ba]^3\, (a^* b^*)$

2) Strings with at most 3 occurrences of ba

$L = \{ \square (ba + E)\, \square (ba + E)\, \square\, (ba + E)\, \square \}$

$\square \to a^*,\ b^*,\ (ab)^*$ — covered by a^*b^*

$R.E = [(a^* b^*)(ba + E)]^3\ a^* b^*$

3) L = { a^n b^m | m + n is even}

$n+m = 0, 2, 4, 6, \dots$ even strings

$L = \{ \epsilon, aa, ab, ba, bb, aaaa, aabb, \dots \}$

$R.E = [(a+b)(a+b)]^*$

$= (aa, ab, ba, bb)^*$

4) Set of all the strings with exactly 2 a’s over {a,b}

$L = \{ \square\, a\, \square\, a\, \square \}$

$\square \to b^*$

$R.E. = (b^* a)^2 b^*$

5) All the strings containing the substring ccc over the set {a, b, c}

$L = \{ \square\square\, ccc\, \square \}$

$(a+b)^*$ $(ca)^*$, $(cb)^*$, $(cca)^*$, $(ccb)^*$ $(a+b+c)^*$

$$R.E. = [(a+b)^* ((ca)^* + (cb)^* + (cca) + (ccb))]^* ccc(a+b+c)^*$$

6) set of all the strings when ever run of ‘a’ has a length of 4 over {a,b, c}.

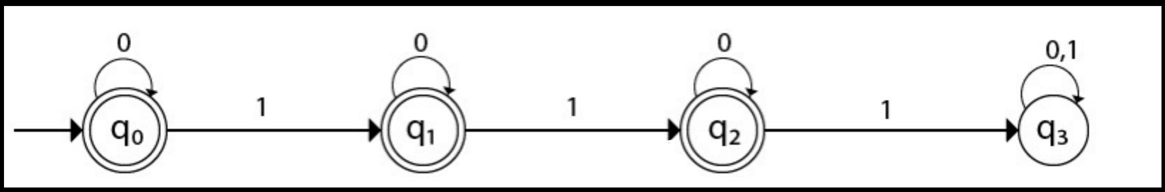
I think the question means:

every substring starting w/ 'a' has a length of 4

$$L = \epsilon,\ \underbrace{\boxed{}}_{(a+b)^*}\, a\, (\underbrace{\boxed{}}_{(a+b)^*})^3$$

$$R.E. = \epsilon + (a+b)^* a (a+b)^3$$

3) Create the regular expression for the given DFA. Must use the process of creating an equation for each state then solving the equation for the final state.



$$\boxed{\begin{array}{l}\text{Arden's Theorem:}\\ \text{If } R \neq E,\\ R = Q + RP \rightarrow R = QP^*\end{array}}$$

step 1: simplify q_0, q_1, q_2, q_3 into final equations

$q_0 = q_0 0 + E$

$q_0 = 0^*$

$q_1 = q_0 1 + q_1 0$

$q_1 = q_0 1(0^*)$

$q_2 = q_1 1 + q_2 0$

$q_2 = q_1 1(0)^*$

$q_3 = q_3 0 + q_3 1 + q_2 1$

$q_3 = q_3(0+1) + q_2 1$

$q_3 = q_2 1(0+1)^*$

step 2: solve by plugging final equations into each other

$\begin{cases} q_0 = 0^* \\ q_1 = q_0 1(0^*) \end{cases}$

$q_1 = 0^* 1 0^*$

$\begin{cases} q_1 = 0^* 1 0^* \\ q_2 = q_1 1(0)^* \end{cases}$

$q_2 = 0^* 1 0^* 1 0^*$

$q_2 = (0^* 1)^2 0^*$

$\begin{cases} q_2 = (0^* 1)^2 0^* \\ q_3 = q_2 1(0+1)^* \end{cases}$

$q_3 = (0^* 1)^2 0^* 1 (0+1)^*$

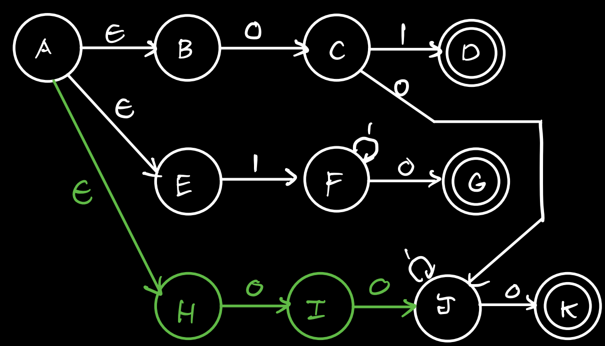
$q_3 = (0^* 1)^3 1 (0+1)^*$

4) Create the NFA for the following regular expression.
Create the DFA from the NFA you created
$01 + (1 + 00)1^* 0$

NFA (Systematic approach)

$01 + (1 + 00)1^* 0$

$L = \{01, 1(1^*)0, 00(1^*)0\}$

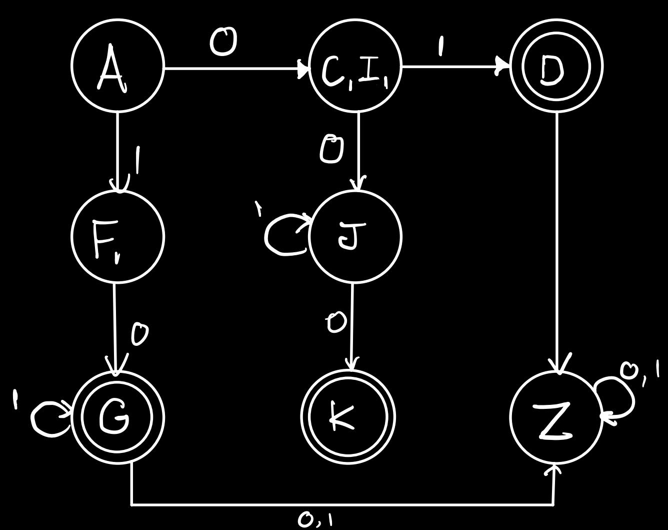


not needed
but I used it to solve

NFA TT

	0	1	ε
A	Ø	Ø	B,E,H
B	C	Ø	
C	Ø	D	
D	Ø	Ø	
E	Ø	F	
F	G	F	
G	Ø	Ø	
H	I	Ø	
I	J	Ø	
J	K	J	
K	Ø	Ø	

DFA



DFA TT

	0	1
A	CI	F
CI	J	D
F	G	F
D	Z	Z
G	Z	Z
J	K	J