My AUA ID is A09170218

Question 1:

The alphabet is integers from 0 to 9 inclusively ∑\_10 = {0, 1, 2, …, 9}.

A number is divided by 8, if its last 3 digits are divisible by 8. For example, 17216 is divisible by 8 as 216 is divisible by 8.

Inputs represent numbers written in the reversed order 🡺 the decision to accept / reject is done based on the reading of the first three symbols (digits) from the left.

Let *q\_a* be the accept state, *q\_r* reject state, ⇒ δ(*q\_r*, *a*) = *q\_r* and δ(*q\_a*, *a*) = *q\_a*, where *a* ∈ Σ.

Let *q\_s* be the start state.

Let *q\_b* be (b is for bad) the state in which δ(*q\_b*, *a*) = *q\_b* where *a* ∈ Σ.

Let *q\_i\_j* be the state in which i digits have been read and remainder of them mod 8 is j.

We need to consider only these states {q\_1\_0, q\_1\_2, q\_1\_4, q\_1\_6, q\_2\_0, q\_2\_4}

because q\_1\_1 is impossible and same is applied for q\_1\_3 q\_1\_5, q\_1\_5 because if we read the digit in the ones place of a number, and it's 1,3,5,7,9 then the number is not divisible by 8 🡺 so if we ever read 1,3,5,7,9 at the beginning, we go to q\_b 🡺 which means the only states we care about are q\_1\_0, q\_1\_2, q\_1\_4, q\_1\_6.

Now, as for the states associated with reading a 2nd digit, we just try all the possibilities (😊)

Having an odd number as the remainder means we've failed 🡺 only q\_2\_0, q\_2\_2, q\_2\_4, q\_2\_6 are relevant.

But one point in modular arithmetic is important. Since 100 = 4 (mod 8) 🡺 q\_2\_2, q\_2\_6 are useless too and in fact, instead of going to them, we should just fail because if a number ends with 82, then

82 = 2 (mod 8)

182 = 6 (mod 8)

282 = 2 (mod 8)

382 = 6 (mod 8)

482 = 2 (mod 8) etc...

As 100 = 4 (mod 8) so every time we increase the hundreds digit by 1, we increase the remainder mod 8 by 4 that's how we cancelled away q\_2\_2 and q\_2\_6.

Q = {q\_s, q\_1\_0, q\_1\_2, q\_1\_4, q\_1\_6, q\_2\_0, q\_2\_4, q\_b, q\_a}

F = {q\_a}

S= q\_a

Σ = {0, 1, 2, …, 9}

The DFA is as follows:

CHECK THE PICTURES Q1 and Q2

…………………………………………………………………

Question 2:

All we need to do is that we prove:

s1 and s4 are equivalent, s2 and s5 are equivalent too. (This is obvious as 30 = 0 (mod 6)).

And the transitions from s2 which are like (10\*2 + a) mod 6 are equivalent if the 2 were replaced with a 5, since it is equivalent to adding 0 = 30 (mod 6).   
Meaning 10\*2 + a = 10\*5 + a (mod 6) [Proving them is so easy, we just need to apply the definition]

Cancelling the s4 and s5 states completely 🡺

1. Any transition to s4 would be made to go to s1.
2. And any transition to s5 would be made to go to s2.
3. So basically, we end up having 4 states s0, s1, s2 and s3.

…………………………………………………………………………

Question 3:

It is obvious that 0 and 1 are cyclic numbers by themselves.

∑ = {0, 1, …, 9}

We can construct a DFA which branches to different sections based on the first digit read. If the first we see is a 0, we will only accept if we match 0's after.

If the first we see is a 1, we will only accept if we match 1's after.

Otherwise, the first digit is one of 2,3,4,5,6,7,8,9. In each of these cases, the algorithm actually yields exactly one number. The following table outlines which numbers are associated with which digit seen:

2 | 105263157894736842

3 | 1034482758620689655172413793

4 | 102564

5 | 102040816326530612244897959183673469387755

6 | 1016949152542372881355932203389830508474576271186440677966

7 | 1014492753623188405797

8 | 1012658227848

9 | 10112359550561797752808988764044943820224719

Hence, consider the following DFA:

We will have a start state q0, and we will have 2 states q1 and q2 (which will be accepting states), and a third state q\_fail that δ(*q\_fail*, *a*) = q\_fail for any a from the alphabet. 🡺

δ (q0, 0) = q1

δ (q0, 1) = q2

δ (q1, x) = q1 if x is 0, otherwise q\_fail

δ (q2, x) = q2 if x is 1, otherwise q\_fail

This handles the case of 0 and 1 as starting digits.

Let S(k) be the string associated with the digit k, in reverse. Let q\_{match, k, i} be a state representing that we have matched i characters in the S(k) so far. Then:

for each k from 2 to 9, δ (q0, k) = q\_{match, k, 1}

for each k from 2 to 9 and i from 1 to |S(k)| - 1, δ (q\_{match,k,i}, x) = q\_{match,k,i+1} if x = |S(k)|\_{i+1} and q\_fail otherwise.

Also, q\_{match,k,|S(k)|} is an accept state, and δ (q\_{match,k,|S(k)|}, x) = q\_fail for all x because one extra matched character means we fail.

Finally, obviously, δ (q\_fail, x) = q\_fail for all x.

The accept states overall are F = {q1, q2, q\_{match,2,|S(2)|}, ..., q\_{match,9,|S(9)|}}