Midterm

`Question 1 (A)`

Given:

$(P \lor Q) \to R$ and $(P \to R) \land (Q \to R)$
Claim: They are logically equivalent.

Proof:

($\Rightarrow$ direction):

Assume $(P \lor Q) \to R$.

If P is true, then $P \lor Q$ is true $\to$ R is true $\to$ $P \to R$ holds.
If Q is true, then $P \lor Q$ is true $\to$ R is true $\to$ $Q \to R$ holds.

So, $(P \to R) \land (Q \to R)$.

($\Leftarrow$ direction):

Assume $(P \to R) \land (Q \to R)$.

If $P \lor Q$ is true, then either Q is true.
If P, then R (since $P \to R$)
if Q, then R

So,

$(P \lor Q) \to R$ holds.

`Question 1 (B)`

A, C, and D are tautologies.

`Question 1 (C)`

$P \to (Q \to R) \equiv P \to (\neg Q \lor R) \equiv \neg P \lor \neg Q \lor R \equiv \neg (P \land Q) \lor R \equiv (P \land Q) \to R$

So both expressions are logically equivalent.

`Question 1 (D)`

Let:

- C(x): "x owns a cat"
- L(x): "x loves animals"

Then:

- Premise 1: ∀ x (C(x) → L (x))
- Premise 2: ∃ x C (x)
- Conclusion: ∃ x L (x)

`Valid by universal instantiation and modus ponens` .

`Question 2 (A)`

If $\frac{1}{f(x)}$ is not continuous, then f(x) = 0 at some point

Because if f(x) ≠ 0 everywhere and is continiuius, then $\frac{1}{f(x)}$ is continuous.

It's Proved via contrapositive.

`Question 2 (B)`

Proof of contradiction:

Assume $\sqrt{2}$ * $\sqrt{3}$ = r, where r is rational and $\sqrt{2}$ is
irrational

Then:

$\sqrt{3}$ = $\frac{r}{\sqrt{2}}$

Since r is rational and $\sqrt{2}$ is irrational, $\frac{r}{\sqrt{2}}$
is irrational,
-> contradiction: $\frac{r}{\sqrt{3}}$ can't be both rational and irrational

So, the product must be irrational

`Question 2 (C)`

Using inclusion-exclusion:
Total solutions without constraints:

$\frac{(12+3-1)}{2}$ = $\frac{(14)}{2}$ = 91

Subtract:

- x > 5 -> $\frac{8}{2}$ = 28
- y > 6 -> $\frac{7}{2}$ = 21
- z > 7 -> $\frac{6}{2}$ = 15

91 - (28 + 21 + 15) = 27

`Question 2 (D)`

Total circular arrangements of 6 peole: (6 - 1)! = 120

bad cases (two specific peopel together)
(5 -1) ! * 2 = 48

120 - 48 = 72

`Question 3 (A)`

365 possible birthdays
Need at least 366 people to ensure a shared birthday

The answer is 366

`Qestion 3 (B)`

- Total 8-letter passwords with no repeated letters
  Total (no restriction)=P(26,8)=26×25×24×23×22×21×20×19
- We must choose and arrange 8 distinct consonants from the 21 available.
  All consonants (no vowels)=P(21,8)=21×20×19×18×17×16×15×14
- Subtract to get number of valid passwords (at least one vowel)
  Valid passwords=P(26,8)–P(21,8)

$$26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 20 \times 19 - 21 \times 20 \times 19 \times 18 \times 17 \times 16 \times 15 \times 14$$

Which is

$$54{,}786{,}211{,}200$$

`Question 3 (C)`

Given: x_1 + x_2 + x_3 + x_4 = 20

Each x_i > 1 and x_i ≠ 2, 3

Let x_i >= 4: so define y_i = x_i - 4 -> y_i >= 0

y_1 + y_2 + y_3 + y_4 => $\frac{4+4-1}{3}$ = $\frac{7}{3}$ = 35

`Question 4 (A) i`

DFA for accepting binary strings containing `"101"` :

- States: q0 (start), q1, q2, q3 (accepting)
- Alphabet: {0, 1}
- Transitions:
  - q0 —1→ q1, q0 —0→ q0
  - q1 —0→ q2, q1 —1→ q1
  - q2 —1→ q3, q2 —0→ q0

  - q3 —0/1→ q3 (stay in accepting)
- Accepting State: q3
- Accepts strings containing `"101"` as a substring.

`Question 4 (A) ii`

String: "1101"

- Loops on q_1 with extra 1s
- Exits to accept via "01"

`Question 4 (A) iii`

Modify DFA to accept even number of 1s.
Track parity: even ↔ odd ↔ even.

New string accepted: "1100"
Not accepted under old DFA (no "101").

`Question 4 (B)`

Regex for: no "11", ends in "0":

(0+10) * 0

All binary strings that do not contain "11" and end with

`Question 5 (A)`

- abc123
- A1b2C3
- z9

`Question 5 (B)`

Regex: (0 + 1) * 101 (0 + 1)*

Then filter out strings containgin "111"

`Question 5 (C)`

Use Pumping Lemma to show:

L = {a^p | p is prime} is not regualar.

Assume regular -> pumping length p

Pick {a^p | p is prime} is not regular

Assume regual -> pumping length p

Pick a^q, where q > p,q prime

Split a^q = xyz, pump y: get length non-prime → contradiction.

Not regular

`Question 5 (D)`

Let:

- G^1 : S -> aSb | ε
- G^2 : S -> aS | b

String "aabb" ∈ G_1 (balanced recursion)

Not ∈ G_2 (unbalanced structure)

"aabb" ∈ L(G_1), but not ∈ L(G_2)