Question 1

(a) Shrek is six feet tall - proposition

(b) Dr. Cantu and his son - neither

(c) Give an example of an integrable function - neither

(d) $20^2 + 23^2 > 2023^2$ - proposition

(e) $x^2 = 1$ - predicate

(f) Potatoes are awesome - neither

(g) n is a perfect square - predicate

(h) The product of every two prime numbers is odd - proposition

Question 2

(a) P: false

(b) Q: true

(c) $P \vee Q$: true

(d) $P \wedge Q$: false

(e) $P \to Q$: true

(f) $Q \to P$: false

Question 3

For the predicate $P(x) : (x^2 - 9)(x - 1) = 0$ where $x \in \mathbb{R}_{\geq 1}$, x must be either 3 or 1 for $P(x)$ to be true.

Question 4

(a) $\sqrt{3}$ is irrational

(b) 0 is a negative number

(c) The real number r is greater than π

Question 5

(a) **Hypothesis:** a is irrational
Conclusion: $2a$ is irrational

(b) **Hypothesis:** a is an even integer
Conclusion: a^3 is an even integer

(c) **Hypothesis:** $\lim_{x\to 0^+} f(x) = 3$
Conclusion: $\lim_{x\to 0} f(x) = 3$

Question 6

(a)

P	Q	R	$P \vee Q$	$(P \vee Q) \wedge R$	$P \wedge R$	$Q \wedge R$	$(P \wedge R) \vee (Q \wedge R)$
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	F
T	F	T	T	T	T	F	T
T	F	F	T	F	F	F	F
F	T	T	T	T	F	T	T
F	T	F	T	F	F	F	F
F	F	T	F	F	F	F	F
F	F	F	F	F	F	F	F

(b)

P	Q	$P \to Q$	$\neg(P \to Q)$	$\neg Q$	$P \wedge (\neg Q)$
T	T	T	F	F	F
T	F	F	T	T	T
F	T	T	F	F	F
F	F	T	F	T	F

Question 7

(a)
$P \wedge \neg P$ is a contradiction.

(b)

$$\begin{aligned}
&P \to (Q \to P) \\
\equiv &P \to (\neg Q \lor P) \\
\equiv &\neg P \lor (\neg Q \lor P) \\
\equiv &(P \lor \neg P) \lor \neg Q \\
\equiv &T \lor \neg Q \\
\equiv &T
\end{aligned}$$

This is a tautology.

Question 8

(a) I watched 3Blue1Brown's videos on gradient descent and "How to lie using visual proofs."

(b) The gradient descent video was about how deep neural networks learn. Basically, it gave an example of a model that was trying to classify numbers based on handwritten digits. Without getting into too much jargon, the basis for training the model was to minimize the "cost function", which is a measure of how how different the model's output is from the expected output. The cost function is a function of the weights of the model, and typically has thousands or millions of input variables. Finding the minimum of this is much more difficult than in $\mathbb{R}^2$ or $\mathbb{R}^3$, so we use a method called gradient descent. The idea is that we start at some random point, and since the gradient tells us the direction of steepest ascent, we can take a step in the opposite direction to get closer to the minimum. We repeat this process until we get close enough to the minimum, which is where the model is optimized.