Problem 1
Convert to prose

\( S \): the set of all snakes
\( R \): the set of all rabbits
\( E(x, y) \): \( x \) eats \( y \)
\( Y(x) \): \( x \) is yellow

Convert the following to simple, readable English:

1. \( (\exists r \in R, s \in S . E(r, s)) \rightarrow (\neg \forall s \in S . \exists r \in R . E(s, r)) \)

2. \( \forall r \in R, s \in S . (Y(s) \rightarrow \neg E(s, r)) \land (Y(r) \rightarrow E(r, s)) \)

3. \( \forall s_1 \in S . \exists s_2 \in S . \forall s_3 \in S . Y(s_1) \rightarrow (\neg E(s_2, s_3) \land E(s_1, s_2) \land \neg Y(s_2)) \)

Problem 2
Primes and factors

4. ________________ is the prime factorization of 28

5. ________________ is the prime factorization of 256

6. ________________ is the prime factorization of 31

7. ________________ is the prime factorization of \(4^8 \cdot 14^9\)

8. ________________ is the set positive 1-digit numbers relatively prime with 15

9. ________________ is the set positive 1-digit numbers relatively prime with 81
PROBLEM 3 Symbolic proof by contradiction

Write a symbolic proof outline of the following, using proof-by-contradiction.

10. $\frac{2}{3} \notin \mathbb{Z}$

11. $\sqrt{2} \notin \mathbb{Q}$
**PROBLEM 4 Prose from symbols**

Write a prose proof that follows the given symbolic proof outlines.

Assume \( \frac{5}{8} \in \mathbb{Z} \)

\[ \exists x \in \mathbb{Z} . \frac{5}{8} = x \] definition of set membership

\[ \frac{5}{8} = x \] existential instantiation

\[ 5 = 8x \] algebra

12. 2 is a factor of 5 fundamental theorem of arithmetic

\( \bot \) contradiction

Ergo assumption false proof by contradiction

\[ \frac{5}{8} \notin \mathbb{Z} \] conclusion

Proof.
Assume $\sqrt[3]{4} \in \mathbb{Q}$

$\exists x, y \in \mathbb{Z} . \sqrt[3]{4} = \frac{x}{y} \land \gcd(x, y) = 1$  \hspace{1em} \text{definition of set rationals}

$\sqrt[3]{4} = \frac{x}{y}$  \hspace{1em} \text{existential instantiation}

$4y^3 = x^3$  \hspace{1em} \text{algebra}

$\neg(2 \mid x) \lor \neg(2 \mid y)$  \hspace{1em} \text{because } \gcd(x, y) = 1

<table>
<thead>
<tr>
<th>case 1: $\neg(2 \mid x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg(2 \mid x^3)$</td>
</tr>
<tr>
<td>$\bot$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>case 2: $\neg(2 \mid y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \mid x^3$</td>
</tr>
<tr>
<td>$2 \mid x$</td>
</tr>
<tr>
<td>$8 \mid x^3$</td>
</tr>
<tr>
<td>$\neg(8 \mid 2y^3)$</td>
</tr>
<tr>
<td>$\bot$</td>
</tr>
</tbody>
</table>

$\bot$  \hspace{1em} \text{case analysis}

$\bot$  \hspace{1em} \text{contradiction}

Ergo assumption false  \hspace{1em} \text{proof by contradiction}

$\sqrt[3]{4} \notin \mathbb{Q}$  \hspace{1em} \text{conclusion}

*Proof.*
PROBLEM 5 Proof by contradiction

Prove the following using proof-by-contradiction. You may prove them in prose or in symbols or any readable mix of the two.

14. $\sqrt{2} \notin \mathbb{Z}$

Proof.

15. $2^{-1} \notin \mathbb{Z}$

Proof.
16. $\sqrt{7} \notin \mathbb{Q}$

Proof.

17. $3^{1.5} \notin \mathbb{Q}$

Proof.
PROBLEM 6  Additional problems

18. Prove there are infinitely many prime numbers. Use \( p' = 1 + \prod_{p \in P} p \) where \( P \) is the set of all primes to derive the contradiction (e.g. by showing both that \( p' \in P \) and \( p' \notin P \)).

19. Prove there are infinitely many integers. Use \( z + 1 \) where \( z \) is the largest integer to derive the contradiction.

20. Prove there are infinitely many finite-length strings containing the digits 0 and 1. Use the concatenation of \( s \) and \( s' \), where \( z \) is one of the strings of maximal length, to derive the contradiction.

21. Prove there are infinitely many finite natural numbers. Use \( n + 1 \), where \( n \) is the largest finite natural number, to derive the contradiction.

22. Prove that \( \forall n \in \mathbb{N} . 4|(5^n - 1) \). Use the well-ordering principle to derive a contradiction by showing that if \( m > 0 \) is the smallest \( n \) that makes the expression false, then \( m - 1 \) also makes it false. Include a case that shows that the expression holds for \( n = 0 \).

23. Prove that \( \forall n \in \mathbb{Z}^+ . p_1 \land p_2 \land \ldots \land p_m \equiv p_1 \lor p_2 \lor \ldots \lor p_m \). Use the well-ordering principle to derive a contradiction by showing that if \( m > 1 \) is the smallest \( n \) that makes the expression false, then \( m - 1 \) also makes it false. Include a case that shows that the expression holds for \( n = 1 \).

24. Prove there is no smallest positive real number. Use the well-ordering principle to derive a contradiction by showing a smaller positive real number than the smallest positive real. Tools like \( n \div 2 \) or \( n \times n \) might help.

25. Prove there is no real number that is closest to, but not the same as, \( x \). Use the well-ordering principle to derive a contradiction by showing a closer real number than the closest real. Tools like \( \frac{x+y}{2} \) might help.

26. Prove there is no best rational approximation of \( \sqrt{2} \) by showing that, for every approximation \( x \), the value \( \frac{\sqrt{2} + \frac{1}{3}}{2} \) is a better approximation; you may need to a lemma to show that that \( \forall x \in \mathbb{Q} . \frac{\sqrt{2} + \frac{1}{3}}{2} \neq x \).

27. Prove that \( \forall x \in \mathbb{Z} . (x + 1)(x - 1) = x^2 - 1 \) without using the distributive law of multiplication. Instead show that it holds for some \( x \) (pick any you wish) and that there’s no largest or smallest \( x \) for which it does not hold.

28. Prove that there is no largest two-argument function \( f(x, y) \) that returns \( x + y \) in the programming language of your choice. Do this by showing that if there was a largest program, you can make a larger one that has the same behavior.

29. Prove that there is no most-complicated two-argument function \( f(x, y) \) that returns \( x + y \) in the programming language of your choice, where complication is measured by the number of \( \text{if} \) statements and loops. Do this by showing that if there was a most complicated program, you can make a more complicated one that has the same behavior.

30. Prove that there is no longest-running two-argument function \( f(x, y) \) that returns \( x + y \) in the programming language of your choice. Do this by showing that if there was a most longest-running program, you can make a program that takes longer to execute and has the same behavior.