The function *rnalen* that computes the length of RNA strands in S is defined recursively by $rnalen : S \to \mathbb{Z}^+$

Basis step: If $b \in B$ then $rnalen(b) = 1$

Recursive step: If $s \in S$ and $b \in B$, then $rnalen(sb) = 1 + rnalen(s)$

The function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively by $basecount : S \times B \to \mathbb{N}$

Basis step: If $b_1 \in B$, $b_2 \in B$, $basecount(b_1, b_2) = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$

Recursive Step: If $s \in S$, $b_1 \in B$, $b_2 \in B$, $basecount(sb_1, b_2) = \begin{cases} 1 + basecount(s, b_2) & \text{when } b_1 = b_2 \\ basecount(s, b_2) & \text{when } b_1 \neq b_2 \end{cases}$

Prove or disprove $\exists s \in S\, (rnalen(s) = basecount(s, \mathtt{A}))$:

Prove or disprove $\forall s \in S\, (rnalen(s) \geq basecount(s, \mathtt{A}))$:

Proof by universal generalization: To prove that $\forall x\, P(x)$ is true, we can take an arbitrary element e from the domain and show that $P(e)$ is true, without making any assumptions about e other than that it comes from the domain.
New! Proof by Structural Induction (Rosen 5.3 p354) To prove a universal quantification over a recursively defined set:

Basis Step: Show the statement holds for elements specified in the basis step of the definition.

Recursive Step: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.

Definition The set of natural numbers (aka nonnegative integers), $\mathbb{N}$, is defined (recursively) by:

Basis Step:	$0 \in \mathbb{N}$
Recursive Step:	If $n \in \mathbb{N}$ then $n+1 \in \mathbb{N}$ (where $n+1$ is integer addition)

The function *sumPow* with domain $\mathbb{N}$, codomain $\mathbb{N}$, and which computes, for input i, the sum of the first i powers of 2 is defined recursively by $sumPow : \mathbb{N} \to \mathbb{N}$ with

Basis step: $sumPow(0) = 1$.

Recursive step: If $x \in \mathbb{N}$ then $sumPow(x+1) = sumPow(x) + 2^{x+1}$.

Prove or disprove $\forall n \in \mathbb{N}\,(sumPow(n) = 2^{n+1} - 1)$:

Extra example Connect the function *sumPow* to binary expansions of positive integers.

Definition The set of linked lists of natural numbers L is defined by:

Basis Step:	$[] \in L$
Recursive Step:	If $l \in L$ and $n \in \mathbb{N}$, then $(n, l) \in L$

Definition The function $append : L \times \mathbb{N} \to L$ that adds an element at the end of a linked list is defined by:

		$append : L \times \mathbb{N}$	$\to L$
Basis Step:	If $m \in \mathbb{N}$ then	$append([], m)$	$= (m, [])$
Recursive Step:	If $l \in L$ and $n \in \mathbb{N}$ and $m \in \mathbb{N}$, then	$append((n, l), m)$	$= (n, append(l, m))$