Question 1. [10 FAKE (not real; does not count) points]
Write a simplified expression (not a summation) that gives the number if binary strings of length less than or equal to $n$:

**Answer:** $2^{n+1} - 1$.

Question 2. [90 FAKE (not real; does not count) points] Let us define $HP = \{ i \mid i \in L(M_i) \}$. You may assume that $M_1, M_2, M_3, \ldots$ is a standard enumeration of Turing machines and that $HP$ (using exactly the definition above) is undecidable (we implicitly use the standard 1-to-1 bijection between $N$ and $\Sigma^*$ to treat $i$ both as integer and string).

Prove that $A = \{ i \mid L(M_i) \neq \emptyset \}$ is undecidable. (Do not invoke Rice’s Theorem. Rather, either use the so-called contradiction method or, better probably, prove (in doing that, one part will be to GIVE the reduction, and another will be to cover the three elements of proving that the function you gave in fact IS a reduction between these two sets) that $HP$ reduces to $A$ by some recursive many-one reduction (aka recursive mapping reduction).

(Side comment: This is an usually easy problem, since we don’t have much time for this quiz. But you should also be able to solve such harder problems as proving that, for example, $A = \{ i \mid L(M_i) \text{ is an infinite language} \}$ is undecidable. But on this quiz, I’m asking NOT about that but about Question 2.)

**Answer:**
To prove that $A$ is undecidable, it suffices to prove $HP \leq_m A$.

Namely, we need to construct a recursive reduction $\sigma$, such that

$$(\forall x)[x \in HP \iff \sigma(x) \in A].$$

We now define such a $\sigma$. $\sigma(x)$ outputs $j$ (yes, it implicitly is a function of $x$, but throughout I’ll just write it as $j$), where $j$ is the encoding of a Turing Machine $M_j$. The construction of the Turing Machine $M_j$ is as follows:

On input $y$, $M_j$ (ignores its own input $y$, and) immersively simulates $M_x(x)$. That is, if $M_x(x)$ accepts then $M_j$ halts and accepts. If $M_x(x)$ halts and rejects, $M_j$ halts and rejects. And since we’re doing an immersive simulation, if $M_x(x)$ runs forever then so will $M_j$ as it simulates $M_x(x)$.

1. $\sigma$ is clearly a recursive reduction.
2. $x \in HP \Rightarrow \sigma(x) = j \in A$. Why? Because when $x \in HP$, $M_x(x)$ accepts. Thus $M_j$ accepts all $y \in \Sigma^*$. Thus $L(M_j) = \Sigma^*$, which is infinite, and more to the point, is not the empty set. Thus we have $\sigma(x) \in A$.
3. $x \notin HP \Rightarrow \sigma(x) = j \notin A$. Why? $x \notin HP$ means $M_x(x)$ either runs forever or halts and rejects.

If $M_x(x)$ runs forever, then by simulating $M_x(x)$, $M_j(y)$ will also run forever for each $y$. Thus $L(M_j) = \emptyset$. Thus $\sigma(x) \notin A$.

If $M_x(x)$ halts and rejects, then due to it simulating $M_x(x)$, $M_j(y)$ will halt and reject for each $y$. Thus $M_j$ rejects all $y \in \Sigma^*$. Thus $L(M_j) = \emptyset$. Thus $\sigma(x) \notin A$.

So, we have shown $x \notin HP \Rightarrow \sigma(x) \notin A$ and thus completed our proof of item 3.

Since we have done 1/2/3, we now we have proven that $HP \leq_m A$, thus $A$ is undecidable.