Infinity and Countability

1. Which of the following are true for the cardinality of the set $\{1, 2, 3\}$?

   - Finite.
   - Countable.

   This is a finite set since it has only 3 elements. A set is countable if and only if there is a bijection between it and a subset of $\mathbb{N}$. But this is actually a subset of $\mathbb{N}$ itself. So the identity function is the desired bijection. In general any finite set is countable.

2. Which of the following are true for the cardinality of the set of multiples of 5?

   - Finite.
   - Countable.

   This set obviously has infinitely many elements since $5k$ for any $k \in \mathbb{N}$ is in it. The function $x \to \frac{x}{5}$ is a bijection between this set and the set of natural numbers. So this is a countable set.

3. Suppose that a set S is countable. Which of the following are true about S?

   - Any subset of S is also countable.
   - Any strict (not equal to S itself) subset of S is finite.
   - If we add an element to S it remains countable.

   By definition there is a bijection between S and some subset of $\mathbb{N}$. That same bijection restricted to any subset of S is also a bijection between that subset and some subset of $\mathbb{N}$.

   It is not true that any strict subset of S is finite. For example the set of natural numbers is countable, but the set of odd numbers is a strict subset which is not finite.

   If S is finite, adding an element to it makes it finite. If S is infinite, then there is a bijection between it and $\mathbb{N}$. Now when we add an element to S we can shift everything up by 1 (so if an element was being mapped to x, now it's mapped to $x + 1$), and map the new element to the lowest element of $\mathbb{N}$, i.e. 0.

4. Which of the following sets are uncountable?

   - The interval $[0, 1]$.
   - The total number of computer programs that could ever be written.
   - The interval $[0, 10]$.
   - The set of real numbers that have a digit 1 when written down.

   The interval $[0, 1]$ can be shown to be uncountable using Cantor's diagonalization method.

   Every computer program is simply a finite sequence of characters. If we assign each character a natural number, then a computer program is just a finite sequence of natural numbers. These are countable, since there is a bijection between them and the polynomials with natural coefficients which was shown to be a countable set.

   There is a bijection between the interval $[0, 10]$ and the interval $[0, 1]$. Namely $x \to \frac{x}{10}$.

   The set of real numbers that have a digit 1 in their representation obviously includes the set $[10, 11]$ which is uncountable.

5. In what cases does a set S have the same cardinality as the set S' obtained from S by adding an element to it?

   - If S is finite.
   - If S is countable and infinite.
   - If S is uncountable and infinite.

   When S is finite, then S plus one element has strictly larger cardinality.

   But when S is infinite, we can extract an infinite sequence out of it by the following method: we pick an element a_1; there must be another element a_2; there must be another element a_3, and so on. This process never stops because S is infinite. Now construct a bijection between S and S' by mapping each a_i to a_{i+1}, mapping the new element to a_1, and mapping other elements to themselves. This shows that S and S' have the same cardinality.

Computability

1. Why can't we solve the Halting problem by just simulating the given program on the given input and seeing if it stops?

   - Because we cannot simulate programs.
   - Because in case the answer is no, our simulation would never stop.

   We can indeed simulate programs. That is what virtualization softwares that most people are familiar with do.

   The reason is that we need to output an answer no matter whether it's Yes or No. But if a program is not going to halt (in which case the answer is No), our simulation also never stops.

2.