You may assume that $M_1, M_2, M_3, \ldots$ is a standard enumeration of Turing machines and that $HP = \{ i \mid i \in L(M_i) \}$ is undecidable (we implicitly use the standard 1-to-1 bijection between $\mathbb{N}$ and $\Sigma^*$ to treat $i$ both as integer and string).

**Question 1.** [50 FAKE (not real; does not count) points]
Prove that $A = \{ i \mid L(M_i) \text{ is an infinite language} \}$ is undecidable.

**Answer:**
To prove that $A$ is undecidable, we just need to prove $HP \leq_m A$.

Namely, we need to construct a recursive reduction $\sigma$, such that $(\forall x)[x \in HP \iff \sigma(x) \in A]$.

$\sigma(x)$ outputs $j$, where $j$ is the encoding of a Turing Machine $M_j$. The construction of the Turing Machine $M_j$ is as follows:

On input $y$, $M_j$ simulate $M_x(x)$ and if $M_x(x)$ accepts then $M_j$ halts and accepts. If $M_x(x)$ halts and rejects, then $M_j$ goes into an infinite loop.

1. $\sigma$ is a recursive reduction.
2. $x \in HP \Rightarrow \sigma(x) = j \in A$. Because when $x \in HP$, $M_x(x)$ accepts. Thus $M_j$ accepts all possible $y \in \Sigma^*$. Thus $L(M_j) = \Sigma^*$, which is infinite. Thus we have $\sigma(x) \in A$.
3. $x \notin HP \Rightarrow \sigma(x) = j \notin A$. $x \notin HP$ 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 any $y$. Thus $L(M_j) = \emptyset$. Thus $\sigma(x) \notin A$.

   If $M_x(x)$ halts and rejects, then according to the construction of $M_j$, $M_i(y)$ will go into an infinite loop and run forever for any $y$. Thus $L(M_j) = \emptyset$. Thus $\sigma(x) \notin A$.

   To sum up, we have $x \notin HP \Rightarrow \sigma(x) \notin A$.

   Now we have proved that $HP \leq_m A$, thus $A$ is undecidable.

**Question 2.** [50 FAKE (not real; does not count) points]
Prove that $B = \{ i \mid L(M_i) \text{ is a finite language} \}$ is undecidable.

**Answer:**
To prove that $B$ is undecidable, we just need to prove $HP \leq_m B$. 

1
Namely, we need to construct a recursive reduction $\sigma$, such that
$(\forall x)[x \in HP \iff \sigma(x) \in B]$.

$\sigma(x)$ outputs $j$, where $j$ is the encoding of a Turing Machine $M_j$. The
construction of the Turing Machine $M_j$ is as follows:

On input $y$, $M_j$ simulate $M_x(x)$ and if $M_x(x)$ does not accept within $|y|$ steps then $M_j$ halts and accepts. Otherwise $M_j$ rejects. Here $|y|$ is the length of the input $y$.

1. $\sigma$ is a recursive reduction.
2. $x \in HP \Rightarrow \sigma(x) = j \in B$. Because if $x \in HP$ then there exists a $k$ such that $M_x(x)$ accepts within $k$ steps. Thus for any input $y$ with length larger than $k$, $M_j$ rejects. Thus $L(M_j)$ is a finite language. Thus we have $\sigma(x) \in B$.

3. $x \notin HP \Rightarrow \sigma(x) = j \notin B$. $x \notin HP$ means $M_x(x)$ never accepts. Thus for all $y$, $M_x(x)$ does not accept within $|y|$ steps. Thus for all $y$, $M_j(y)$ accepts. Thus $L(M_j) = \Sigma^*$, which is infinite. Thus $\sigma(x) \notin B$.

Now we have proved that $HP \leq_m B$, thus $B$ is undecidable.