Exercise 3.1. Give a non-empty program in a programming language of your choice that does not take any external input and that prints out its own source code.

Hint: 

Exercise 3.2. Use the Recursion Theorem to give an alternative proof of Rice’s Theorem.

Exercise 3.3. (An alternative proof of the fixpoint version of the recursion theorem.) Let $f: \Sigma^* \rightarrow \Sigma^*$ be a computable function that maps Turing machine encodings to Turing machine encodings. Show that there exists a Turing machine $M$ such that $M(x) = f(M)(x)$, where $M(x)$ denotes the result of $M$ when running on input $x$.

Proceed as follows: denote with $g: \Sigma^* \rightarrow \Sigma^*$ the computable function that maps Turing machines $M$ to the Turing machine $M(\langle M \rangle)$, if this denotes a Turing machine, and otherwise to a Turing machine that always rejects. Let $M_f$ and $M_g$ be the Turing machines for $f$ and $g$, respectively. Show that $g(\langle M_f \circ M_g \rangle)$ is the desired fixpoint.

Exercise 3.4. Let $M_1, M_2, \ldots$ be an arbitrary enumeration of all Turing machines. Use the fixpoint formulation of the recursion theorem to show that there exists some $i \in \mathbb{N}$ such that both $M_i$ and $M_{i+1}$ compute the same function.

Exercise 3.5. Denote for $n \in \mathbb{N}$ with $\mathbb{Z}_n$ the ring of integers modulo $n$. Show that $\text{Th}(\mathbb{Z}_n)$ is decidable.

Exercise 3.6. Show that $\text{Th}(\mathbb{N}, <)$ is decidable by reducing $\text{Th}(\mathbb{N}, <)$ to $\text{Th}(\mathbb{N}, +)$. 