Homework problems:

1. You are being offered the following programming assignment:

   Intel Septium code optimisation
   A large company producing embedded systems software would like to have a code
   optimiser that, given as input any machine language program for the new Intel
   Septium processor, will output the smallest machine language program that is
   functionally equivalent to the given one (i.e. has the same input-output behaviour).

   Your comments? Under what conditions would you accept the assignment? Justify
   your answer.

2. Formulate Rice's general undecidability theorem and the related notions ("decidable
   properties of Turing machines" etc.) in terms of some programming language familiar
   to you (C, Java, etc.) rather than Turing machines. [If your textbook doesn't discuss
   this fundamental result, look it up in some reference work on computability theory or
   on the Internet.] — You do not need to try to reconstruct the *proof* of the theorem,
   although thinking about that would also be quite instructive.

3. Consider application programs running under some given operating system. Let us say
   that a program P is *dangerous*, if it on some input modifies the operating system's
   system files. A *general purpose virus tester* is a program that receives as input an
   arbitrary application program text P, analyses it and returns output "DANGER", if
   the program is dangerous, and "OK" otherwise. Show that if any dangerous programs
   exist at all, then general-purpose virus testing is impossible.

Demonstration problems:

4. Prove, without appealing to Rice's theorem, that the following problem is undecidable:

   $$\text{Given a Turing machine } M; \text{ does } M \text{ accept the empty string?}$$

5. Prove the following connections between recursive functions and languages:

   (i) A language $A \subseteq \Sigma^*$ is recursive ("Turing-decidable"), if and only its characteristic
       function
       $$\chi_A : \Sigma^* \to \{0,1\}, \qquad \chi_A(x) = \begin{cases} 1, & \text{if } x \in A; \\ 0, & \text{if } x \notin A \end{cases}$$
       is a recursive ("Turing-computable") function.

   (ii) A language $A \subseteq \Sigma^*$ is recursively enumerable ("semidecidable", "Turing-recognisable"),
        if and only if either $A = \emptyset$ or there exists a recursive function $g : \{0,1\}^* \to \Sigma^*$
        such that
        $$A = \{g(x) \mid x \in \{0,1\}^*\}.$$