Starred exercises are optional.

1) Which of the following specifications *defines* f as a (total) function on the natural numbers? Explain why (not). In case it does, compute $f(5)$ (you may use a program). Below, n and m range over the set of natural numbers (which includes 0):

   a) for all n, $f(n+1) = f(n) + 1$;

   b) $f(0) = 17$ and for all n, $f(n+1) = f(n) + 1$;

   c) $f(0) = 17$ and for all n, $f(n+1) + 1 = f(n)$;

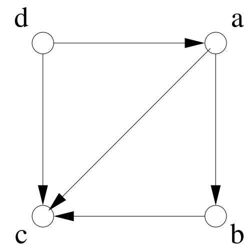
   d) $f(0) = 1$, $f(1) = 1$, and for all n, $f(n+2) = f(n) + f(n+1)$;

   e∗) for all n, $f(n) = g(n,n)$ where for all n and m, $g(0,n) = n+1$, $g(m+1,0) = g(m,1)$, and $g(m+1,n+1) = g(m, g(m+1,n))$;

   f) for all n, $f(n)$ is the smallest number greater than all $g(n)$, where g ranges over all defined functions in the other items of this exercise;

   g) for all n, $f(n) = f(n)^2$.

2) Consider the relation R on $\{a,b,c,d\}$ whose graph is



   Use Warshall's algorithm to compute the transitive closure R^+ of R. Read off from the resulting matrix whether $a\ R^+\ c$ holds, $c\ R^+\ a$ holds, and whether $d\ R^+\ b$ holds. How does (the matrix of) R^+ relate to the distances computed in exercise 3) of the 1st Exercise sheet?

3) Let $T = \{n \mid 2 \leq n \leq 12\}$, the set of natural numbers between 2 and 12. Let $|_T$ be the T-*divisibility* relation on T defined by $x \mid_T y$ if for some $x' \in T$, $x \cdot x' = y$.

   - Draw the digraph G_T of $|_T$ and show that $|_T$ is irreflexive and transitive.
   - Repeat the following transformation on graphs, starting with G_T, until no transformation can be performed:
     - if there is an edge e from n to m, and also a path from n to m not containing e, then remove e from the graph.

   Draw the final graph G'_T, and argue that the procedure terminates and the graph obtained does not depend on the order in which edges were chosen. How does G_T relate to G'_T?

4∗) Implement some function in two distinguishable ways. More precisely, choose a function $f : D \to C$ for some domain D and co-domain C, and implement f in two ways f_1 and f_2 in a programming language of your choice, such that f_1 and f_2 can be distinguished from each other by suitable experiments, i.e. by applying them to some elements $c, c', \ldots$ of C and observing a difference between the runtime behaviour of $f_1(c), f_1(c'), \ldots$ and of $f_2(c), f_2(c'), \ldots$.