Homework 3 — Due February 17 in class

1. Recall the definition of a 1-counter machine (with input numbers rather than input strings). Such a machine M has a list of states, and one counter called "x" which stores a natural number. The "input" to the machine is simply the initially value of x. Each state of M is an instruction of one of the following four forms:

   - increment x and goto state i;
   - decrement x and goto state i;
   - if $x = 0$ goto state i else goto state j;
   - halt.

   (We hope the semantics are clear; the only tiny catch is that if the counter holds 0 and it's decremented, it just stays 0.)

   Prove that for this model of computation, the halting problem is decidable. That is, there is an algorithm (which you may describe in pseudocode) which, on input $\langle M, x\rangle$ (where M is a 1-counter machine and $x \in \mathbb{N}$) correctly decides whether or not $M(x)$ halts or loops forever.

   (Thus 1-counter machines *cannot* simulate Turing Machines!)