input started. These values can be arbitrary, but the choice does impact how the system responds to a given input. **One** choice gives rise to a linear system: Make the initial conditions zero. The reason lies in the Definition of a linear system (Section

2.6.6: Linear Systems): The only way that the output to a sum of signals can be the sum of the individual outputs occurs when the initial conditions in each case are zero.

###### Exercise 5.12.1

The initial condition issue resolves making sense of the diference equation for inputs that start at some index. However, the program will not work because of a programming, not conceptual, error. What is it? How can it be "fxed?"

###### Example 5.5

Let's consider the simple system having *p* =1 and *q* =0.



(5.43)

To compute the output at some index, this diference equation says we need to know what the previous output *y (n − 1)* and what the input signal is at that moment of time. In more detail, let's compute this system's output to a unit-sample input: *x (n)=* δ *(n).*

Because the input is zero for negative indices, we start by trying to compute the

output at *n* = 0.



(5.44)

What is the value of *y (−1)*? Because we have used an input that is zero for all negative indices, it is reasonable to assume that the output is also zero. Certainly, the diference equation would not describe a linear system (Section 2.6.6: Linear Systems) if the input that is zero for **all** time did not produce a zero output. With this assumption, *y (−1) = 0*, leaving *y* (0) = b. For n> 0, the input unit-sample is zero, which leaves us with the diference equation *y (n)*= *ay (n* − 1) , *n*> 0 . We can envision how the flter responds to this input by making a table.



(5.45)