Mark Floryan
Sample Assignment – Description

This problem reduces to finding the second smallest number in a list of numbers of size n. I approached this problem by creating an array (A) of size n to store the numbers. Then, I create two variables: low and sec. Low represents the lowest number in the list, which is maintained in the usual way: We loop through the array and maintain the lowest value we've seen so far. Sec represents the second lowest value we've seen so far.

We initialize these values to low = min(A[0],A[1]) and sec = max(A[0],A[1]). We then proceed to look at each element in the list from 2 to A.length-1 and do the following: if the next element is lower than low, we shift low into sec (it is now the second largest) and set low to this new element. If the new element is less than sec (but not low), we merely overwrite sec with this new element. When the loop terminates, we output sec.

This algorithm runs in Theta(n) time, because only one pass through the list is necessary. It uses Theta(n) space (for the array), but could easily be adapted to run the loop while reading in the input simultaneously. I used an array here because most other assignments will require storing the input, and I wanted some practice with that.

No pseudocode required here, because the code is available to the grader if they'd like to look at it.