Test 3 today
Lab 7 will be due next week

**Sorting Arrays**

Sorting is the process of arranging a list of items in a well-defined order, for example, alphabetic, numeric, ascending or descending.

A sorting algorithm describes how to traverse an array for the purpose of re-ordering its elements.

There are many sorting algorithms; the book demonstrates selection sort.

**Selection sort algorithm:**

outer loop performs a fixed # of passes thru the array = the number of elements - 1

Two pointer variables are maintained, one points to the beginning of the unsorted part of the array, the second to the smallest element found in this pass.

The inner loop makes one pass thru the unsorted part of the array, looking for the smallest value, which is swapped with the element at the beginning of the unsorted part of the array. After each pass, the sorted part of the array grows by 1.

Let’s look at the SelectionSortDemo.java program on the student CD, and add some debug code to trace the process.

**Searching Arrays**

Linear search – visit each element sequentially from the beginning of the array to the end. Assume the array is unsorted, so in no particular order. Compare each element to some search key. If found, stop searching, and return the index where found. If not, return a null value, such as -1. worst case scenario: it is the last value! But on average, we would look thru half the array before finding the search key.

Let’s look at SearchArray.java from the book (p, 499)
Binary search – works only for sorted arrays. “Divide and Conquer”, the way you would search for a name in a phone book. After each comparison, half the array is eliminated for searching. Demo binary search as guessing game – low end/hi end