Exercise 1. Searching and Sorting Arrays

1. Write a program (in C++) to solve the following problem:

    a) Read 10 integers from the standard input and store them in an array.
    b) Find all negative values and replace them by their absolute value.
    c) Display the number of negative values entered by the user.

2. Enhanced Bubble Sort: Make the following simple modifications to improve the performance of the bubble sort:

> After the first pass, the largest number is guaranteed to be in the highest-numbered element of the vector; after the second pass, the two highest numbers are "in place"; and so on. Instead of making nine comparisons (for a 10-element vector) on every pass, modify the bubble sort to make only the eight necessary comparisons on the second pass, seven on the third pass, and so on.

3. Modify the bubble sort to sort the array in decreasing order.

4. A cocktail shaker sort designed by Donald Knuth is a modification of bubble sort in which the direction of bubbling changes in each iteration: In one iteration, the smallest element is bubbled up; in the next, the largest is bubbled down; in the next, the second smallest is bubbled up; and so forth. Implement this new algorithm (in pseudo code) and explore its complexity.

5. The selection sort algorithm sorts an array by repeatedly

- Finding smallest (or equivalently largest) element in the list
- Moving it to the beginning (or end) of the list by swapping it with element in beginning (or end) position

The selection sort algorithm in the lecture slides is not optimal: the inner for loop swap every pair of items (a_j,a_{min}) if a_j <a_{min}. In fact, it only needs to find the minimum value.

1. Modify the selection sort to improve its performance.
2. Modify the selection sort to sort the array in decreasing order.

6. Modify the quicksort algorithm to sort the array in increasing order. In the quicksort process, the pivot is in right. Elements that are smaller than the pivot are in the left part of the array. Elements that are larger than the pivot are in the right part of the array but in the left of the pivot.

7. Insertion sort goes sequentially through the array when making comparisons to find a proper place for an element currently processed. Consider using binary search instead and give a complexity of the resulting insertion sort.