CS 300 Data Structures
Problem Set #11: Elementary Sorting Algorithms

1. Show the steps required to do a *selection sort* on the following array.

	# of compares	# of swaps
n-1	5	1
n-2	4	1
n-3	3	1
	2	1
+1	1	1
+	15 compares	5 swaps

$\approx \frac{N \cdot (N+1)}{2} + N$

$O(N^2)$

0	1	2	3	4	5		
12	10	16	11	9	7	INDEX-OF-LARGEST = 2	LAST=5
12	10	7	11	9	16	INDEX-OF-LARGEST = 0	LAST=4
9	10	7	11	12	16	INDEX-OF-LARGEST = 3	LAST=3
9	10	7	11	12	16	INDEX-OF-LARGEST = 1	LAST=2
9	7	10	11	12	16	INDEX-OF-LARGEST = 9	LAST=1
7	9	10	11	12	16		

```
for (int last=n-1; last>0; last--){
    int indexOfLargest = findLargest(array, last+1)
    swap (array[last], array[indexOfLargest])
}
```

2. How many compares does selection sort make when the input array is already sorted?
   a. constant
   b. logarithmic
   c. linear
   d. quadratic $O(N^2)$
   e. exponential

3. Show the steps required to do an *insertion sort* on the following array.

								#COMPARES	#SWAPS
9	**7**	**6**	**15**	**17**	**5**	**10**	**11**	1	1
7	9	6	15	17	5	10	11	2	2
6	7	9	15	17	5	10	11	1	0
6	7	9	15	17	5	10	11	1	0
6	7	9	15	17	5	10	11	5	4
5	6	7	9	15	17	10	11	3	2
5	6	7	9	10	15	17	11	3	2
5	6	7	9	10	11	15	17		

6 9

5 17

5 15

5 9

5 7

5 6

10 17

10 15

9 10

11 17

11 15

4. Consider the following lists of partially sorted numbers. The double bars represent the sort marker. How many comparisons and swaps are needed to sort the next number [Insertion Sort].

[1 3 4 8 9 || 5 2] 1 COMPARE 1 SWAP

1 3 4 8 5 9 2 1 COMPARE 1 SWAP

1 3 4 5 8 9 2 1 COMPARE 0 SWAP

3 COMPARISONS
2 SWAPS

5. Consider the following lists of partially sorted numbers. The double bars represent the sort marker. How many comparisons and swaps are needed to sort the next number [InsertionSort].

   [**1 3 4 5 8 9 || 2**]

   6 COMPARES
   5 SWAPS

6. How many compares does insertion sort make on an input array that is already sorted
   a. constant
   b. logarithmic
   c. linear O(N)
   d. quadratic
   e. exponential