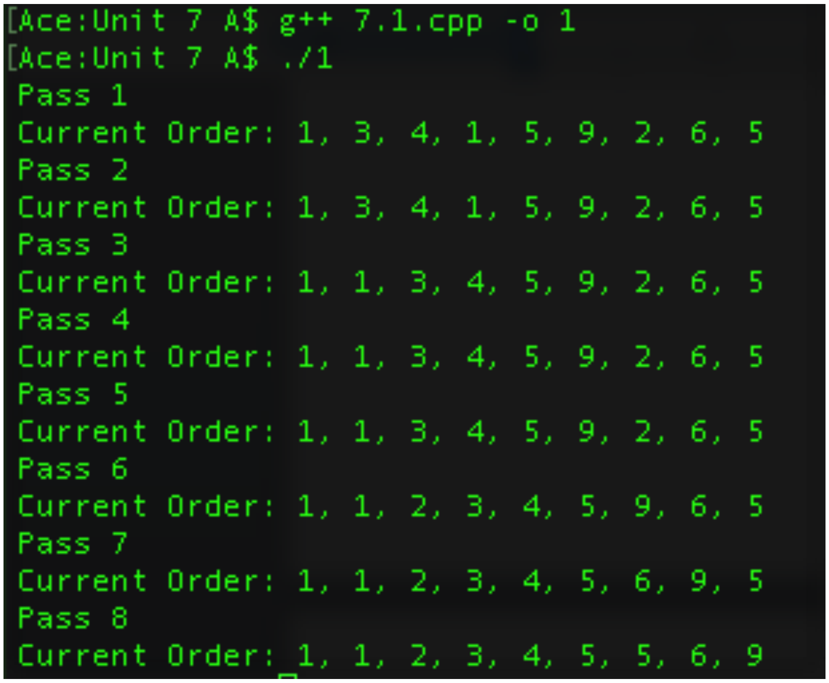
Aaron Williams

Assignment 5 (Homework Unit 7)

01 October 2017

7.1
Sort the sequence 3, 1, 4, 1, 5, 9, 2, 6, 5 using insertion sort.

-Conducted using 8 passes (N-1) starting at the first element (p = 1). The program was designed to follow the insertion sort algorithm by analyzing each element in the array and moving it left if the value is less than previous elements.



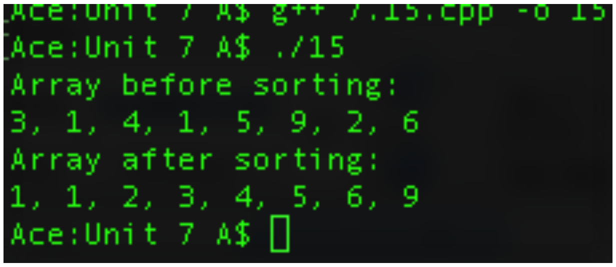
```
[Ace:Unit 7 A$ g++ 7.1.cpp -o 1
[Ace:Unit 7 A$ ./1
 Pass 1
 Current Order: 1, 3, 4, 1, 5, 9, 2, 6, 5
 Pass 2
 Current Order: 1, 3, 4, 1, 5, 9, 2, 6, 5
 Pass 3
 Current Order: 1, 1, 3, 4, 5, 9, 2, 6, 5
 Pass 4
 Current Order: 1, 1, 3, 4, 5, 9, 2, 6, 5
 Pass 5
 Current Order: 1, 1, 3, 4, 5, 9, 2, 6, 5
 Pass 6
 Current Order: 1, 1, 2, 3, 4, 5, 9, 6, 5
 Pass 7
 Current Order: 1, 1, 2, 3, 4, 5, 6, 9, 5
 Pass 8
 Current Order: 1, 1, 2, 3, 4, 5, 5, 6, 9
```

7.2
What is the running time of insertion sort if all elements are equal?

-If all elements were equal, the run time would be linear (O(n)). This would be the best case scenario for insertion sort run time, and would occur only because 0 positions would be moved.

7.15
Sort 3, 1, 4, 1, 5, 9, 2, 6 using mergesort.

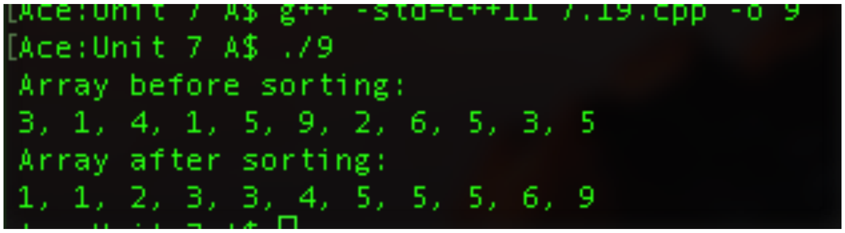
-Conducted using the concept of divide and conquer. The program was designed to follow the mergesort algorithm by splitting the initial array into smaller sub-arrays, and then sorting those sub-arrays. Once the sub-arrays are sorted, they are then compared and re-inserted into the original array in ascending order.



```
Ace:Unit 7 A$ g++ 7.15.cpp -o 15
[Ace:Unit 7 A$ ./15
Array before sorting:
3, 1, 4, 1, 5, 9, 2, 6
Array after sorting:
1, 1, 2, 3, 4, 5, 6, 9
Ace:Unit 7 A$ 
```

7.19
Sort 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5 using quicksort with median-of-three partitioning and a cutoff of 3.

-Conducted using a median-of-three partitioning algorithm that calculated the median of the first element, middle element, and last element, which was used to determine the pivot. A cutoff of 3 was used per the problem statement, which was probably not optimal given the size of the array. The quicksort algorithm was used in conjunction with insertion sort to handle smaller sub-arrays.



```
[Ace:Unit 7 A$ g++ -std=c++11 7.19.cpp -o 9
[Ace:Unit 7 A$ ./9
 Array before sorting:
 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5
 Array after sorting:
 1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9
```