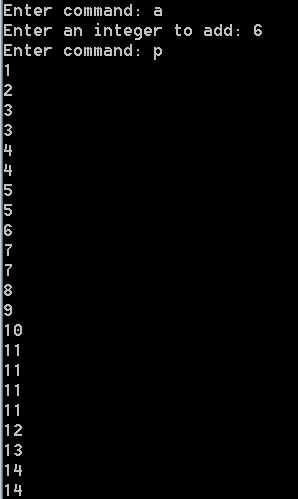
Kyle Cooper

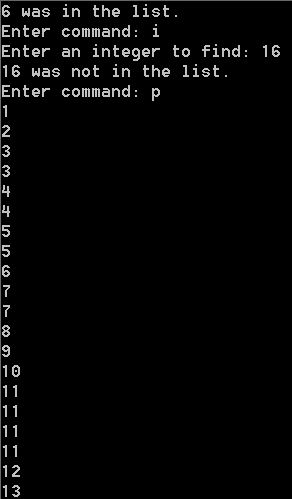
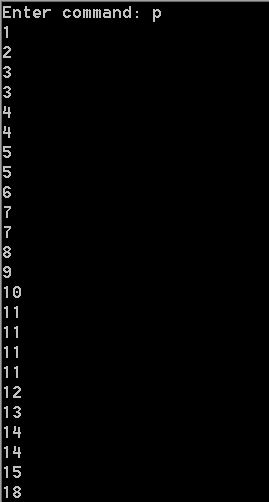
HW4

The two pictures below, show the before and after of adding '6' to the sorted link list.

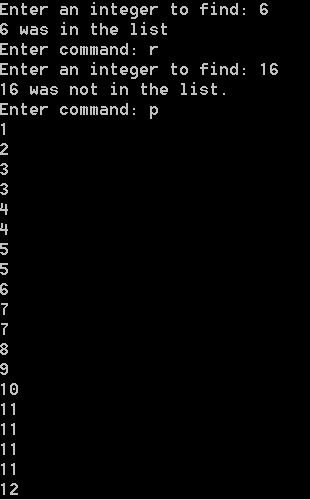
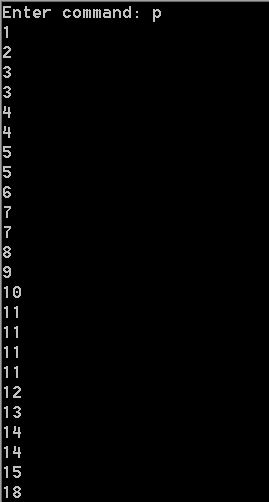
Before After

Below is a successful(6) and non-successful(16) search using an iterative function.

6 found, 16 not no 16

Below is a successful(6) and non-successful(16) search using an recursive function.

6 found, 16 not no 16

The time complexity for part a is O(n), the reason for this is since the list is already sorted and my linklist is linear, finding were to put a new number takes at least 1 iteration if it's placed at the start, and a max of n if placed at the end. Which means the time complexity is O(n).

For part b the time complexity is O(n), the reason for this is because if the number we are searching for is the first number in the list, it will only take one iteration, up to a max of n if the number is at the end of the list, meaning the time complexity is O(n).

For part c the time complexity is O(n)

T(n) = 1+T(n-1), T(n-1) = 1+1+T(n-2), T(0)=1

T(n-n) = n + T(0)=n+1 = O(n)

This can also be seen by the fact that since the list is sorted, and the recursive function just moves the node = node->nextnode before calling itself again and then checking node->num for the num. This means if num is at the start of the list, the recursive function will end with only 1 call, but if num is at the end of the list, the recursive function will make n calls.