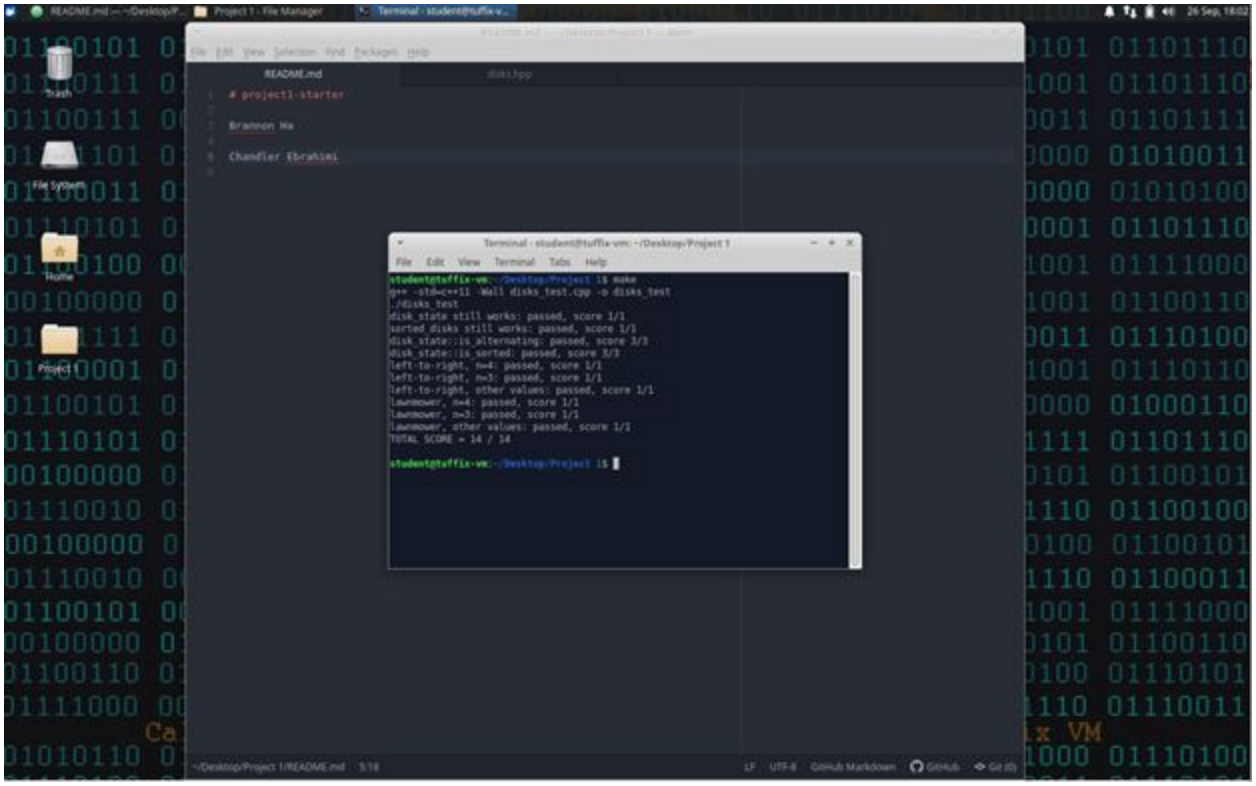
Project 1

Chandler Ebrahimi csebra@csu.fullerton.edu

Brannon Ha brannonha@csu.fullerton.edu



Left-to-Right Algorithm Pseudocode:

```
def sort_left_to_right(L)

Set end equal to Total Disks

Swaps = 0

Set local to L

For each i between 0 and end

        For each j between i and end

                If the current j index is a light disk and j+1 index is a dark disk

                        Swap

                        swaps++

Return local
```

Lawnmower Algorithm Pseudocode:

```
def lawnmower(L)

Set end equal to Total Disks

Swaps = 0

Sorted = false

Set local to L

While sorted = false

                For each disk in L

                    Compare local[i] to local[i+1]

                    If out of order swap

                    Swaps++

                For Max_disk_count to Zero

                    Compare local[i] to local[i+1]

                    If out of order swap

                    Swaps++

                    Check if sorted

                    If sorted set sorted = true

Return local
```

Time Complexity of Algorithms:

The Left to Right algorithm has a its total step count to be equal to this equation: $1+1+1+(n*n*3) + 1 = 3n^2 + 3$ which can be simplified down to n^2. Therefore the Left to Right algorithm falls under $O(n^2)$ complexity.

The Lawnmower algorithm has a total step count of: 1+1+1+(n-1)+1+1+(n-1)+1+1+1+1=n+7 which can be simplified to n. Therefore the Lawnmower algorithm falls under tO(n) complexity.