

Group Members:

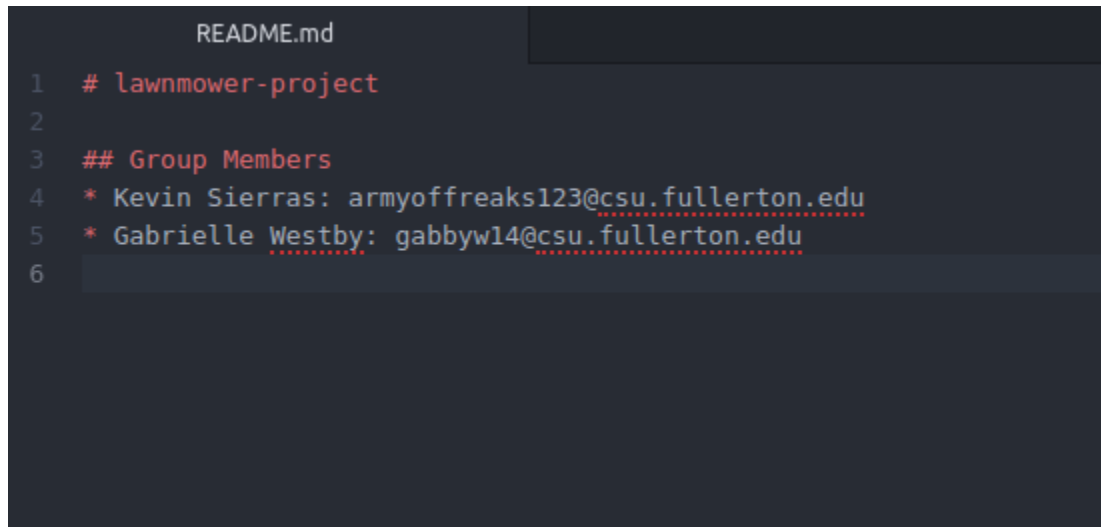
- Kevin Sierras: armyoffreaks123@csu.fullerton.edu
- Gabrielle Westby: gabbyw94@csu.fullerton.edu

Github Link:

- <https://github.com/kevsieras/lawnmower-project>

Screenshots:

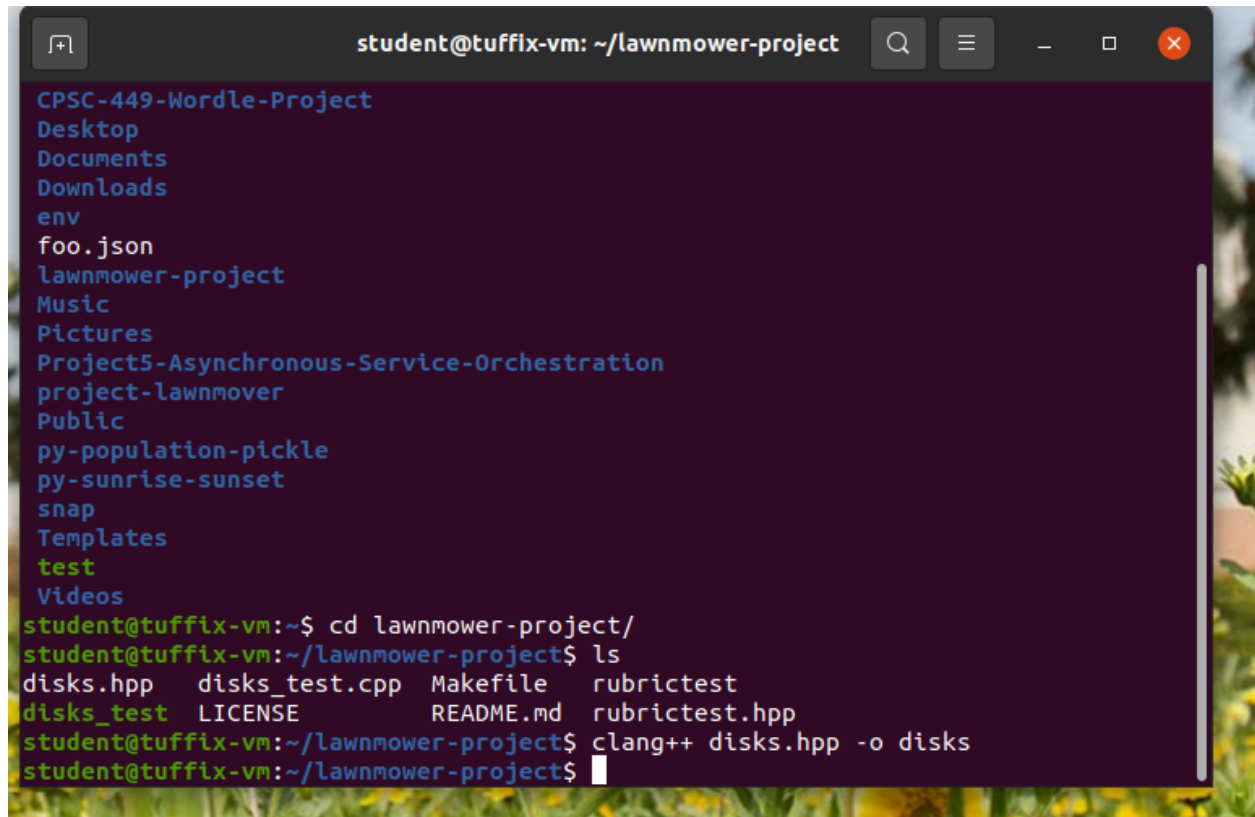
Group Names

A screenshot of a code editor showing a README.md file. The file content is as follows:

```
1 # lawnmower-project
2
3 ## Group Members
4 * Kevin Sierras: armyoffreaks123@csu.fullerton.edu
5 * Gabrielle Westby: gabbyw14@csu.fullerton.edu
6
```

The email addresses in the original image were incorrectly transcribed as 'gabbyw94' and 'gabbyw14'. The screenshot shows the correct email address 'gabbyw14'.

Compiling

A terminal window titled "student@tuffix-vm: ~/lawnmower-project" with standard window controls. The terminal shows a directory listing of the home directory, followed by a command to change into the "lawnmower-project" directory. Inside this directory, another listing shows source files and a Makefile. Finally, the "clang++" command is used to compile "disks.hpp" into an executable named "disks".

```
student@tuffix-vm: ~/lawnmower-project
CPS-449-Wordle-Project
Desktop
Documents
Downloads
env
foo.json
lawnmower-project
Music
Pictures
Project5-Asynchronous-Service-Orchestration
project-lawnmover
Public
py-population-pickle
py-sunrise-sunset
snap
Templates
test
Videos
student@tuffix-vm:~$ cd lawnmower-project/
student@tuffix-vm:~/lawnmower-project$ ls
disks.hpp  disks_test.cpp  Makefile  rubrictest
disks_test LICENSE      README.md  rubrictest.hpp
student@tuffix-vm:~/lawnmower-project$ clang++ disks.hpp -o disks
student@tuffix-vm:~/lawnmower-project$
```

Running

```
student@tuffix-vm: ~/lawnmower-project
student@tuffix-vm:~/lawnmower-project$ ./disks_test
disk_state still works: passed, score 1/1
sorted_disks still works: passed, score 1/1
disk_state::is_initialized: passed, score 3/3
disk_state::is_sorted:
    TEST FAILED:
    line 81 of file disks_test.cpp, message: is_sorted() for n=3
    score 0/3
alternate, n=4: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 11 / 14

student@tuffix-vm:~/lawnmower-project$
```

(We don't know why n=3 is failing, every attempt to fix it does not work.)

Pseudocode and Step Count

Sort Alternate

```
set number of swaps to 0
set disk state to keep track of state before swap
for each column of light count do
    set index counter to 0
    while index + 1 is less than the total count do
        if the state at index is greater than state at index + 1 then
            swap states at index
            increase number of swaps
        end if
    end while
    increase index counter
end for
return to sorted disks with new state and number of swaps
```

Sort Alternate Step Count

```
sorted_disks sort_alternate(const disk_state& before) {
```

```
    int numOfSwap = 0; 1 + n
```

```
    disk_state state = before; 1 + n
```

```
    for(int i = 0; i < state.light_count(); ++i) { n - 0 + 1 = n
```

```
        int index = 0; 1 + n
```

```
        while(index + 1 < state.total_count() - i - 1) {
```

```
            if(state.get(index) > state.get(index + 1)) {
```

```
                state.swap(index); 1 + n
```

```
                ++numOfSwap; 1 + n
```

```
            } 1 + n
```

```
            ++index; 1 + n
```

```
        }
```

```
    }
```

```
    return sorted_disks(disk_state(state), numOfSwap);
```

```
}
```

$$\sum_{i=0}^n \sum_{j=1}^{\log_2 i} 2$$

$$\log_2 i = \frac{\log i}{\log 2}$$

$$\frac{1}{\log 2} \cdot 2 \sum_{i=1}^n \log i = 2 + 2 \log(n)$$

$$S.C = 2(1 + n \log(n))$$

Sort Alternate Proofs

Proof #1

$$2 \cdot n \leq C \cdot n \log(n)$$

$$C = 2$$

$$2 \cdot n \leq 2(n \log(n))$$

$$0 \leq 0 \quad \checkmark$$

Proof #2

$$2 \cdot n \in n \log(n)$$

$$\lim_{n \rightarrow \infty} \frac{2 \cdot n}{n \log(n)} \rightarrow \frac{n}{n \log(n)} \quad \frac{n}{n \log(n)}$$

$L \geq 0$, thus by definition $2 \cdot n \in n \log(n)$

Sort Lawnmower

```

set number of swaps to 0
set disk state to keep track of state before swap
for each column of total count / 2 do
    set index counter to 0
    while index + 1 is less than total count do
        if the state at current index does not equal the state at index + 1 then
            if the state at current index equals a dark disk & the state at index + 1
                equals a light disk then
                    swap states at index
                    increase number of swaps
            end if
        end if
        increase index
    end while

    while the index is greater than zero do
        if the state at index - 1 does not equal the state at index then
            if state at index - 1 equals a dark disk & state at index equals a light disk
                then
                    swap states at index - 1
                    increase number of swaps
            end if
        end if
        decrease index
    end while
end for
return to sorted disks with new state and number of swaps

```

Sort Lawnmower Step Count

Sort Lawnmower Proof

[illegible]