Problem 1

1. *Someone plops you into the middle of this directory system. What command will allow you to know what your absolute path is?*

```
$ pwd
```

2. *Your terminal tells you that ˜/music/ is your absolute path. What command will allow you to view all of the directories and files inside of ˜/music/? What files should be listed?*

$ ls for key repositories, **$ ls -a** for everything. All files and repositories inside the repository “music” will be listed.

3. *What command(s) will allow you to open up patterns.txt inside of ˜/music/ and edit the file?*

```
$ nano patterns.txt
```

4. *What command will allow you to print the file on your terminal instead of opening it up in a text editor?*

```
$ cat patterns.txt
```

5. *Bonus: What command will allow you to see the difference between patterns.txt and symmetry.txt?*

```
$ diff patterns.txt symmetry.txt
```

6. *What command will allow you to move patterns.txt to ˜/math/calculus/?*

```
$ mv ~/music/patterns.txt ~/math/calculus/patterns.txt
```

7. *What command will allow you to remove triangle.txt from˜/math/geometry?*

```
$ rm triangle.txt
```

8. *The file patterns.txt is now only located at˜/math/calculus/patterns.txt. What singular command will allow you (the user) to move from ˜/music/ to ˜/math/? (Hint: What is your relative path to ˜/math/?)*

```
$ cd ../math
```

9. *Your absolute path is now ˜/math/. What command would make ~/math/ a local git repository?*

```
$ git init
```

10. *The math/ directory is now a local git repository. How would you connect your local math/ repository with a remote math/ repository?*

```
$ git remote add origin [ssh url]
```

11. *Your local math/ repository is now connected to a remote one. What command will allow you to copy symmetry.txt from ˜/music/ to your local math/ repository?*

```
$ cp symmetry.txt ../math
```

12. *What command will allow you to view the current state of your working directory and staging area? Is git currently tracking symmetry.txt, why or why not?*

```
$ git status
```

“symmetry.txt” is not currently being tracked, because it was just added to the local repository and we did not run “$ git add” after its addition.

13. *What command adds only the copy of symmetry.txt to your math/ staging area? What command adds all changes to all files and directories to your staging area?*

```
$ git add symmetry.txt
$ git add .
```

14. *The file symmetry.txt is now in your staging area. Is git currently tracking symmetry.txt? Bonus: What command would allow you to unstage symmetry.txt?*

"symmetry.txt" is now being tracked.
$ restore --staged symmetry.txt

15. *What command will allow you to commit your staged files to your local repository? Include the message.*

$ git commit -m "Committed"

16. *Your staged files have now been committed to your local repository. What command will allow you to push your local repository to your remote repository. What branch should you be pushing to? Why?*

$ git push origin main
"origin" is a nickname for the repo we see when we run "$ git remote -v" and this is the branch we want the file to go to

17. *What command will allow you to make a new branch off of math/ called applied math?*

$ mkdir "applied math"

18. *What command will allow you to move into the applied math branch?*

$ cd "applied math"

19. *You are now located inside of the applied math branch. You make a new directory called physics/ and inside of that directory you make a file called mechanics.txt with some fun facts in it. How would you push all of these changes to your remote repository? What would happen if you switched back to your original branch before committing these changes?*

```
$ cd ../..
$ git add .
$ git commit -m “Committed”
$ git push origin main
```

I would switch back to the original branch before committing so I can commit everything at once instead of doing them one at a time.

20. *What command will list all of your branches? What branch will be currently highlighted in green?*

```
$ git branch -a
```

The highlighted branch is the one I am on.

21. *What command will allow you to move back to your original branch?*

```
$ cd .. (but I’m already in “math” as of question 19)
```

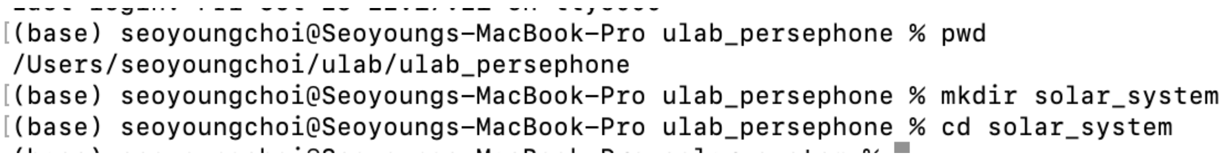
22. *Bonus: What command will allow you to combine applied math with your original branch?*

```
$ rsync -avh “applied math”/* math/ (?)
```

23. *Make a tree diagram of your directory system (with all the files and folders) after working through this problem. Specify which directories are repositories and what the branch of a repository will look like. Take a screenshot or a photo (if you drew it by hand) and save it. You will need it for later.*

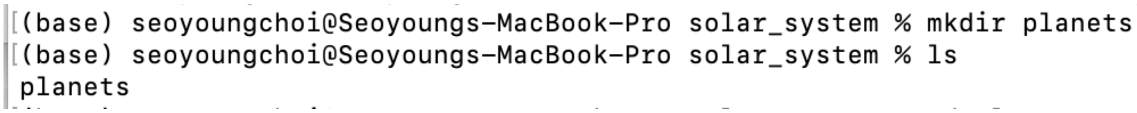
---

1. Inside of ~/ulab/ulab_yourname make a new branch called solar_system. Move into the solar_system branch. Take a screenshot of making and moving into the branch.



```
[(base) seoyoungchoi@Seoyoungs-MacBook-Pro ulab_persephone % pwd
/Users/seoyoungchoi/ulab/ulab_persephone
[(base) seoyoungchoi@Seoyoungs-MacBook-Pro ulab_persephone % mkdir solar_system
[(base) seoyoungchoi@Seoyoungs-MacBook-Pro ulab_persephone % cd solar_system
```

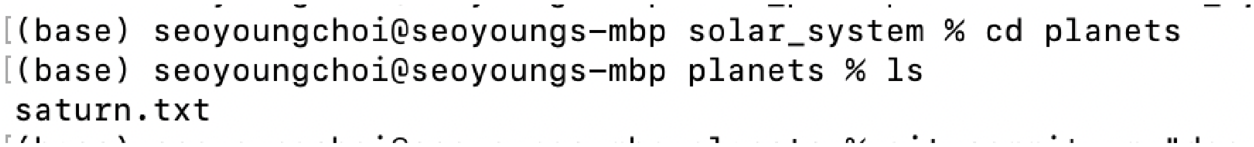
2. Inside of solar_system make a directory called planets. Remove everything inside of this branch. CAREFULLY!



```
[(base) seoyoungchoi@Seoyoungs-MacBook-Pro solar_system % mkdir planets
[(base) seoyoungchoi@Seoyoungs-MacBook-Pro solar_system % ls
planets
```

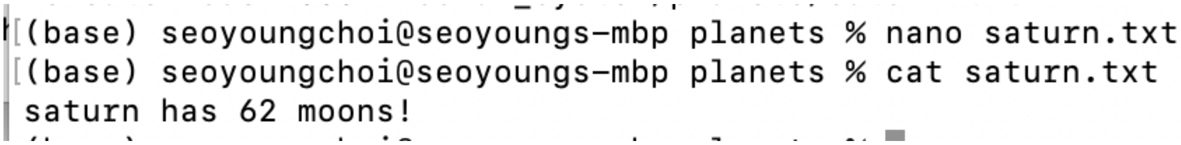
(I don't know the command to "remove everything inside of this branch" so I was afraid to try)

3. Inside of planets make a file named after your favorite planet in our solar_system. Example: ~/ulab/ulab_yourname/earth.txt.



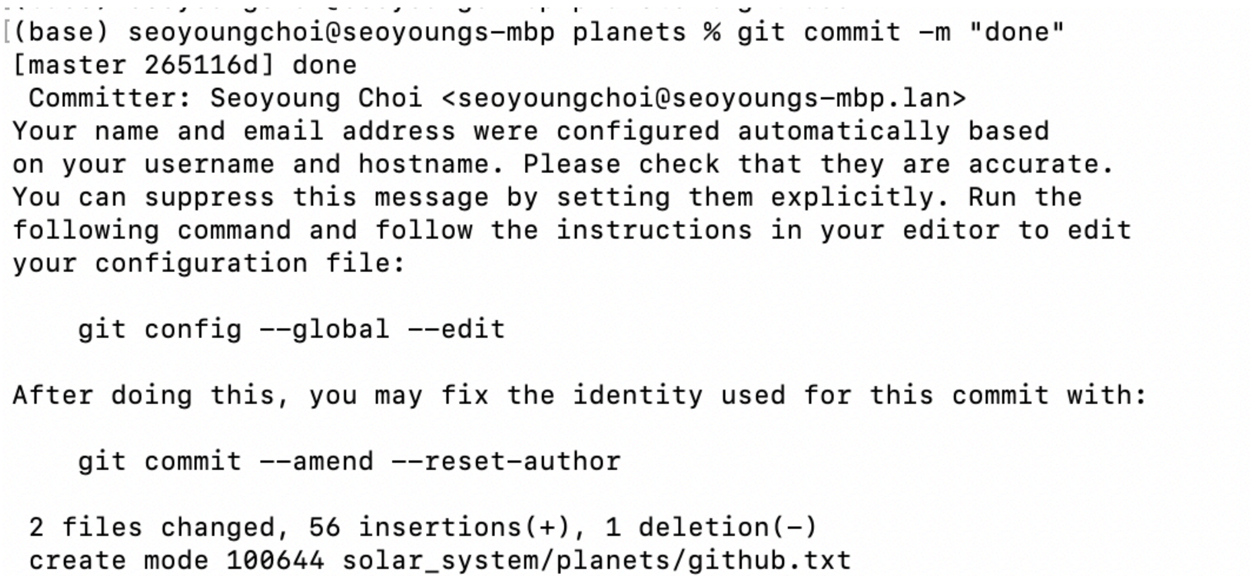
```
[(base) seoyoungchoi@seoyoungs-mbp solar_system % cd planets
[(base) seoyoungchoi@seoyoungs-mbp planets % ls
saturn.txt
```

4. Look up a fact about your favorite planet and add at least one sentence to your .txt file you created last step.



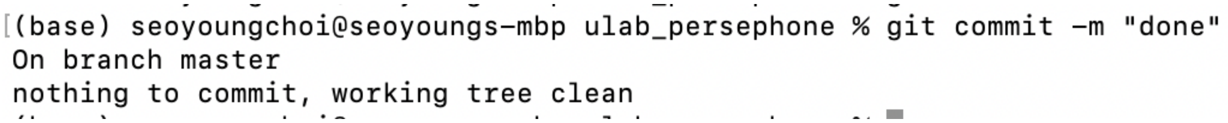
```
[(base) seoyoungchoi@seoyoungs-mbp planets % nano saturn.txt
[(base) seoyoungchoi@seoyoungs-mbp planets % cat saturn.txt
saturn has 62 moons!
```

5. Inside of your solar_system branch make another file called github.txt that contains all your answers to Problem 1. Take a screenshot of committing github.txt and solar_system to your remote repository.

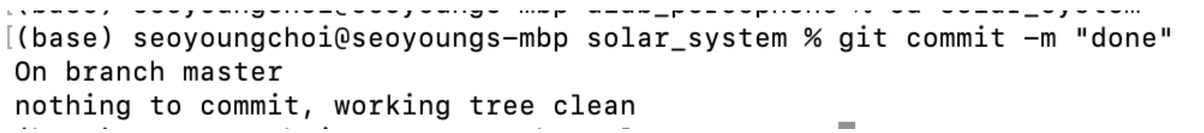


```
[(base) seoyoungchoi@seoyoungs-mbp planets % git commit -m "done"
[master 265116d] done
 Committer: Seoyoung Choi <seoyoungchoi@seoyoungs-mbp.lan>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 56 insertions(+), 1 deletion(-)
 create mode 100644 solar_system/planets/github.txt
```

(I already committed solar_system)

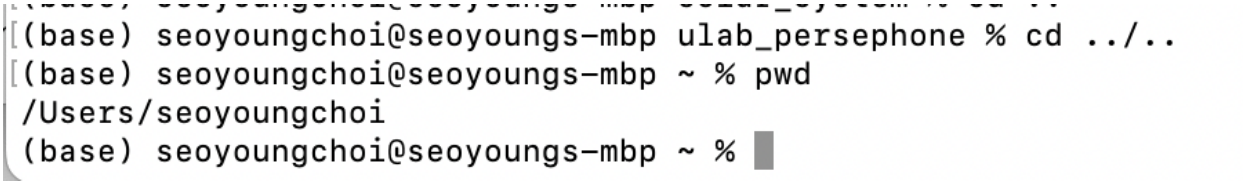


```
[(base) seoyoungchoi@seoyoungs-mbp ulab_persephone % git commit -m "done"
On branch master
nothing to commit, working tree clean
```



```
[(base) seoyoungchoi@seoyoungs-mbp solar_system % git commit -m "done"
On branch master
nothing to commit, working tree clean
```

6. Switch back to your main branch. Take a screenshot of switching back into your main branch after commiting solar system to your remote repository.



```
[(base) seoyoungchoi@seoyoungs-mbp ulab_persephone % cd ../..
[(base) seoyoungchoi@seoyoungs-mbp ~ % pwd
/Users/seoyoungchoi
(base) seoyoungchoi@seoyoungs-mbp ~ %
```