LAB 3

GIT AND GITHUB

Full name: Nguyen Minh Nhut

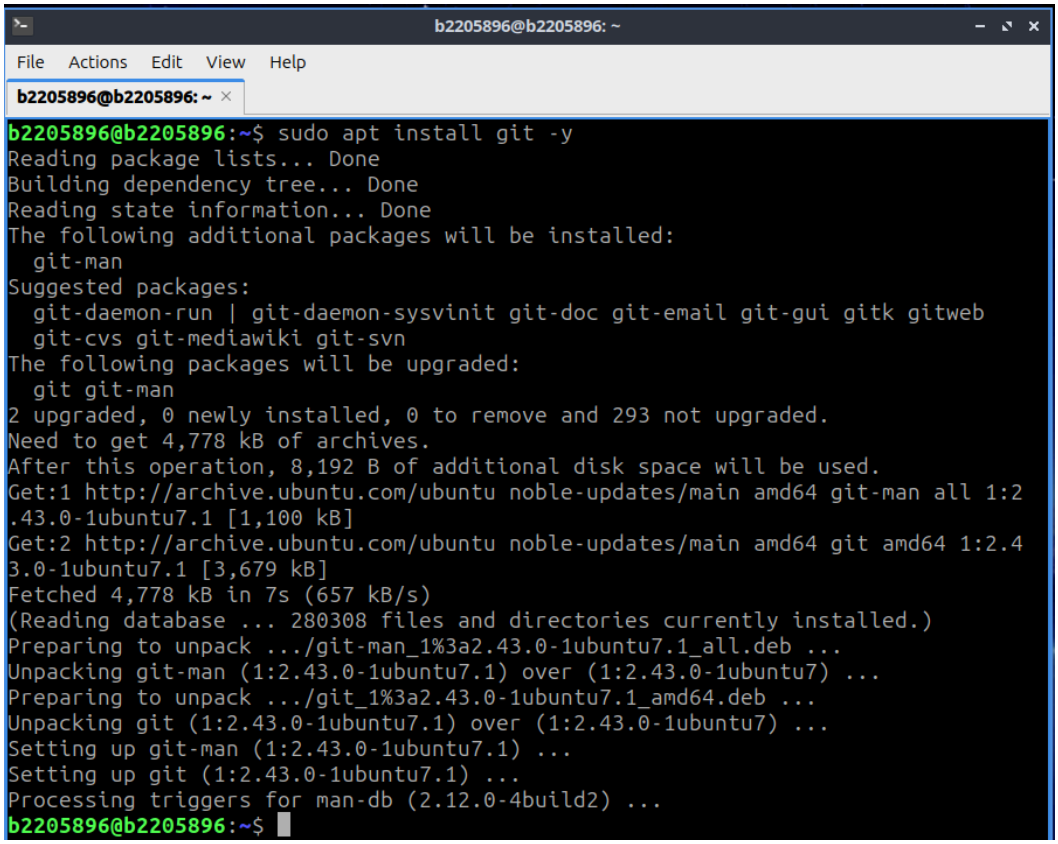
Student ID: B2205896

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

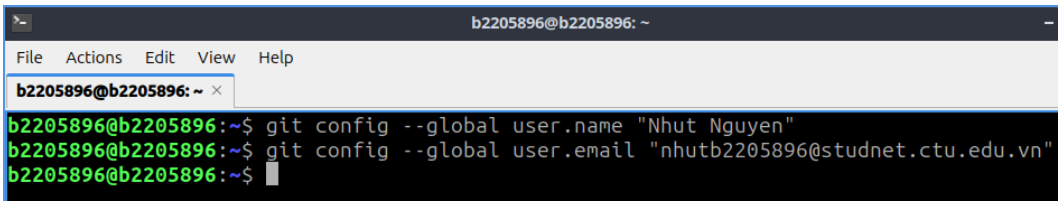
1.Setup and making commits

- Install and setup git on your computer (remember to set your name/email)

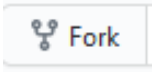
```
$sudo apt update ; sudo apt install git -y
```




```
$git config --global user.name "Firstname Lastname"
$git config --global user.email "example@.ctu.edu.vn"
```

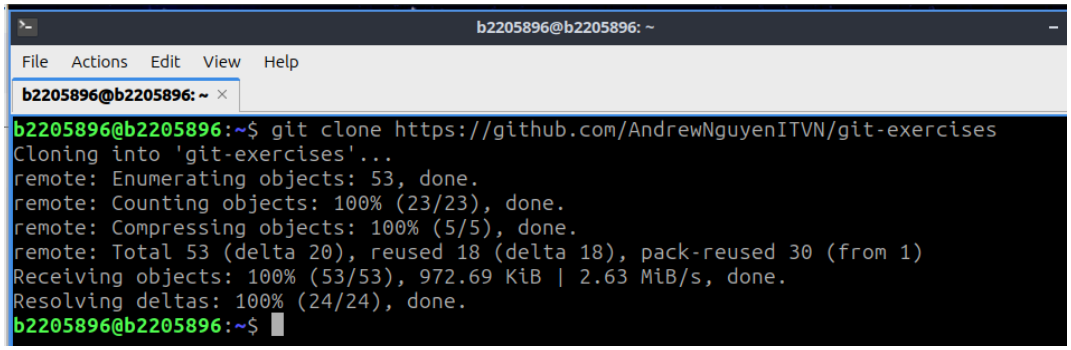


- Create an account (or login) to GitHub at https://github.com

- Fork (

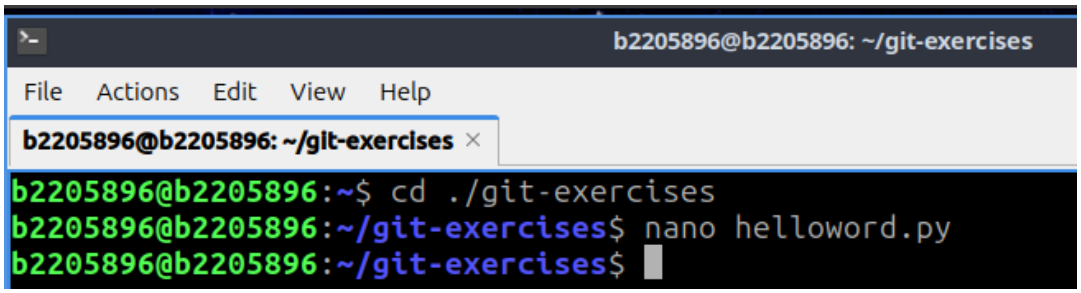
)this repository https://github.com/TuanThai/git-exercises.git to your Github account. https://github.com/AndrewNguyenITVN/git-exercises
- Clone the forked repository to your computer

```
$git clone <url of your forked repository>
```

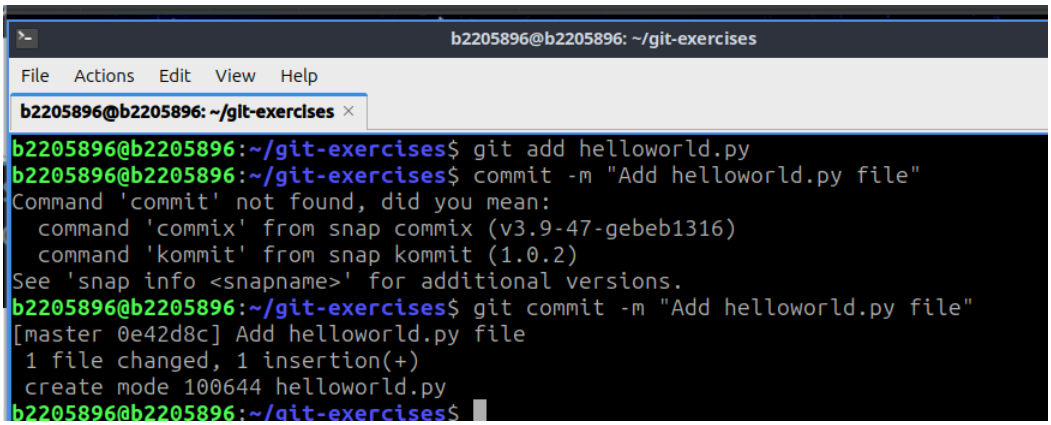


- Create and add a new file `helloword.py`

```
$cd ./git-exercises
$nano helloword.py
      #!/usr/bin/python3
      print("Hello world")
```

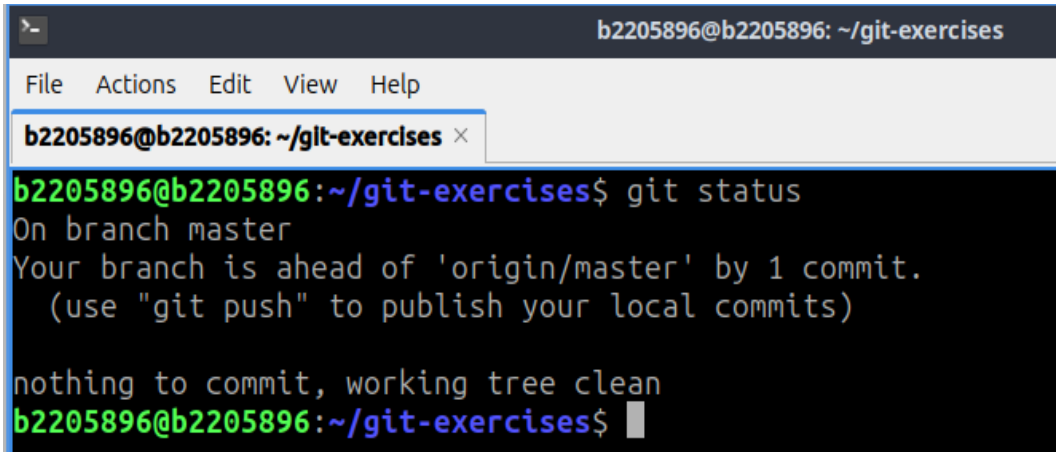



```
$git add helloworld.py
$git commit -m "Add hellworld.py file"
```



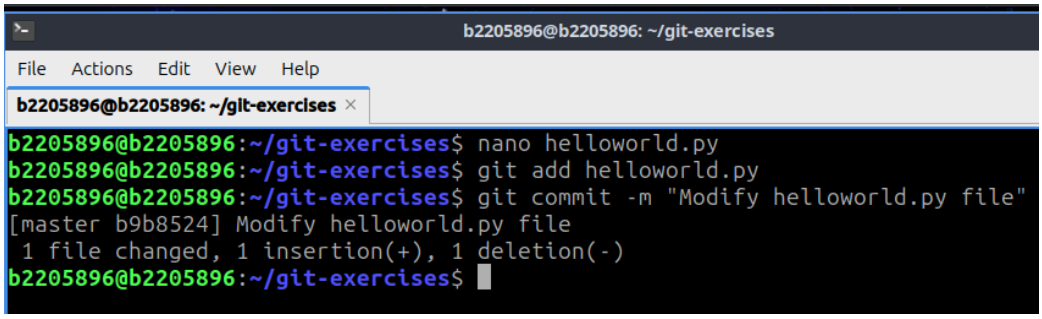
- Examine the state of your repo

```
$git status
```



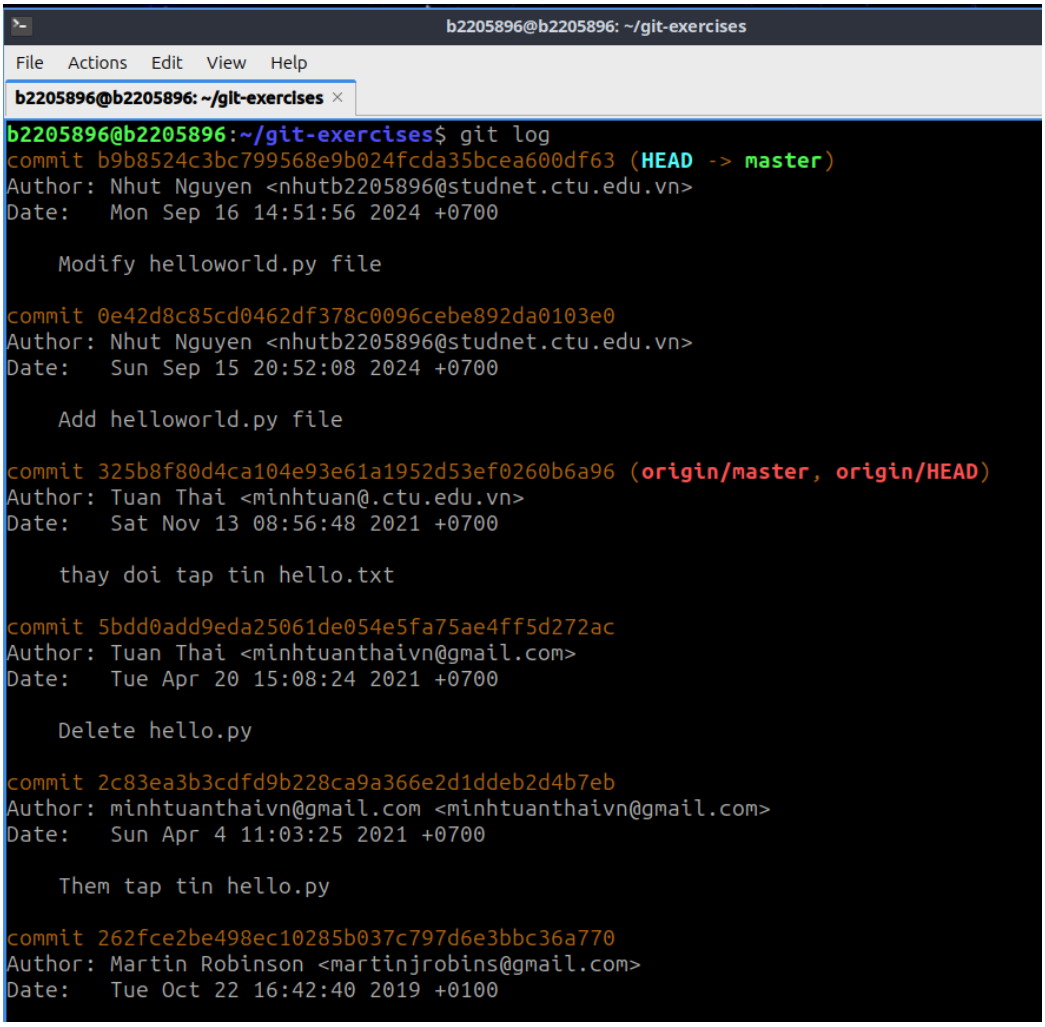
- Edit and save your new file, then add it to the staging area. Finally make a new commit with the edited file. At all stages use git status to see how your repository changes

```
$nano helloworld.py
      #!/usr/bin/python3
      print("Hello world")
      print("Hello Git")
$git add helloworld.py
$git commit -m "Modify hellworld.py file"
```



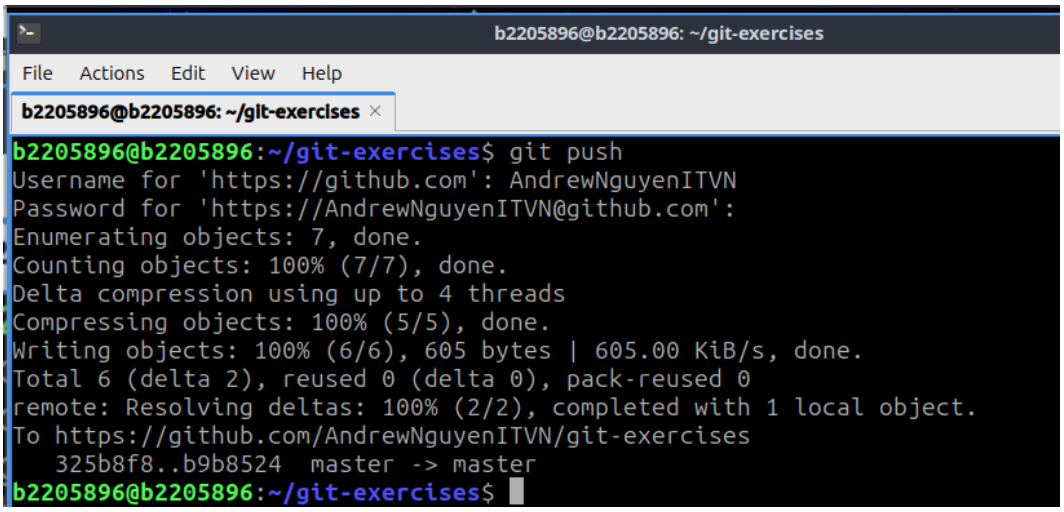
- View the log

```
$git log
```



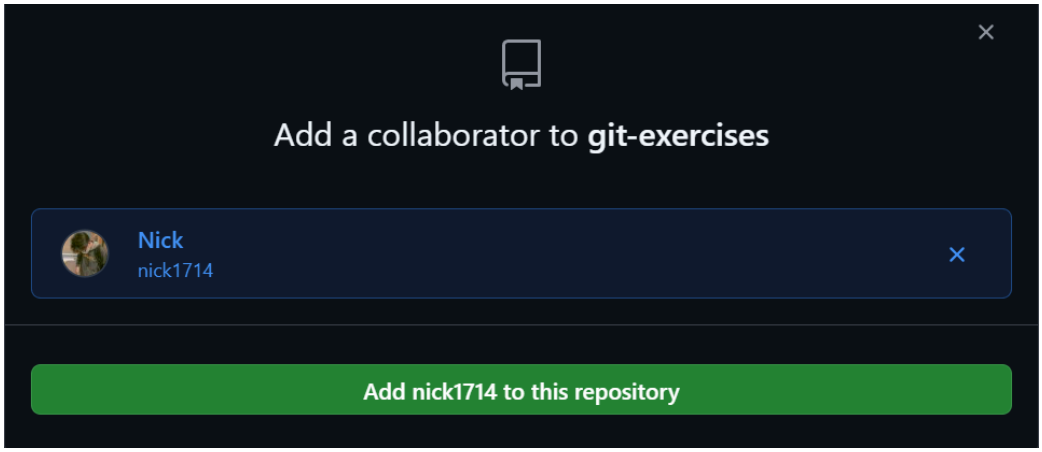
- Push the commits to the server

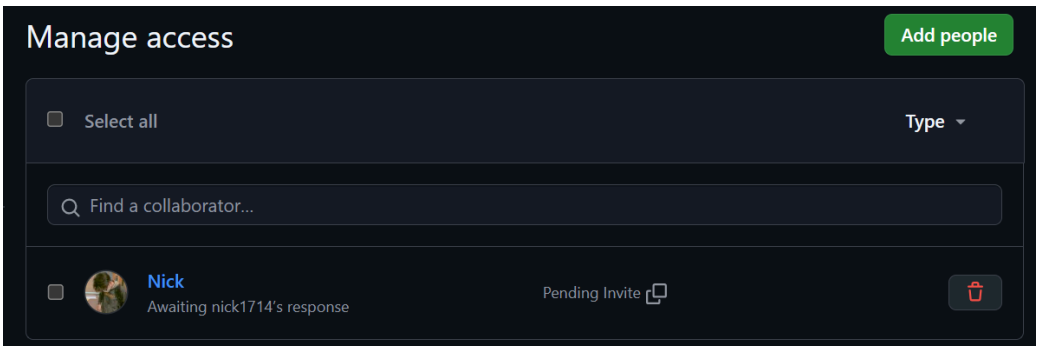
```
$git push
```



(Note: You need to provide a personal access token as a password, please follow this article to create one)

- Invite someone to be a collaborator of Github repository



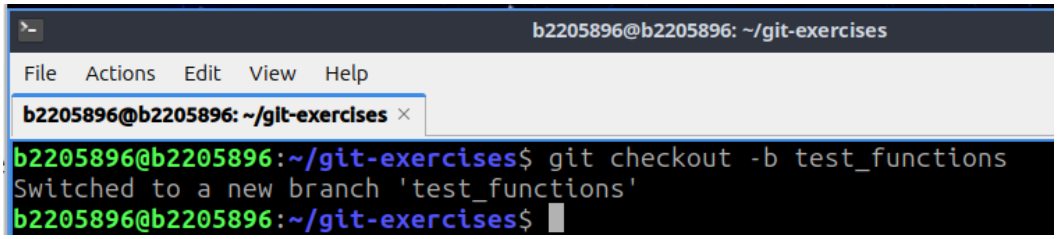


(take screenshots to show that you finish this exercise)

2.Branching and Merging

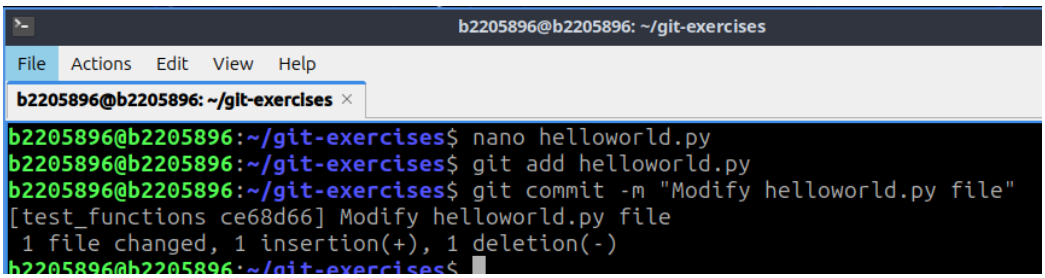
- Create a new branch “test_functions”

```
$git checkout -b test_functions
```



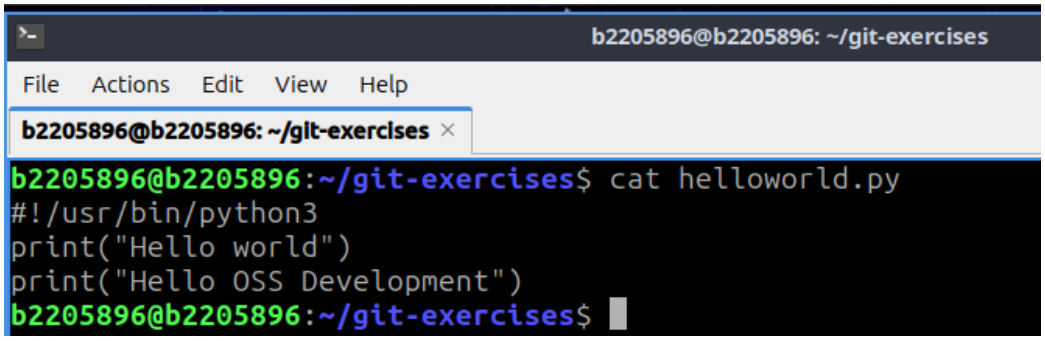
- Edit `helloworld.py` file and commit the result

```
$nano helloworld.py
      #!/usr/bin/python3
      print("Hello world")
      print("Hello OSS Development")
$git add helloworld.py
$git commit -m "Modify hellworld.py file"
```



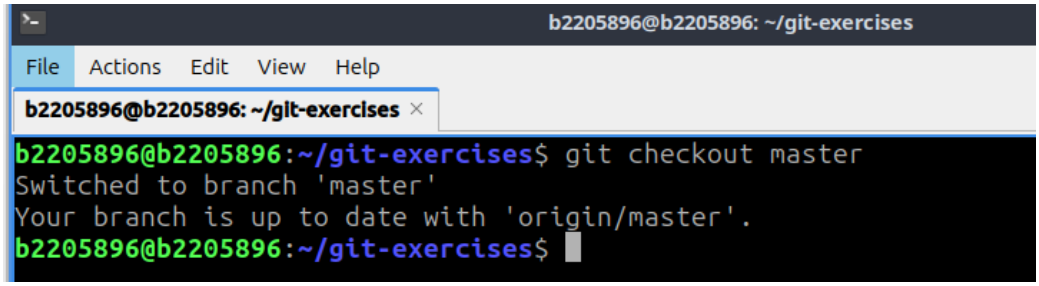
- Display `helloworld.py` content

```
$cat helloworld.py
```



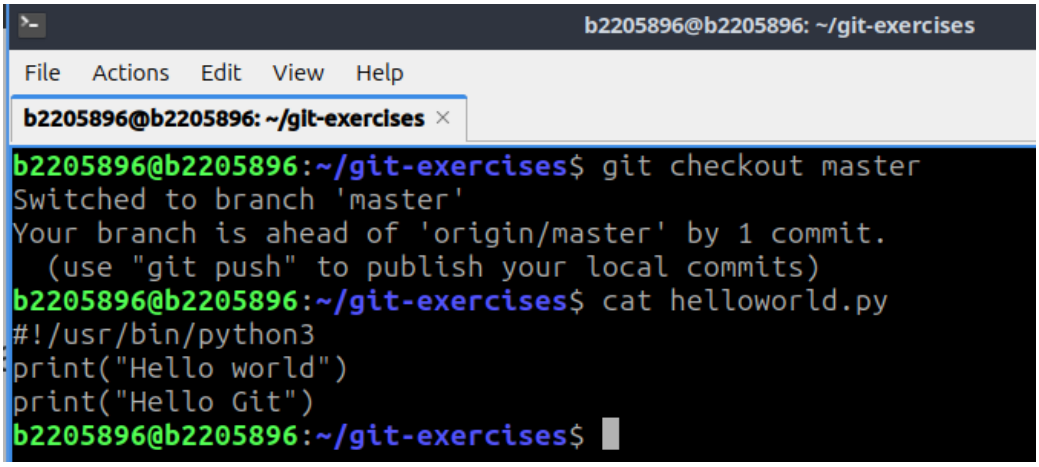
- Swap back to the "master" branch

```
$git checkout master
```



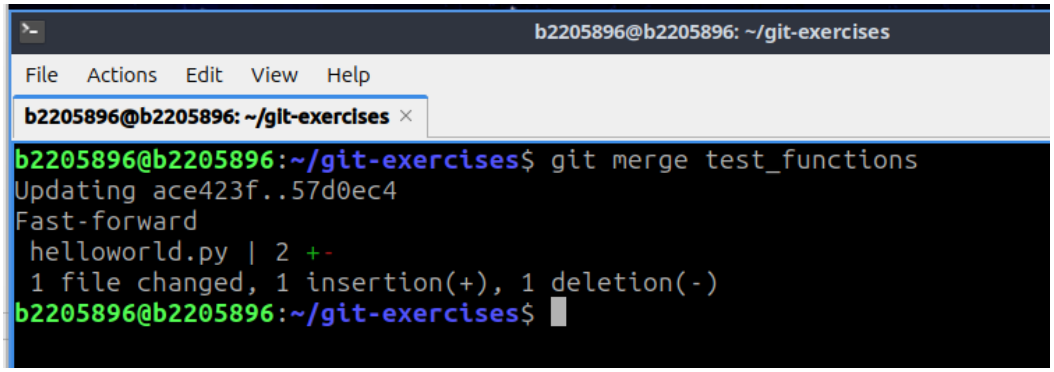
- Display `helloworld.py` content

```
$cat helloworld.py
```



- Merge “test_functions” branch to “master”

```
$git merge test_functions
```



- Display `helloworld.py` content. Are there any conflicts?

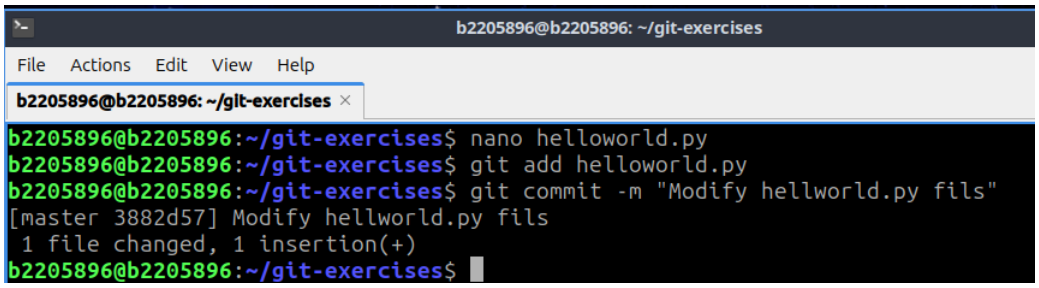
```
$cat helloworld.py
```

- Edit `helloworld.py` file and commit the result

```
$nano helloworld.py
```

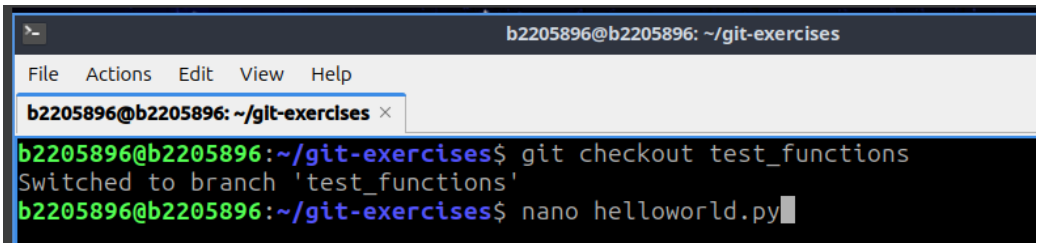
```
#!/usr/bin/python3
print("Hello world")
print("Hello OSS Development")
print("Hello CTU")
```

```
$git add helloworld.py
$git commit -m "Modify hellworld.py file"
```

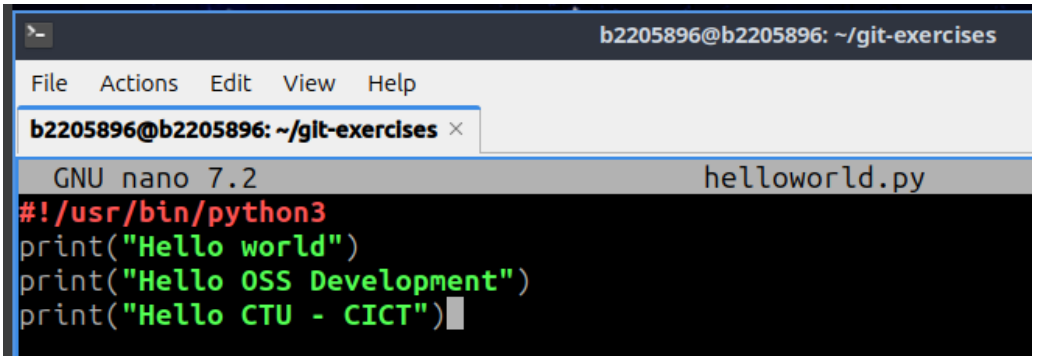


- Checkout to `test_functions`, edit `helloworld.py` file and commit the result

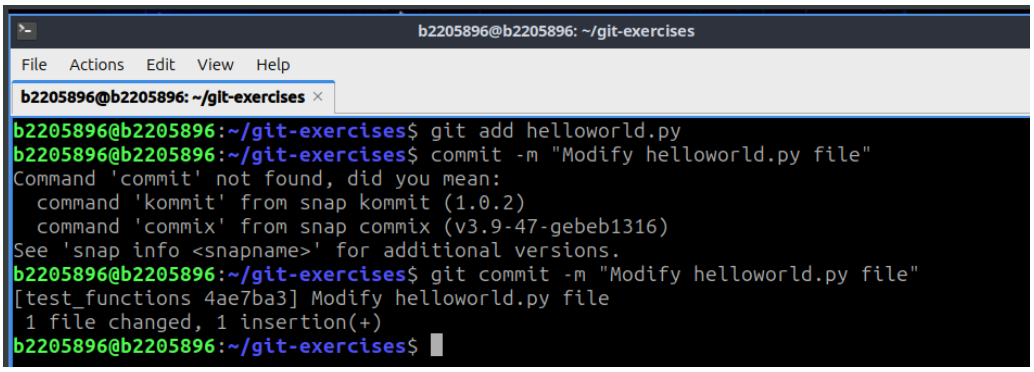
```
$git checkout test_functions
```




```
$nano helloworld.py
#!/usr/bin/python3
print("Hello world")
print("Hello OSS Development")
print("Hello CTU - CICT")
```

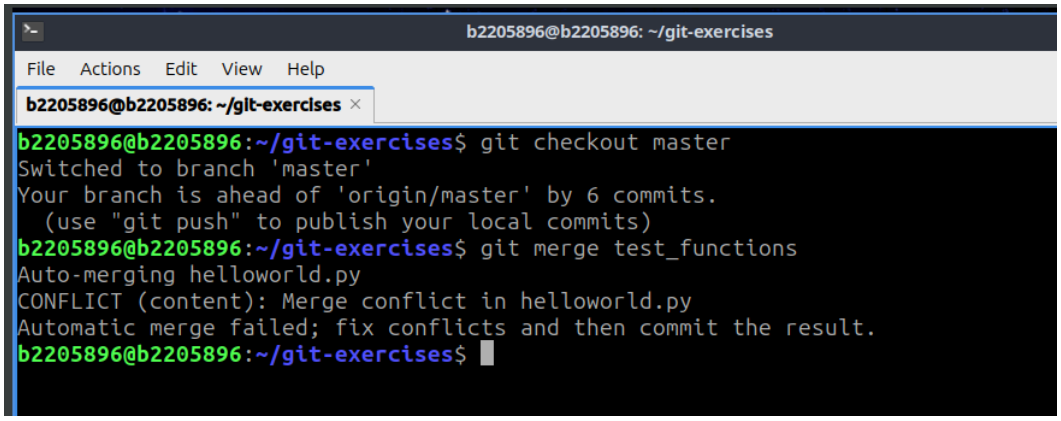



```
$git add helloworld.py
$git commit -m "Modify hellworld.py file"
```

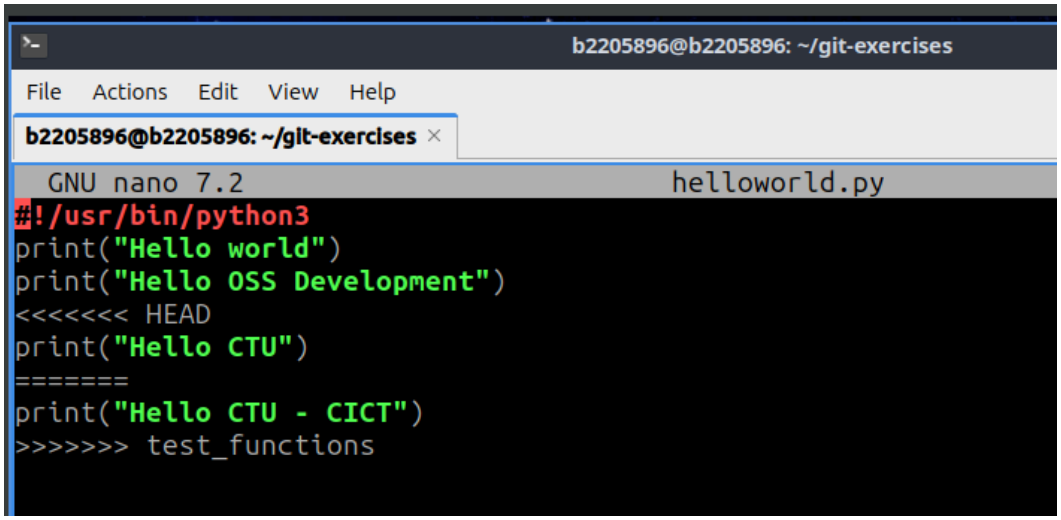


- Swap back to the “master” branch, merge “test_functions” branch to “master”

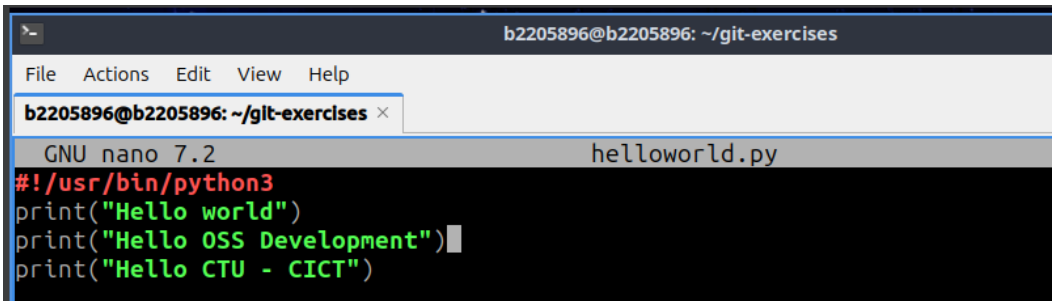
```
 $git checkout master
$git merge test_functions
```

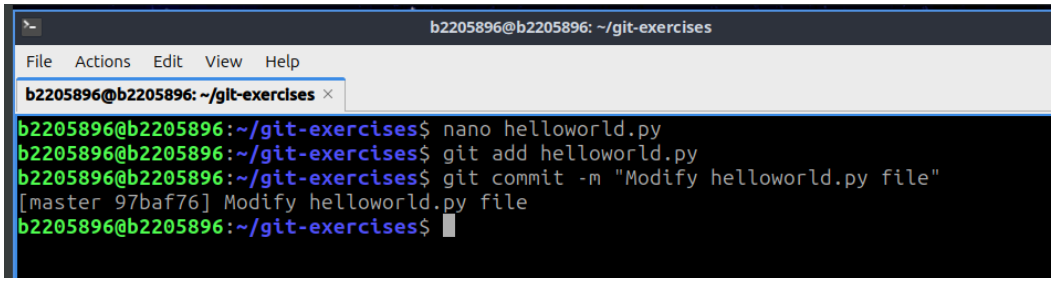


- Are there any conflicts? If YES, resolve the conflict (i.e. edit the conflict markers to match how you want the file to look - like) and commit the result. Use git log to see the resulting commits on the master branch.



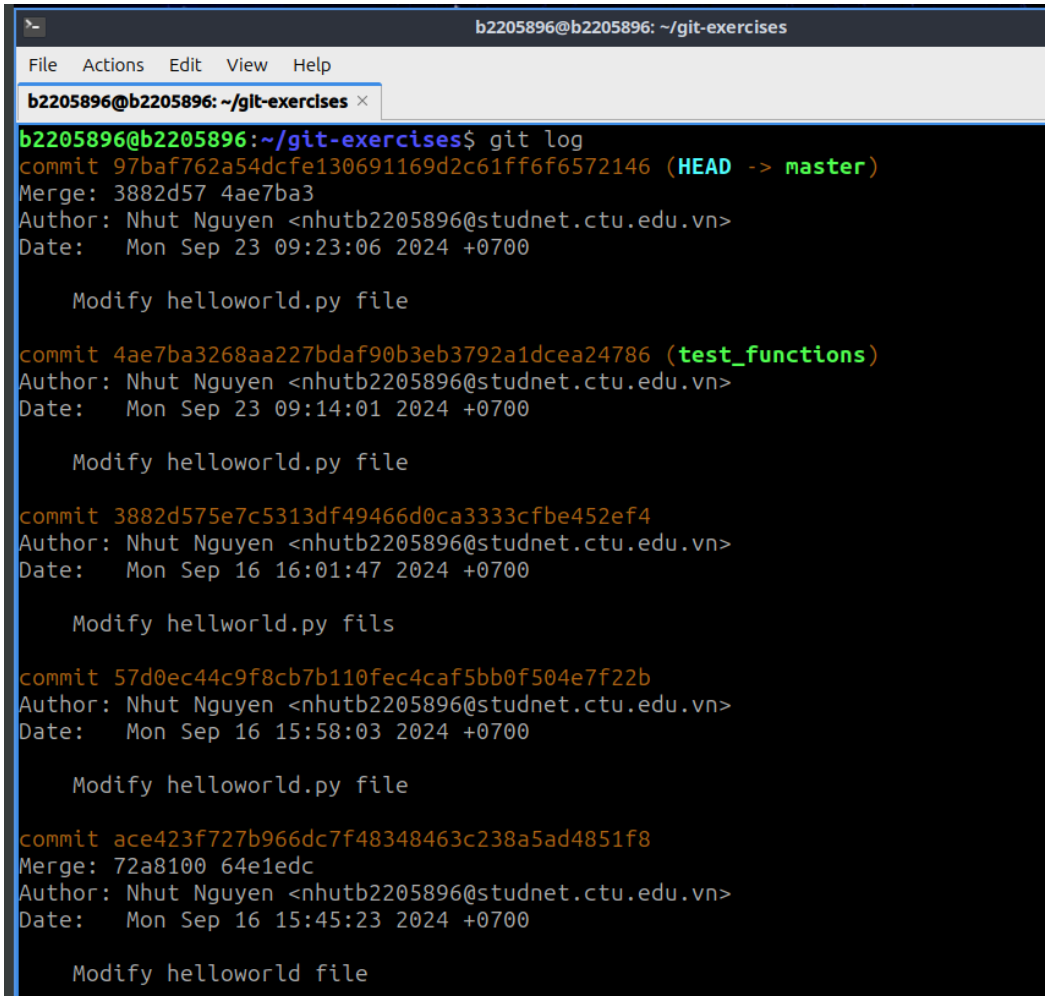
Fix the conflicts





- View the log to see the resulting commits on the master branch.

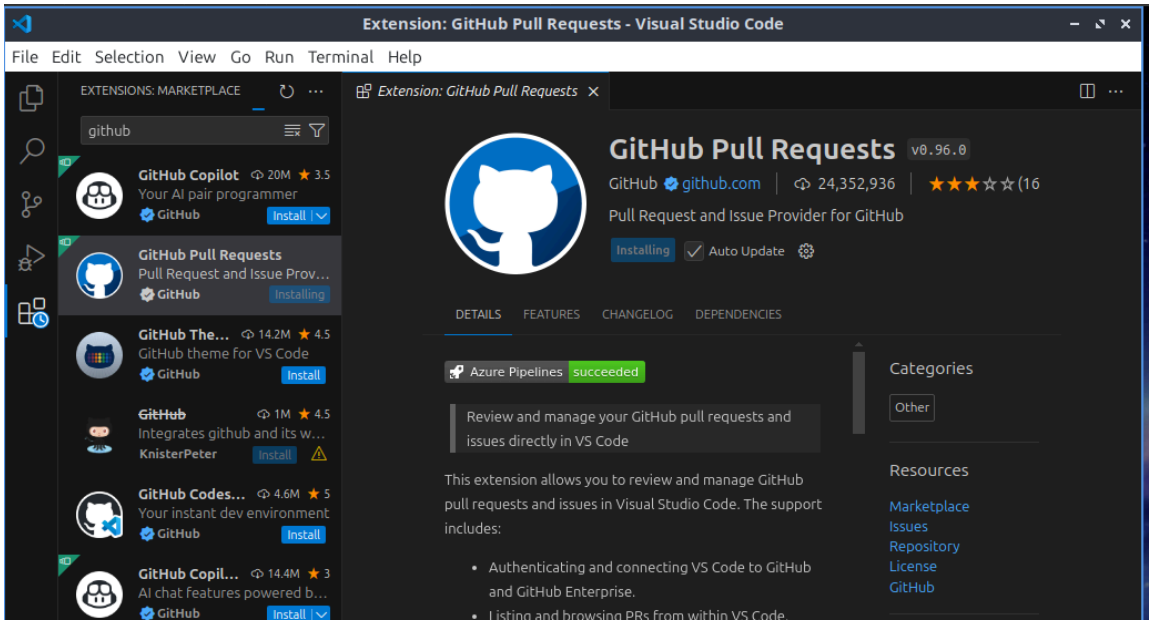
```
$git log
```



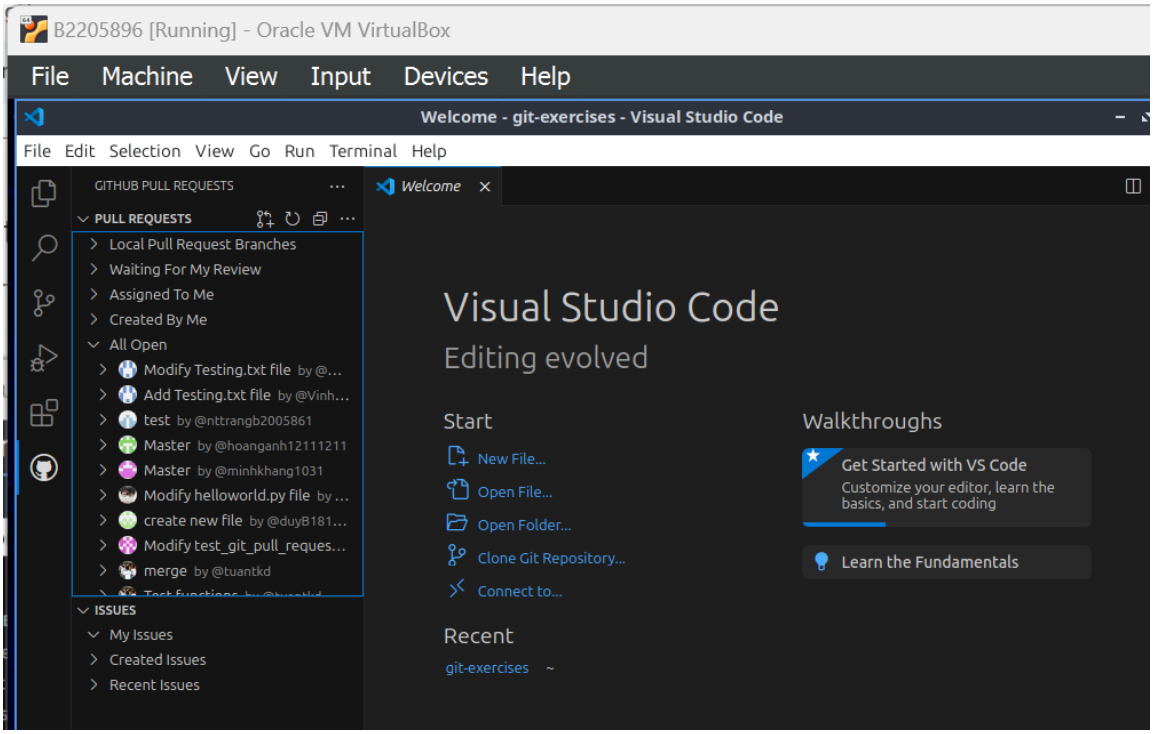
(take screenshots to show that you finish this exercise)

3.Working with GitHub in VS Code

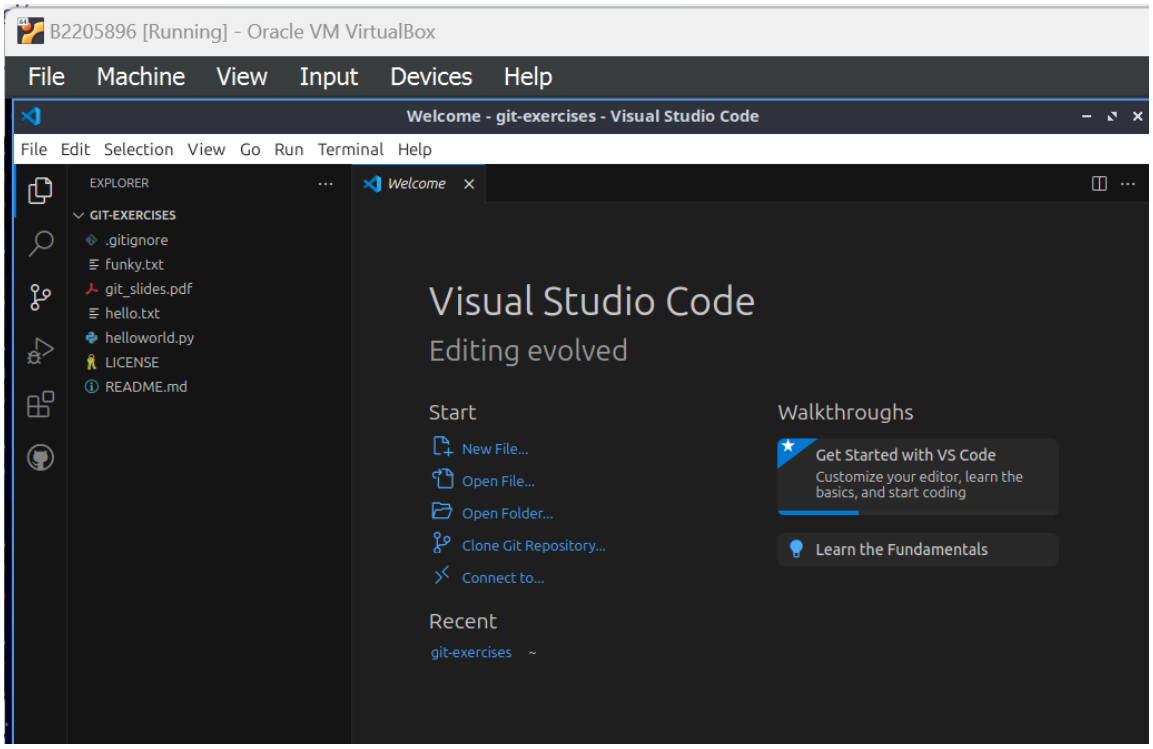
- Install VS Code and its GitHub Pull Requests-Issues extension to your computer.



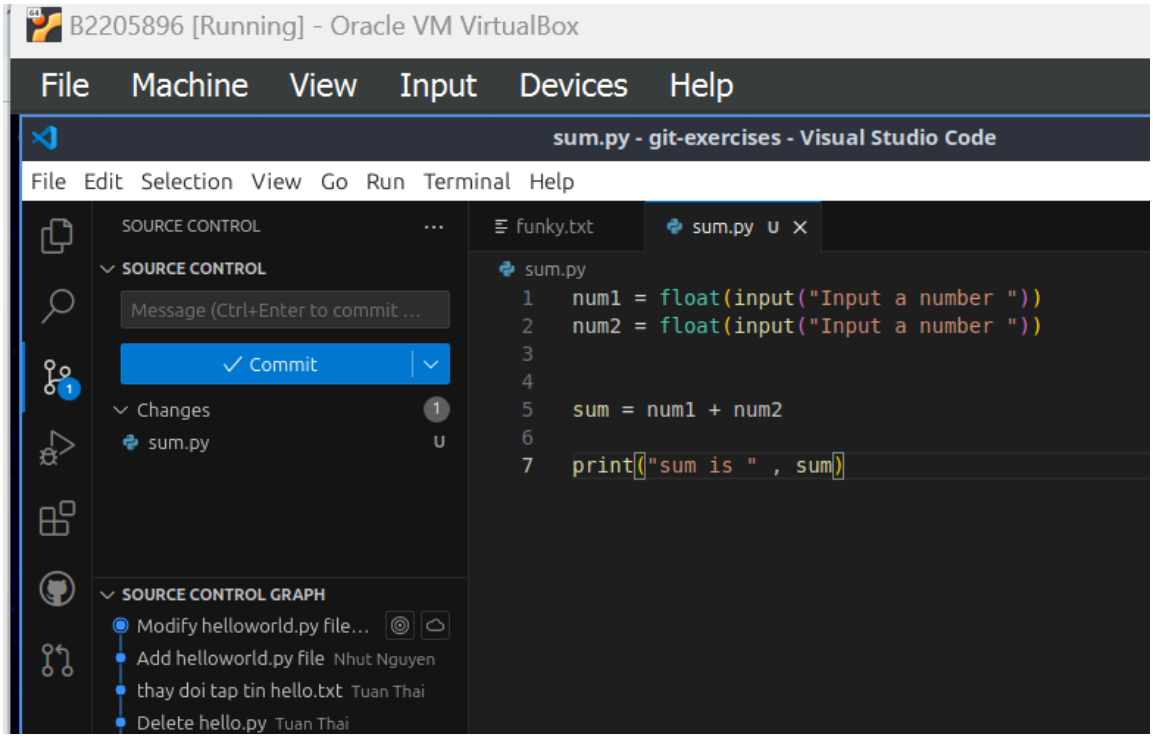
- Sign in GitHub using GitHub Pull Requests and Issues extension.

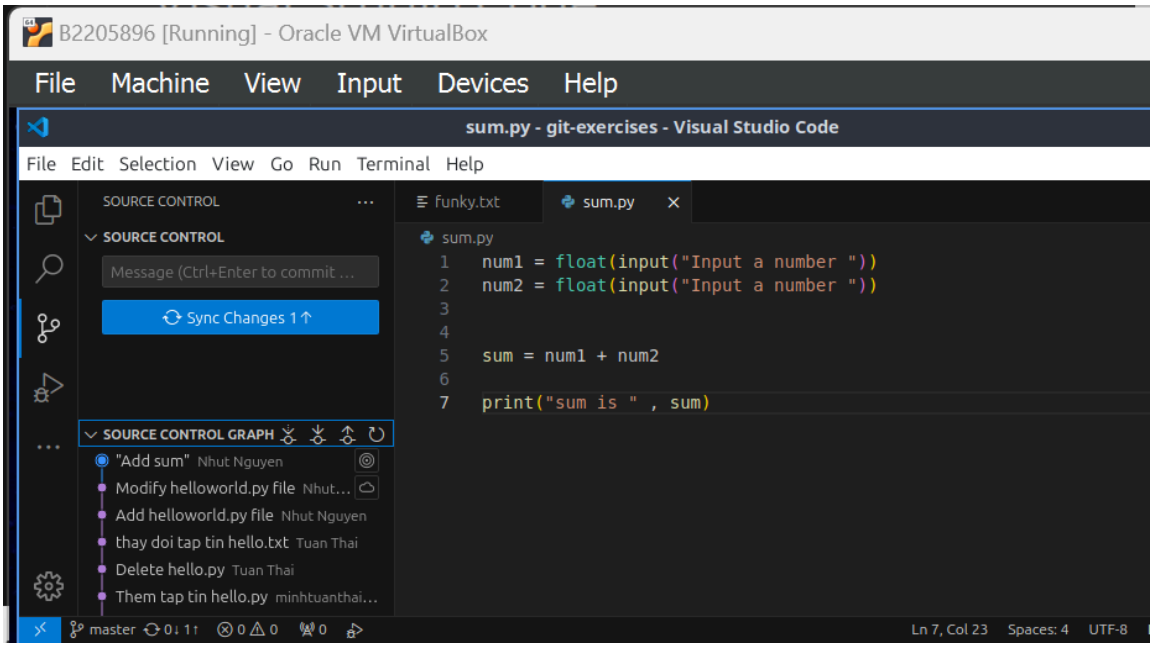


- Clone a remote repository to your computer.

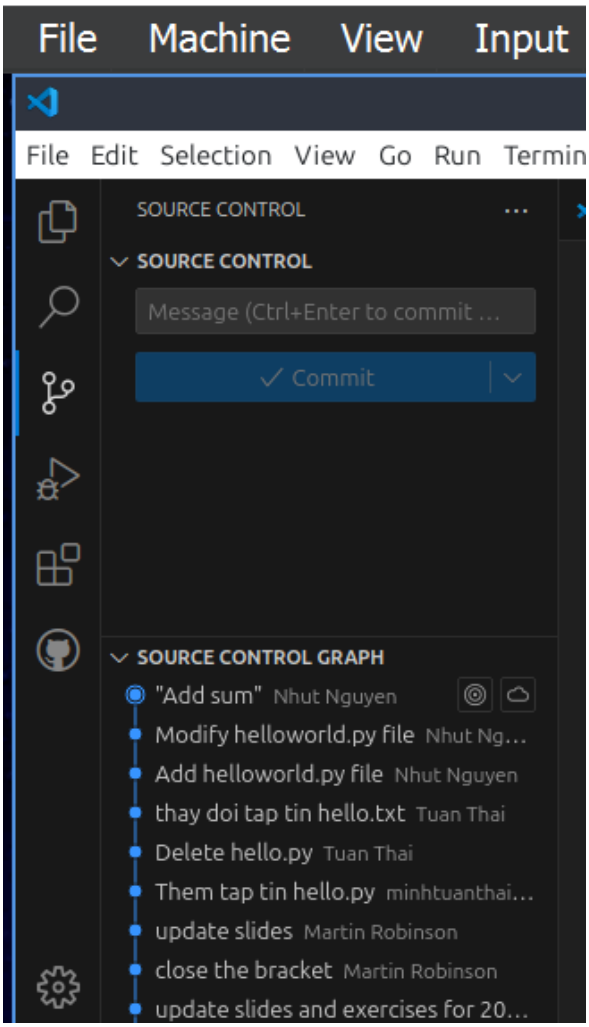


- Create and add a new file to the local repository, then commit everything you have done so far.





- Push the commits to the remote repository.



(take screenshots to show that you finish this exercise)

---END---