1. Accept github classroom assignment
   https://classroom.github.com/a/heoHK3yx

2. Generate and add a SSH key
   - Open Terminal
   - ssh-keygen -t ed25519 -C your_email@example.com
   - Enter a file in which to save the key (/Users/YOU/.ssh/filename_here): [Press enter]
   - Paste in terminal: open /Users/YOU/.ssh/filename_here.pub
     Or open ~/.ssh/filename_here.pub
     (There will be 2 keys generated, one with .pub, is a public key that you need, copy it to clipboard)
   - Go to Github /settings /SSH and GPG keys
   - Add the generated public key there and put a desired title
     Here MacOS users must do to fix errors:
     - Open terminal again
     - Run: eval "$(ssh-agent -s)"
     - Check if you have a config file by running: open ~/.ssh/config
     - If not, to create a config file run: touch ~/.ssh/config
     - Run: open ~/.ssh/config
     - Modify the file with:

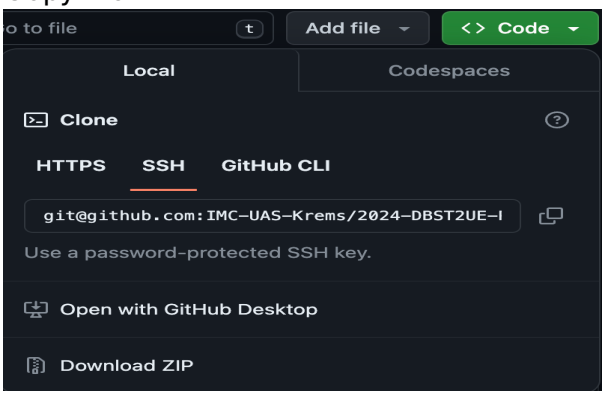
```
Host github.com
AddKeysToAgent yes
UseKeychain yes
IdentityFile ~/.ssh/id_ed25519
```

```
If you chose not to add a passphrase to your key, you should
omit the UseKeychain line.
If you see a Bad configuration option: usekeychain error, add
an additional line to the configuration's' Host *.github.com
section.
```

     - ssh-add --apple-use-keychain ~/.ssh/id_ed25519

3. Naviagte to repository generated by github classroom
   The links shoul look like this:
   https://github.com/IMC-UAS-Krems/2024-dbst2ue-Your_nick_name_here
   On this step github classroom made a repository for us, with same structure, it can be viewed by our professor.
4. Open in terminal desired location, for example: cd ~/Documents/Uni
5. Naviagte to your repo
   https://github.com/IMC-UAS-Krems/2024-dbst2ue-Your_nick_name_here

Copy the link



6. Run: git clone YOUR_LINK_HERE
7. Add our course's Public Repository as git submodule to your project under the public folder
8. https://github.com/IMC-UAS-Krems/2024-DBST2UE-Public
   Run: git add submodule LINK_HERE public
   We have generated by githubclassroom repository is visible only for us and only we can change it, it contains an README.md file and .github folder. Now we are basically ading the public repository ofr exercises, where the professor display the exercises, to our repository. So we will have a private own repo with our homework, exercises and we will have another folder inside which will be updated in future by professor.
9. Run: git add .
   This command add all our files in the folder to the version contrl system
10. Run: git commit -m “Your message here”
    A commit in Git captures a snapshot of your project's currently staged changes
11. Run: git push
    After push, git will push all your changes on local machine to the remote repository(online one)
12. Run: git submodule update –remote
    This command will update the submodule public, which is our public repository where professor added exercises.

Here are some useful git commands:
https://www.freecodecamp.org/news/10-important-git-commands-that-every-developer-should-know/

Here are some basic bash commands:
https://www.educative.io/blog/bash-shell-command-cheat-sheet

Also my personal recommendation to windows users, use bash insetad of cmd or windows power shell. If you have installed git on PC, you can search in your apps for Git Bash, or you can use Hyper, a super customizable terminal and free, I used this one with windows.
https://hyper.is/