Exercise 1

use bash and git commands

1. Make a directory (***exercise-1***) and change the working directory to ***exercise-1***
    a. mkdir exercise-1 && cd exercise-1
2. Create a repository and run **git status** and **git log**
    a. git init
    b. git status
    c. git log
3. Create **README.md** file of your repository and write **# *Exercise 1*** to it
    a. echo “# Exercise 1” > README.md
4. Run **git status**. You should have an **untracked** file.
    a. git status
5. **Try** to remove **REAME.md** using **git rm**. Remove the file using bash command, **rm**.
    a. It is not possible to delete an untracked file using git rm
    b. rm README.md
6. Do **step 3** and add **README.md** to the **staging** area of the repository.
    a. echo “# Exercise 1” > README.md
    b. git add README.md
7. Run **git status** and **unstage** the change using **git rm**.
    a. git status
    b. git rm --cached README.md
8. Add **README.md** to the **staging** area and **try** to unstage it using **git restore** and **git checkout**.
    a. git add README.md
    b. It is not possible to unstage changes when there is no commit (no HEAD).
9. Make a **commit** with message “**Initial commi**t”
    a. git commit -m “Initial commit”
10. Run **git status**, **git log** and **git log --oneline**. What is the short commit hash?
    a. git status
    b. git log
    c. git log --oneline
11. Remove **README.md** from the repository using **git rm**.
    a. git rm --force README.md
12. Run **git status** and then **unstage** the change using **git restore** or **git reset**.
    a. git status
    b. git restore --staged README.md
13. Run **git status** and **discard** changes using **git restore** or **git checkout**.
    a. git status
    b. git restore README.md

14. Create **text.txt** file and **ignore** it. Run **git status**.
    a. touch text.txt && echo "text.txt" > .gitignore
    b. git status
15. Add the change to the staging area and the **commit** it with message “**First commit**”
    a. git add .
    b. git commit -m “First commit”
16. Run **git status** and **git log**. Change the message of the last commit to “**Ignored text.txt**”
    a. git status
    b. git log
    c. git commit --amend -m “Ignored text.txt”
17. Run **git log**. Create **document.txt** and add it to the staging area.
    a. git log
    b. touch document.txt && git add document.txt
18. Run **git status** and **ignore** all **.tx**t files.
    a. git status
    b. echo "*.txt" > .gitignore
19. Unstage **document.txt** and stage **.gitignore** and run **git status**.
    a. git restore --staged document.txt
    b. git add .gitignore
    c. git status
20. Commit the change with message “**Ignored all .txt files**”
    a. git commit -m “Ignored all .txt files”
21. Run **git log --oneline** and **checkout** the **first commit**.
    a. git log --oneline
    b. git checkout a3acf67 (a3acf67 is the commit id of the first commit)
22. Run **git log --oneline** and **checkout** the **last** commit using **git checkout master** or **git switch -**.
    a. git log --oneline
    b. git checkout master **or** git switch - **or** git switch master