TASK	Commands ran (where applicable)	Output for (git log --oneline)
Get a github.com free account. Install a local CLI tool.	X	X
Create/Use a pair of SSH keys to authenticate from CLI.	x	X
Create your first test repo named "**mytestrepo"** using Github UI, initialise with a readme.	x	X
Clone your first test repo locally (and ensure you use the **main** branch).	git clone https://github.com/robagabriel/mytestrepo.git	X
Create a file called "**main.txt**" containing the text "**main**", commit it with "**main**" message and push it to github.	git status; git add .; git status; git commit -m “main”, git status; git push	X
Create a new branch called "**test1**" from **main**, create a file called "**test1.txt**" containing the text "**test1**", commit it with "**test1**" message and push it to github.	git checkout -b test1;git branch -a; git status; git add .; git commit -m “test1”; git push –set-upstream origin test1	PS D:\learning\mytestrepo> git log --oneline d6f687d (HEAD -> test1, origin/test1) test1 42500c4 (origin/main, origin/HEAD, main) main e2e1d1c Initial commit
Clone the same repository from github (again) under a separate directory called "**mytestrepo2**". You will now have **[mytestrepo]** and **[mytestrepo2]** to simulate 2 devs working on the same repo.	git clone https://github.com/robagabriel/mytestrepo.git mytestrepo2	PS D:\learning\mytestrepo2> git log --oneline d6f687d (HEAD -> test1, origin/test1) test1 42500c4 (origin/main, origin/HEAD, main) main e2e1d1c Initial commit
[mytestrepo2] -> Checkout a new branch called test2 from test1, create a new file named "test2.txt" containing the text "test2", commit it with "test2" message and push it to github.	git checkout -b test2;git branch -a; git status; git add .; git commit -m “test2”; git push –set-upstream origin test2	PS D:\learning\mytestrepo2> git log --oneline 2ddefc7 (HEAD -> test2, origin/test2) test2 d6f687d (origin/test1, test1) test1 42500c4 (origin/main, origin/HEAD, main) main e2e1d1c Initial commit
[mytestrepo2] -> Edit the previously created file (test2.txt), change the text inside to "none", commit it with "test2a" message and push it to	git add .; git commit -m “test2a”; git push –set-upstream origin	PS D:\learning\mytestrepo2> git log --oneline fc68f47 (HEAD -> test2, origin/test2) test2a 2ddefc7 test2

github.	test2	d6f687d (origin/test1, test1) test1 42500c4 (origin/main, origin/HEAD, main) main e2e1d1c Initial commit
[mytestrepo] -> Ensure you are still on the test1 branch. Edit test1.txt, change the text inside to "none", commit it with "test1a" message and push it to github.	git branch -a;git checkout test1; git add .; git commit -m "test1a"; git push --set-upstream origin test1	PS D:\learning\mytestrepo> git log --oneline 98b7cc3 (HEAD -> test1, origin/test1) test1a d6f687d test1 42500c4 (origin/main, origin/HEAD, main) main e2e1d1c Initial commit
[mytestrepo] -> Merge test1 into main using the fast forward way.	git branch -a;git checkout main; git merge test1	PS D:\learning\mytestrepo> git log --oneline 98b7cc3 (HEAD -> main, origin/test1, test1) test1a d6f687d test1 42500c4 (origin/main, origin/HEAD) main e2e1d1c Initial commit
[mytestrepo] -> Checkout a new branch called "test3" from test1. Create a new file named "test3.txt" containing the text "test3". commit it with "test3", but DO NOT Push it.	git checkout -b test3; git add .; git commit -m "test3";	PS D:\learning\mytestrepo> git log --oneline cafa5ef (HEAD -> test3) test3 98b7cc3 (origin/test1, test1, main) test1a d6f687d test1 42500c4 (origin/main, origin/HEAD) main e2e1d1c Initial commit
[mytestrepo2] -> Checkout test2 and *rebase* it with latest version of test1. Push to remote by forcing history.	git rebase test1; git pull; git push --set-upstream origin test2	PS D:\learning\mytestrepo2> git log --oneline 22513dc (HEAD -> test2, origin/test2) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2 d7876d3 test2a bd241da test2 98b7cc3 (origin/test1, test1) test1a fc68f47 test2a 2ddefc7 test2 d6f687d test1 42500c4 (origin/main, origin/HEAD, main) main e2e1d1c Initial commit
[mytestrepo] -> On test3, ensure all commits from latest test2 are	git branch -a; git checkout test3; git	PS D:\learning\mytestrepo> git log --oneline

present in the tree and that your commit named "test3" is the *last* commit in the history.	rebase test2;	7306315 (HEAD -> test3) test3 22513dc (origin/test3, origin/test2, test2) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2 d7876d3 test2a bd241da test2 98b7cc3 (origin/test1, origin/main, origin/HEAD, test1, main) test1a fc68f47 test2a 2ddefc7 test2 d6f687d test1 42500c4 main e2e1d1c Initial commit
[mytestrepo] -> Ensure all commits are present in main, with a merge commit added (merge commit message "merge commit"). (push to github)	git checkout test1; git merge test3; git checkout main; git merge test1 --no-ff -m "merge commit"; git push	PS D:\learning\mytestrepo> git log --oneline --graph * 3fd919a (HEAD -> main, origin/main, origin/HEAD) merge commit |\ | * 7306315 (test3, test1) test3 | * 22513dc (origin/test3, origin/test2, test2) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2 | |\ | | * fc68f47 test2a | | * 2ddefc7 test2 | * | d7876d3 test2a | * | bd241da test2 |/ / * / 98b7cc3 (origin/test1) test1a |/ * d6f687d test1 * 42500c4 main * e2e1d1c Initial commit
[mytestrepo] -> Delete local & remote (where applicable) branches test2 and test3	**git branch -D test2; git branch -D test3 git push origin --delete test2**	X
Delete local directory **[mytestrepo2]**, continue working on **[mytestrepo]**	Remove-Item .\mytestrepo2\ -Recurse -Force	X
Checkout **test1**, edit the file named "**main.txt**", change the text inside to	git add .; git commit -m "squash1"	PS D:\learning\mytestrepo> git log --oneline --graph

<table>
<tr><td>"**squash1**", commit it with message **squash1**</td><td></td><td>* 0c5be3c (HEAD -> test1) squash1
* 7306315 (origin/test1) test3
*   22513dc (origin/test3) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2
|\
| * fc68f47 test2a
| * 2ddefc7 test2
* | d7876d3 test2a
* | bd241da test2
* | 98b7cc3 test1a
|/
* d6f687d test1
* 42500c4 main
* e2e1d1c Initial commit</td></tr>
<tr><td>Checkout a new branch called "**squash**" from **main**, edit the file named "**main.txt**", change the text inside to "**squash2**", commit it with message **squash2**</td><td>git checkout main; git checkout -b squash; git add .; git commit -m "squash2"</td><td>PS D:\learning\mytestrepo> git log --oneline --graph
* b233969 (HEAD -> squash) squash2
*   3fd919a (origin/main, origin/HEAD, main) merge commit
|\
| * 7306315 (origin/test1) test3
| *   22513dc (origin/test3) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2
| |\
| | * fc68f47 test2a
| | * 2ddefc7 test2
| * | d7876d3 test2a
| * | bd241da test2
|/ /
* / 98b7cc3 test1a
|/
* d6f687d test1
* 42500c4 main
* e2e1d1c Initial commit</td></tr>
<tr><td>Merge **test1** into **squash**. Fix Conflicts by preserving the **HEAD** values.</td><td>git checkout **squash**;git merge test1 --no-ff;git commit -a</td><td>PS D:\learning\mytestrepo> git log --oneline --graph
*   8798ecb (HEAD -> squash) Merge branch 'test1' into squash
|\
| * 0c5be3c (test1) squash1
* | b233969 squash2
* | 3fd919a (origin/main, origin/HEAD,</td></tr>
</table>

<table>
<tr><td></td><td></td><td>main) merge commit
|\|
| * 7306315 (origin/test1) test3
| *   22513dc (origin/test3) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2
| |\
| | * fc68f47 test2a
| | * 2ddefc7 test2
| * | d7876d3 test2a
| * | bd241da test2
|/ /
* / 98b7cc3 test1a
|/
* d6f687d test1
* 42500c4 main</td></tr>
<tr><td>Merge <b>squash</b> into <b>main</b>, but <b>squash</b> ther 3 extra commits you have into 1 commit. Push main.</td><td>git checkout main; git merge --squash squash; git commit</td><td>PS D:\learning\mytestrepo> git log --oneline --graph
* 71dbdd3 (HEAD -> main) Squashed commit of the following:
*   3fd919a (origin/main, origin/HEAD) merge commit
|\
| * 7306315 (origin/test1) test3
| *   22513dc (origin/test3) Merge branch 'test2' of https://github.com/robagabriel/mytestrepo into test2
| |\
| | * fc68f47 test2a
| | * 2ddefc7 test2
| * | d7876d3 test2a
| * | bd241da test2
|/ /
* / 98b7cc3 test1a
|/
* d6f687d test1
* 42500c4 main
* e2e1d1c Initial commit</td></tr>
<tr><td>Search trough your commit history, and create a branch named "<b>cherry</b>" out of the commit with message "<b>test1a</b>" , and add only the commit with message "<b>test3</b>" on top. Push</td><td>git log --oneline --graph --grep="test1a" --grep="test3";git branch cherry</td><td>PS D:\learning\mytestrepo> git log --oneline --graph
* 7605af3 (HEAD -> cherry, origin/cherry) test3
* 98b7cc3 test1a</td></tr>
</table>

cherry to github. Do that in only **5** steps, including the search and push commands.	98b7cc3; git checkout cherry;git cherry-pick 7306315; git push origin cherry	* d6f687d test1 * 42500c4 main * e2e1d1c Initial commit
On **cherry,** edit the file **test2.txt,** change its contents to **"trap".**	No file	X
On **cherry,** edit the file **test3.txt,** change its contents to **"unsure".** Commit with message **"unsure"**	git commit -am "unsure"	PS D:\learning\mytestrepo> git log --oneline --graph * 6e1284c (HEAD -> cherry) unsure * 7605af3 (origin/cherry) test3 * 98b7cc3 test1a * d6f687d test1 * 42500c4 main * e2e1d1c Initial commit
Destroy the last commit, but leave the staged files intact. (soft reset)	git reset --soft 7605af3	PS D:\learning\mytestrepo> git log --oneline --graph * 7605af3 (HEAD -> cherry, origin/cherry) test3 * 98b7cc3 test1a * d6f687d test1 * 42500c4 main * e2e1d1c Initial commit
Unstage the **test3.txt** file. Stash your changes. Apply the stashed changes on the **main** branch. add, commit, push.	git reset --hard 7605af3	PS D:\learning\mytestrepo> git log --oneline --graph * 7605af3 (HEAD -> cherry, origin/cherry) test3 * 98b7cc3 test1a * d6f687d test1 * 42500c4 main * e2e1d1c Initial commit

—--

Git + Bash

Fa un fork la repo-ul LISAv2-1 de pe userul adriansuhov de pe Github

Apoi, pe o masina de linux noua (ubuntu) pe **azure**:

cloneaza forkul pe masina

Gaseste Sysbench.sh in repo-ul clonat
Executa-l
Rezolva problemele daca apar
Scriptul o sa dureze la prima rulare undeva la 5 minute, ai rabdare.

=> vreau copy paste ultimele 3 linii de la output + history la ce ai facut pana acum

```
Thu Apr 21 12:51:26 2022 : Testing fileio. Writing to fileio.log.
Thu Apr 21 12:51:26 2022 : Fileio tests passed.
Thu Apr 21 12:51:26 2022 : Test Failed.
```

=> sa faci un pic munca de detectiv, de ce crezi ca ai output-ul pe care il ai. Sa te uiti in script si in fisierele nou create. Chiar si sa imi zici cam ce face scriptul e o solutie. Take your time & let me know cand ai ceva

=> gitignore
=> un script

—---

Fibonacci (recursive)
Write a program which can compute the factorial of a given numbers.
Hangman bash script - (write the final script below**)**

Cursuri
—---

https://cognizant.udemy.com/course/complete-python-bootcamp/ - pana la Milestone Project 2 inclusiv