**Git Assignment**

Complete the below exercise by pasting the command you followed at each step, on the line next to the step. Send it back once you complete the exercise.

Linus and Junio are software developers who work from different locations, want to start a project. Linus created a folder, want to do the below steps to set up the project repository.

1. Do git init in the created folder

2. Create a README.md file with below text

"""

This is readme file

Add development flow steps to this file

"""

3. Stage the newly created file

4. Commit the file with message "initial commit"

**STEP WE DONE**

1- Create a repo named TASK1

2- Do **git init** in local directory

3- Created a README.md file by using command **echo "content"**

4- Staged the created README file by using command **git add README.md**

5- Finally committed the file by using **git commit** command with the message initial commit

Linus thought to change the commit message which tell more on the context than just the initial commit message.

5. Do change commit message to "Readme file added"

6. Go to gitlab (browser) and create a repository

6. Copy the repository url and add it as "origin" to your local git

7. Check whether origin url is set

8. Push the commit to origin/master

**STEP WE DONE**

1- The changes made by using command **git commit --amend -m "Readme file added"**

2- Create a repo in github

3- Copied the Http url and add by using **git add <url>**

4- Checked the url by using **git remote show origin**

5- push the commit by using **git push -u origin master**

Linus asked Junio to update readme file. Junio started his task with below steps

9. Diverge from master and create a new branch on name "add-steps-to-readme"

10. Edit the file and add development flow. File should have below text after adding steps

"""

This is readme file

1. Requirement Analysis

2. Information gathering

3. Implementation

4. Sync work with remote repository

"""

11. Check the changes you made with diff. Proceed further if the diff is good

12. Stage & Commit the changes with message "steps added to readme"

13. Push the branch and raise a PR against master branch. Keep the PR open, don't merge

Junio shared the PR details with Linus.

**STEP WE DONE**

1- Diverge from master and created a new branch named "add-steps-to-readme" by using **git checkout -b name**

2- Edit the content

3- check by using **git diff** command

4- Stage and commit, by using **git add** and **git commit** command

5- Push the branch by using **git push** command

Linus want to add some more steps to readme file. Because, Junio's PR is not merged with master yet, Linus has to pull Junio branch and diverge from it. Linus started doing below steps

13. Diverge from "add-steps-to-readme" branch by creating a new branch on name "update-readme"

14. Edit the file and update the development flow. File should have below text after updating steps

"""

This is readme file

1. Requirement Analysis

2. Information gathering

3. Implementation

4. Test implementation

5. Sync work with remote repository

"""

15. Check the changes you made with diff. Proceed further if the diff is good

16. Stage & Commit the changes with message "steps updated in readme"

17. Push the branch and raise a PR against "add-steps-to-readme" branch. Keep the PR open, don't merge

**STEP WE DONE**

1- Pull the PR using command **git pull origin add-steps-to-readme**

2- Diverge from "add steps to readme" and created a new branch named "update readme" by using **git checkout -b <name>**

3- Checked the changes by using **git diff** command

4- Stage and commit, by using **git add** and **git commit** command

6- Push the branch by using **git push** command and raise the PR

Linus wants to add a few steps to readme in his branch.

18. Checkout / Change to "add-steps-to-readme" branch

19. Edit the file and update the development flow. File should have below text after updating steps

"""

This is readme file

1. Requirement Analysis

2. Information gathering

3. Discuss on requirement with peers and leads, who ever involve

4. Implementation

5. Sync work with remote repository

"""

20. Check the changes you made with diff. Proceed further if the diff is good

21. Stage & Commit the changes with message "steps updated in readme"

22. Push the changes to "origin/add-steps-to-readme" branch. PR is still open, Don't merge

**STEP WE DONE**

1- switched to existing branch add-steps-to-readme from update-readme using command **git switch add-steps-to-readme**

2- Checked the changes by using **git diff** command

3- Stage and commit, by using **git add** and **git commit** command

4- Push the branch by using **git push** command and raise the PR

Linus told Junio that he updated his branch by pushing a commit. Junio before editing the file further, he wants to update the target branch in his PR. He has two ways to do this. Follow either of the below steps

Way 1:

24.a. Checkout to target branch, i.e., "add-steps-to-readme"

24.b. Update local branch by pulling "origin/add-steps-to-readme"

24.c. Checkout to your branch, i.e., "update-readme"

24.d. Do rebase with "add-steps-to-readme"

**STEP WE DONE**

1- Checkout the target branch by using command **git checkout<target branch>**

2- Pull the local branch by using command **git pull origin add-steps-to-readme**

3- First fetch the remote branch to local by using command **git fetch**

4- Checkout the branch to update readme by using **checkout command**

Way 2:

24.i. Do pull + rebase with the target branch, i.e., "origin/add-steps-to-readme" in a single command

If rebase results in conflicts, Conflicting file paths will be shown in rebase result. Open the files individually, check and keep / delete the code. After adjusting the conflicts in all the shown files, do below steps

25. Add the modified files

26.a. Do continue rebase

26.b. If you fail to resolve conflicts, Do abort rebase

26.c. If you feel the conflicts are adjusted, but if git still identifies particular file as conflicting, Do skip rebase

Repeat 25, 26 steps till all the conflicts are resolved and you are "back to your branch". At any point of time, Can check status to see in which branch you are & the status on conflicting files.

27. Commit and push the resolved conflicts to "origin/update-readme". Do force push if git wont accept normal push.

**STEP WE DONE**

1- Pull the target branch using command **git pull origin add-steps-to-readme**

2- Then conflict occured and we used resolve in **merge editor** in **vs code**

3- Then resolve the conflict by using command **git rebase --continue**

4- Then commit and push by using command **git commit** and **git push**

Linus merged his PR with master & communicated the same to Junio. Now Junio has to change the target branch in his PR to "origin/master". He has four ways to do this. Follow any of the below steps

Way 1:

28.a. Checkout to target branch, i.e., "master"

28.b. Update local branch by pulling "origin/master"

28.c. Checkout to your branch, i.e., "update-readme"

28.d. Do rebase in "interactive-mode" with "master"

Way 2:

28.i. Do pull + rebase in "interactive-mode" with the target branch, i.e., "origin/master" in a single command

Way 3:

28.h. Check log & note your commit ids

28.i. Checkout / go to master

28.j. Pull latest master

28.k. Diverge from master by creating a temporary branch, say "temp" for pushing your commits

28.l. Cherry-pick all commit ids did in "update-readme" branch

28.m. Do forece push from temporary branch, "temp" to "origin/update-readme"

Way 4:

28.t. Check log & note your commit ids

28.u. Do pull + rebase "origin/master"

28.v. Check log & note the recent commit id from master

28.w. Reset to latest commit id from master

28.x. Cherry-pick all commit ids did in "update-readme" branch

28.y. Do forece push to "origin/update-readme"

29. Go to PR raised from "origin/update-readme" against "origin/master"

30. Update target branch in PR to "origin/master"

31. It should not show any conflicts on screen & the PR should be ready to merge. If it is, then merge it else do 24, 25, 26, 27 steps with target branch as "origin/master"

**STEPS WE DONE:**

1- **Merged the PR**