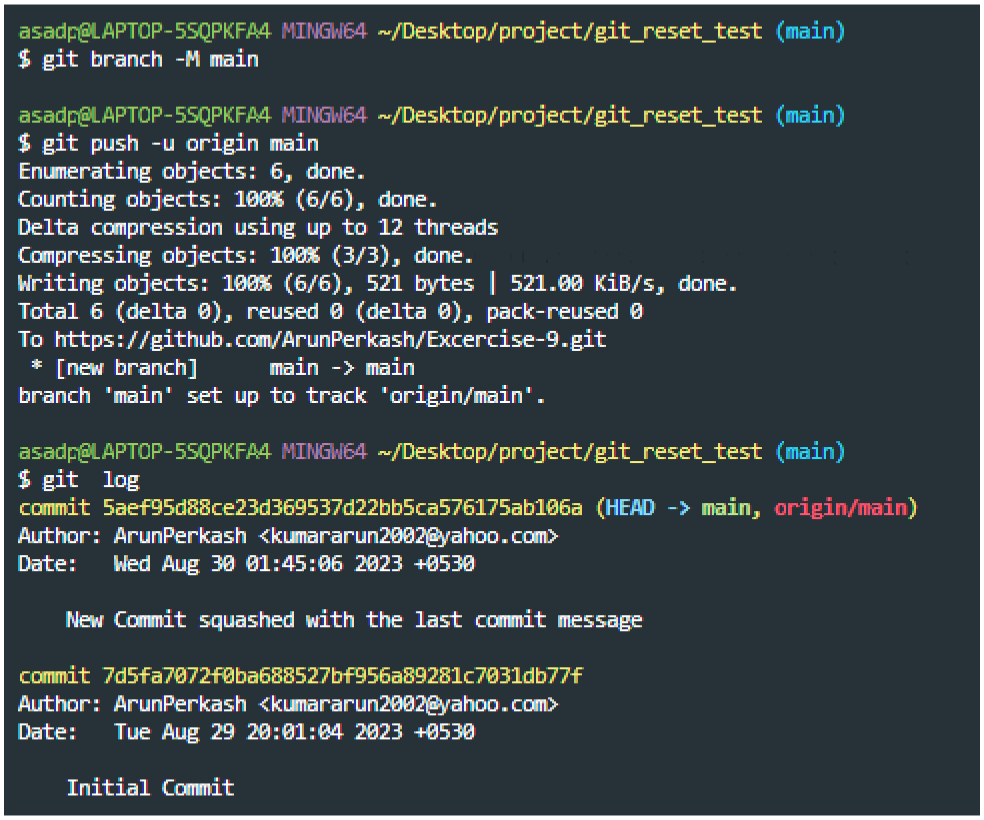
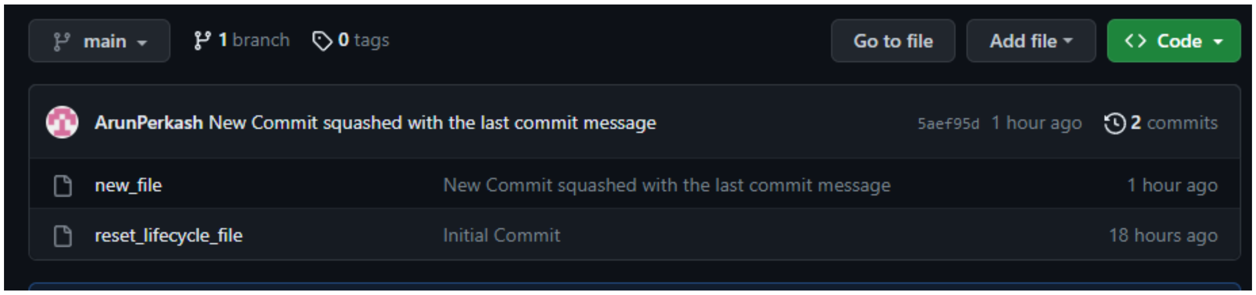
Exercise 9 Report

Q1) We created a new remote repository called “Experiment 9”.

Q2) we pushed all the commits from Exercise 7 and 8 to the remote repository



```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git branch -M main

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 521 bytes | 521.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ArunPerkash/Excercise-9.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git  log
commit 5aef95d88ce23d369537d22bb5ca576175ab106a (HEAD -> main, origin/main)
Author: ArunPerkash <kumararun2002@yahoo.com>
Date:   Wed Aug 30 01:45:06 2023 +0530

    New Commit squashed with the last commit message

commit 7d5fa7072f0ba688527bf956a89281c7031db77f
Author: ArunPerkash <kumararun2002@yahoo.com>
Date:   Tue Aug 29 20:01:04 2023 +0530

    Initial Commit
```



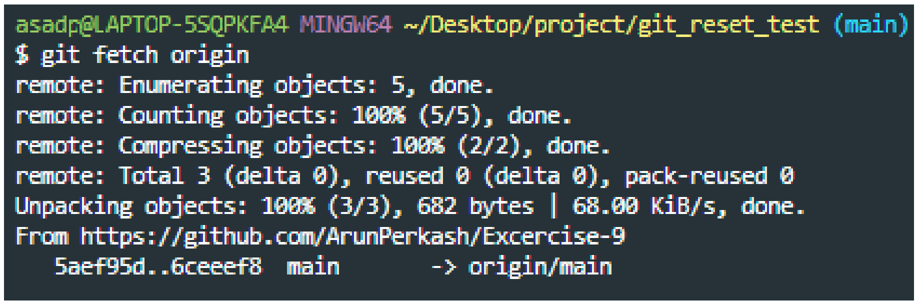
ArunPerkash New Commit squashed with the last commit message 5aef95d 1 hour ago 2 commits

new_file	New Commit squashed with the last commit message	1 hour ago
reset_lifecycle_file	Initial Commit	18 hours ago

Q3)

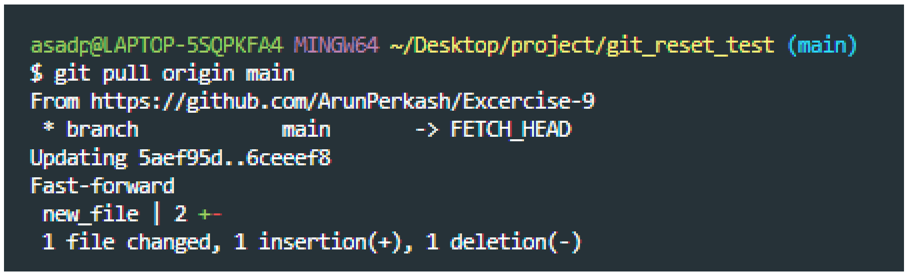
We changed the new_file in remote and added “hello” text to it.

git fetch origin will fetch the changes from the remote repository, hence the new_file we just changed will be fetched (shown in the screenshot below)



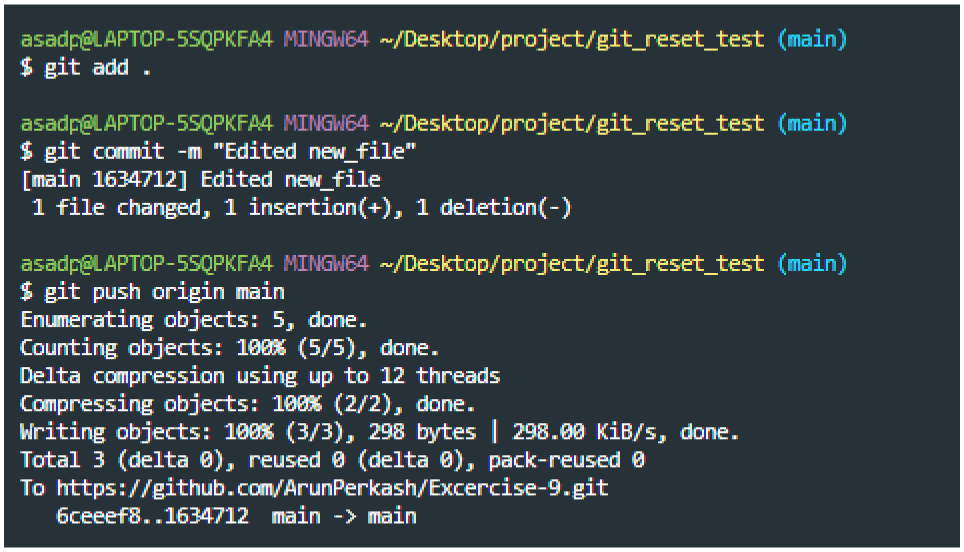
```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git fetch origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 682 bytes | 68.00 KiB/s, done.
From https://github.com/ArunPerkash/Excercise-9
   5aef95d..6ceeef8  main       -> origin/main
```

git pull origin main will pull the changes from the remote repository.



```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git pull origin main
From https://github.com/ArunPerkash/Excercise-9
 * branch            main       -> FETCH_HEAD
Updating 5aef95d..6ceeef8
Fast-forward
 new_file | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

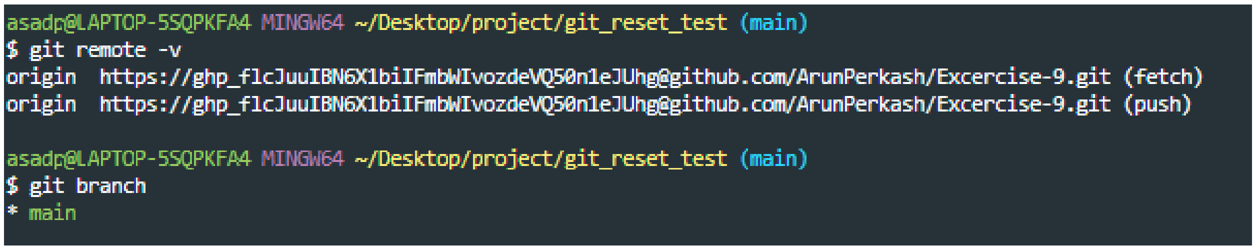
git push origin main will push the changes in our local repository to the remote repository. To demonstrate this, we change the content in new_file and push it to the repository.



```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git add .

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git commit -m "Edited new_file"
[main 1634712] Edited new_file
 1 file changed, 1 insertion(+), 1 deletion(-)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ArunPerkash/Excercise-9.git
   6ceeef8..1634712  main -> main
```

git remote -v will list all the remote repositories.

git branch will list all the current branches.

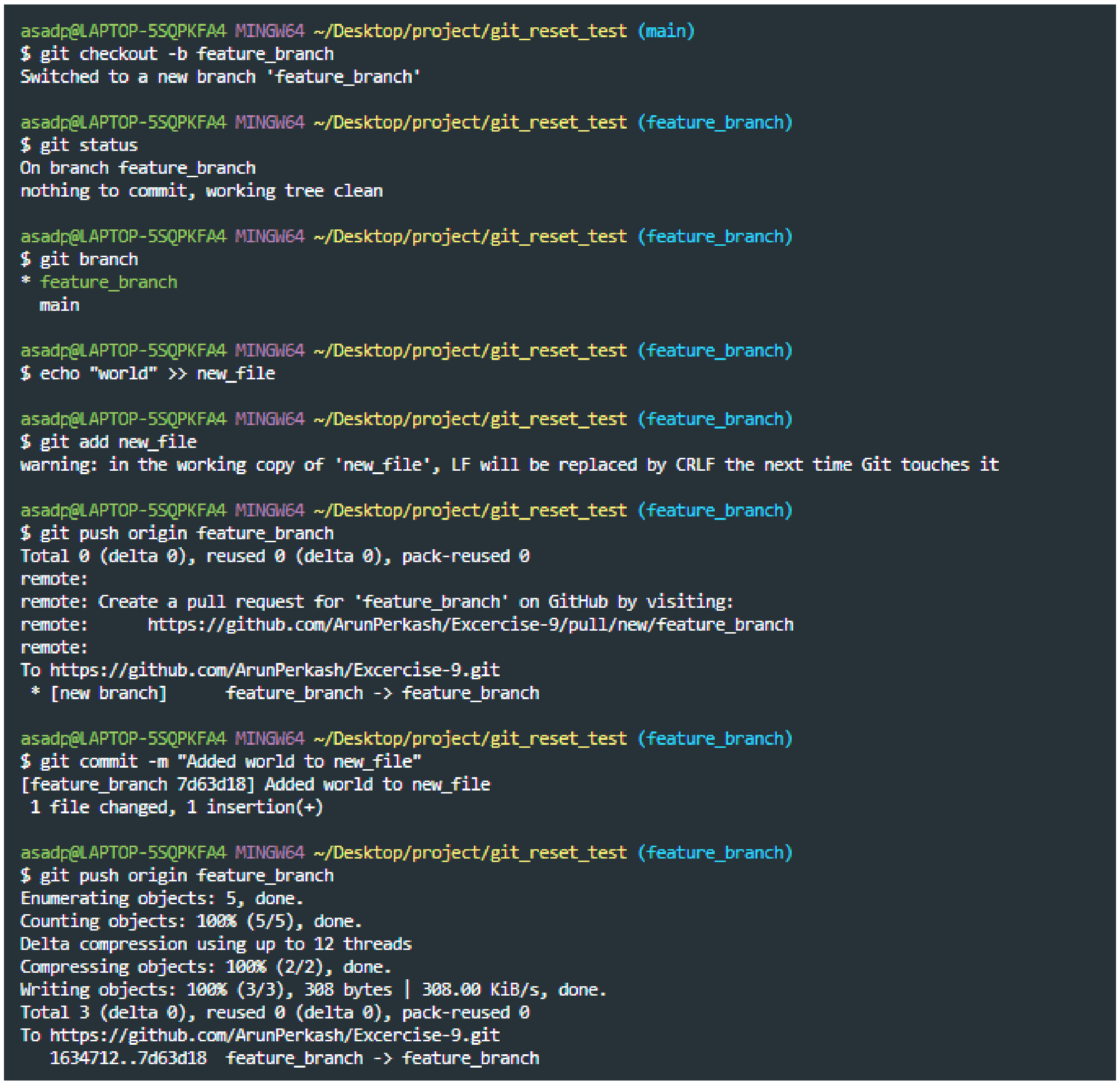


```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git remote -v
origin  https://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@github.com/ArunPerkash/Excercise-9.git (fetch)
origin  https://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@github.com/ArunPerkash/Excercise-9.git (push)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git branch
* main
```

Q4

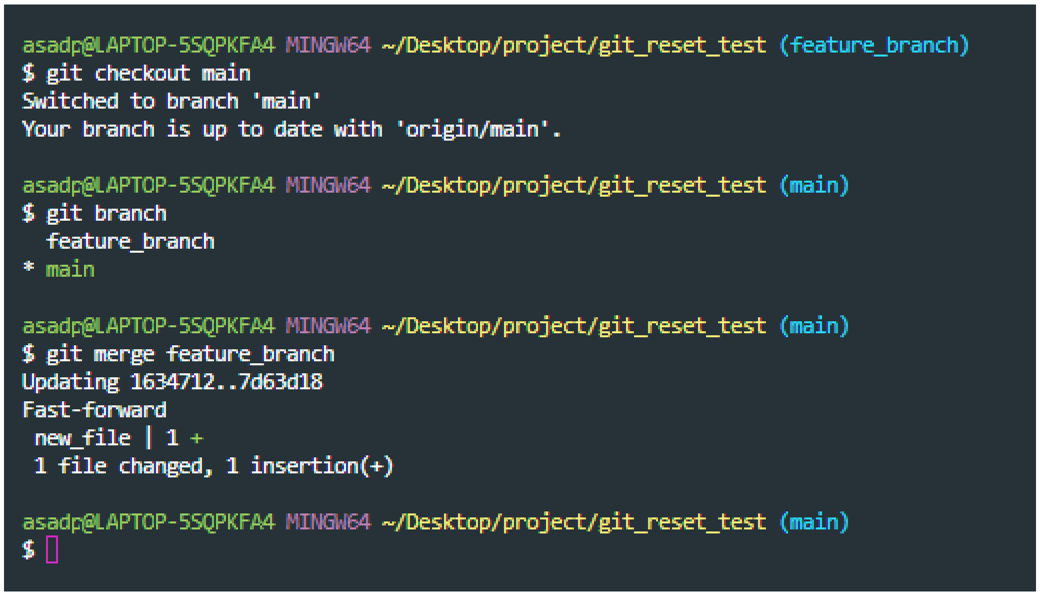
<u>Fast Forward Merge</u>

In order to perform this, we create a new branch called feature_branch from the main branch. We then add a new word “hello” to the new file and stage and commit our changes. The changes are then pushed to the feature branch.



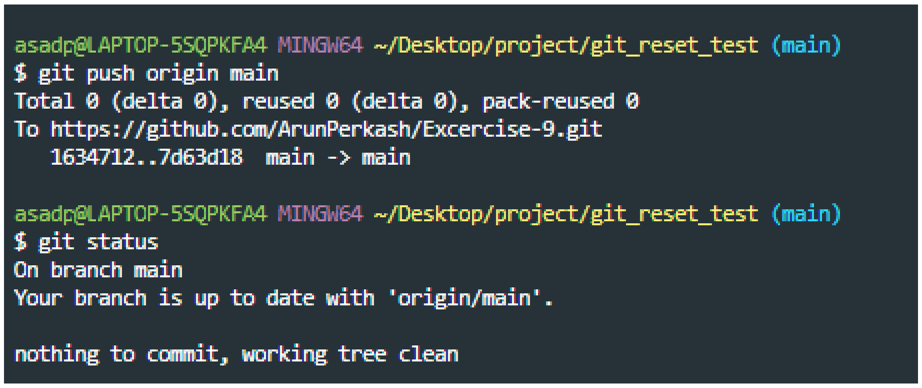
```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git checkout -b feature_branch
Switched to a new branch 'feature_branch'

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git status
On branch feature_branch
nothing to commit, working tree clean

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git branch
* feature_branch
  main

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ echo "world" >> new_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git add new_file
warning: in the working copy of 'new_file', LF will be replaced by CRLF the next time Git touches it

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git push origin feature_branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature_branch' on GitHub by visiting:
remote:      https://github.com/ArunPerkash/Excercise-9/pull/new/feature_branch
remote:
To https://github.com/ArunPerkash/Excercise-9.git
 * [new branch]      feature_branch -> feature_branch

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git commit -m "Added world to new_file"
[feature_branch 7d63d18] Added world to new_file
 1 file changed, 1 insertion(+)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git push origin feature_branch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ArunPerkash/Excercise-9.git
   1634712..7d63d18  feature_branch -> feature_branch
```

Fast forward merge is a type of merge that occurs when a branch you’re trying to merge has a linear history and is directly ahead of the branch you are merging into. In other words, there are no new commits on the branch you’re merging into since the branch you’re merging from diverged. The merge is “fast-forward” because Git can simply move the pointer of the branch you’re merging into to the latest commit of the branch you’re merging from .



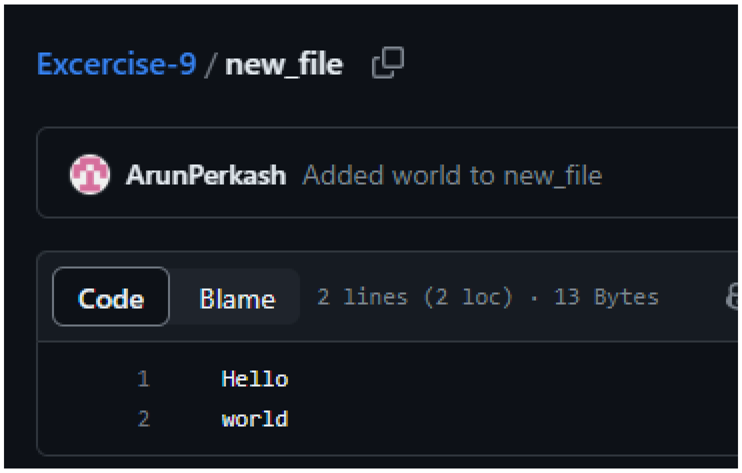
```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git branch
  feature_branch
* main

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git merge feature_branch
Updating 1634712..7d63d18
Fast-forward
 new_file | 1 +
 1 file changed, 1 insertion(+)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$
```

We go back into our main branch. We then merge the feature_branch (which is ahead of the main branch since we added “hello” word to new_file) into the main branch. Hence the main branch’s pointer is now pointed to the last commit that was done.



```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ArunPerkash/Excercise-9.git
   1634712..7d63d18  main -> main

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Finally for our changes to reflect at remote, we push the fast forward merge to the main branch.


Excercise-9 / new_file

ArunPerkash Added world to new_file

Code Blame 2 lines (2 loc) · 13 Bytes

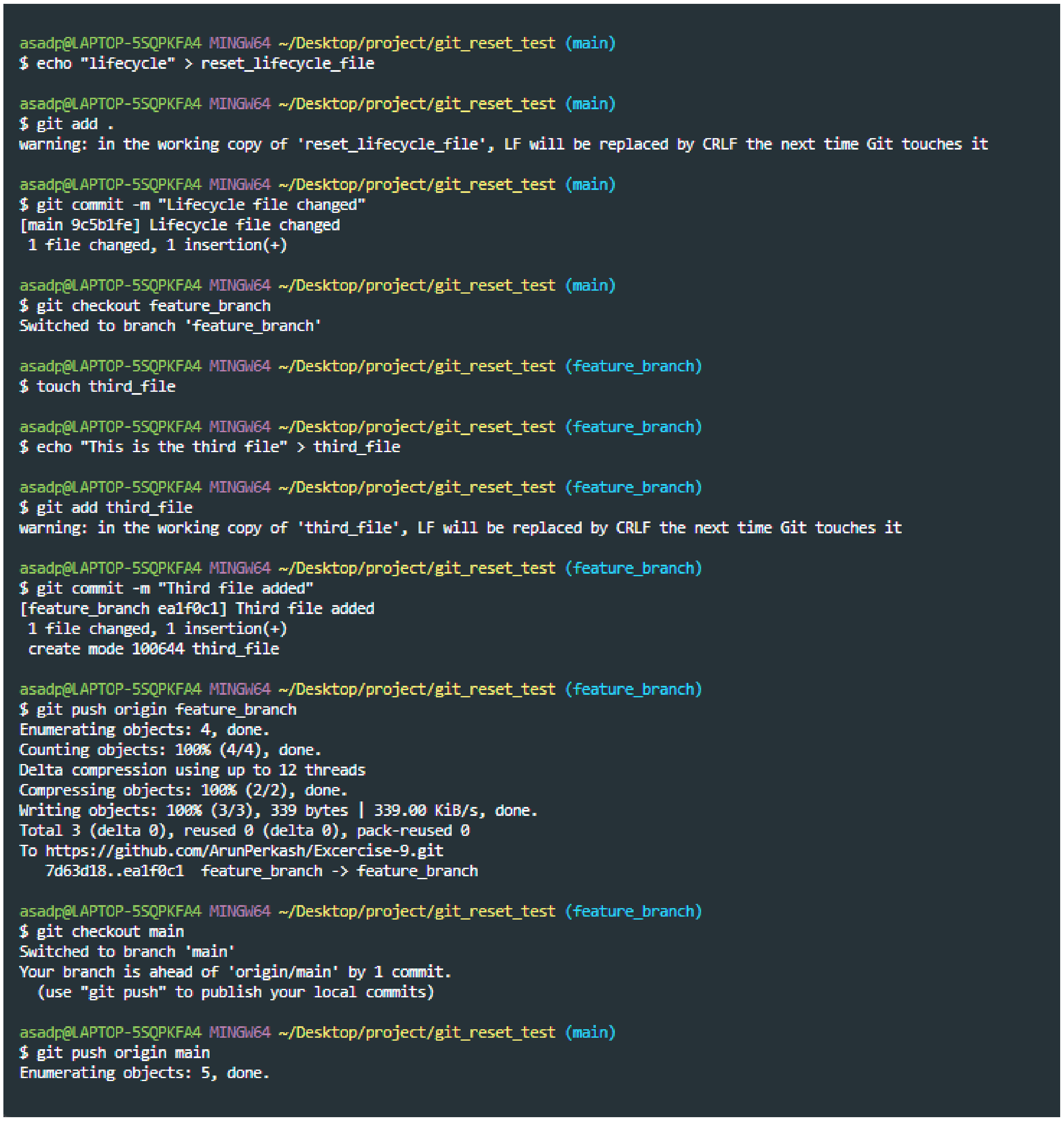
```
Hello
world
```

we can confirm the merge.

Q5

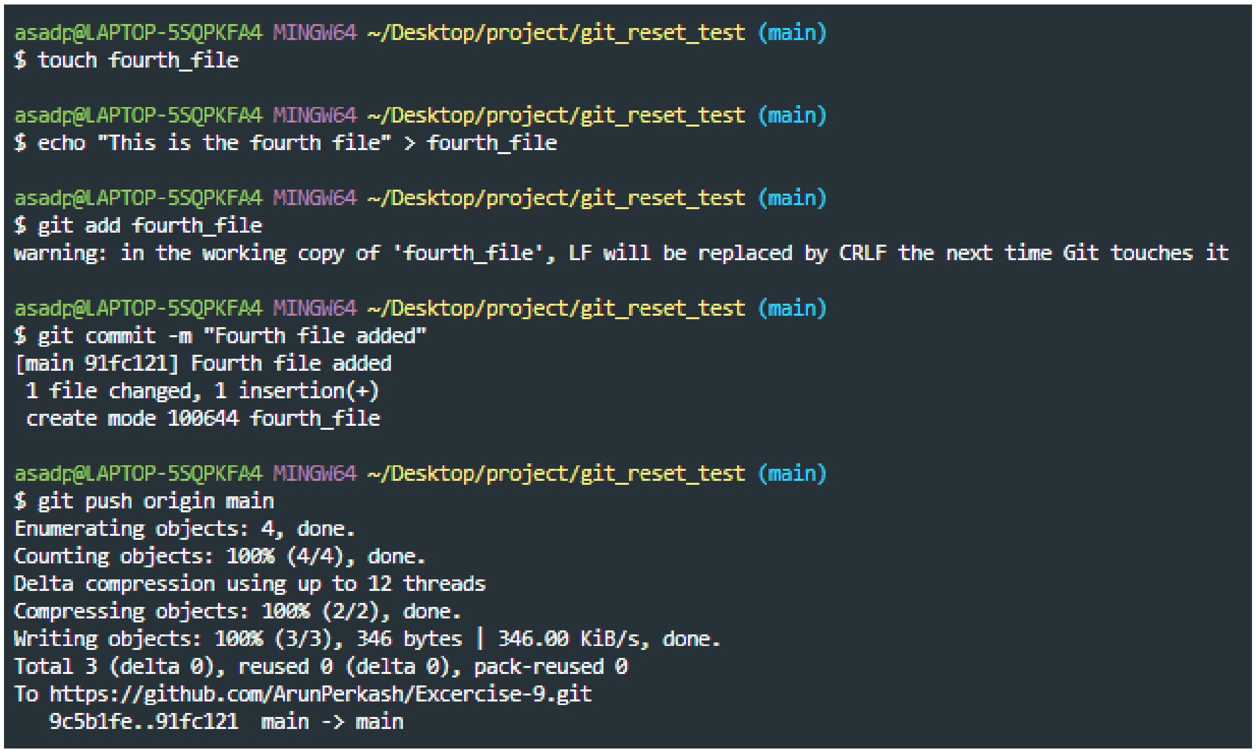
A 3-way merge is a more complex type of a merge that Git performs when there are new changes on both the source and target branches since they diverged.

Setup Scenario:



```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ echo "lifecycle" > reset_lifecycle_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git add .
warning: in the working copy of 'reset_lifecycle_file', LF will be replaced by CRLF the next time Git touches it

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git commit -m "Lifecycle file changed"
[main 9c5b1fe] Lifecycle file changed
 1 file changed, 1 insertion(+)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git checkout feature_branch
Switched to branch 'feature_branch'

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ touch third_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ echo "This is the third file" > third_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git add third_file
warning: in the working copy of 'third_file', LF will be replaced by CRLF the next time Git touches it

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git commit -m "Third file added"
[feature_branch ea1f0c1] Third file added
 1 file changed, 1 insertion(+)
 create mode 100644 third_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git push origin feature_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ArunPerkash/Excercise-9.git
   7d63d18..ea1f0c1  feature_branch -> feature_branch

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git push origin main
Enumerating objects: 5, done.
```

- We first edit the lifecycle file in main branch.
- We then stage and commit it.
- We go into the feature branch and create a new file called “Third file” . We add a text to this file as shown in the screenshot above.
- We then stage this third file we created and commit it to the branch.
- Finally we push all of the commits to the repo.

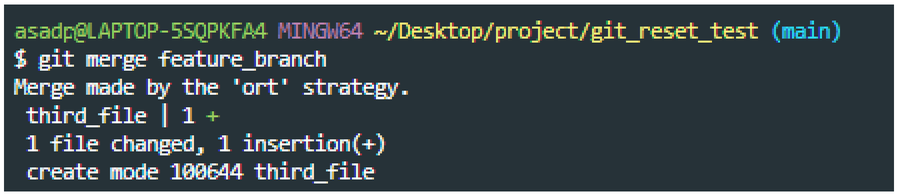

```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ touch fourth_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ echo "This is the fourth file" > fourth_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git add fourth_file
warning: in the working copy of 'fourth_file', LF will be replaced by CRLF the next time Git touches it

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git commit -m "Fourth file added"
[main 91fc121] Fourth file added
 1 file changed, 1 insertion(+)
 create mode 100644 fourth_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ArunPerkash/Excercise-9.git
   9c5b1fe..91fc121  main -> main
```

We then switch to the main branch and similarly, we create a fourth file , we stage the changes, commit it and push to our remote repository.

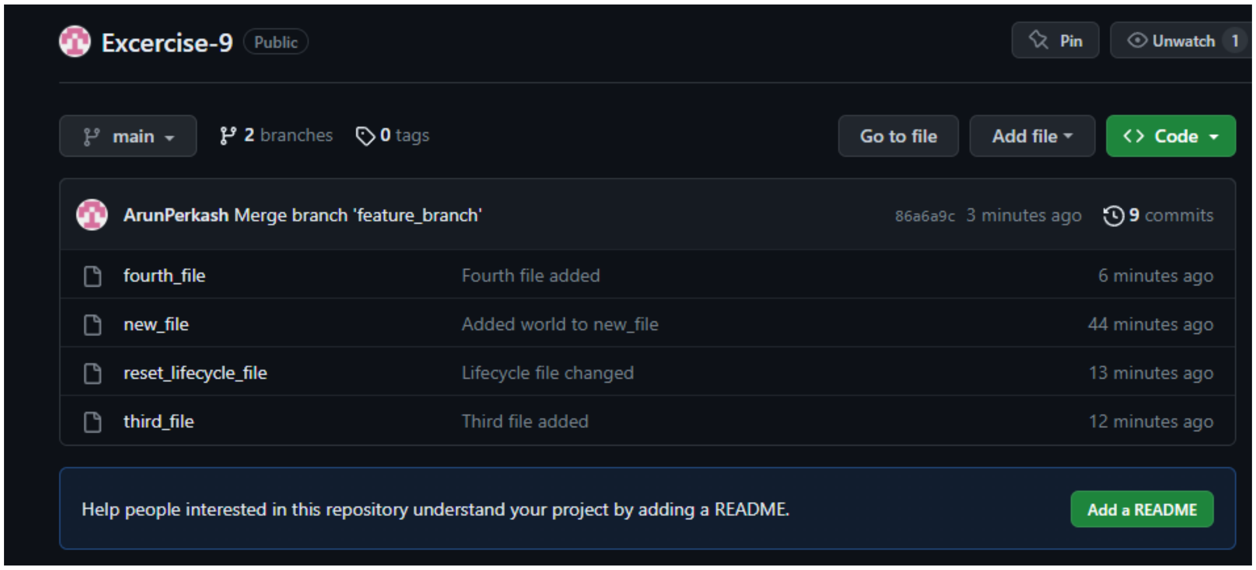
Performing the 3-Way merge

Now, we need to merge all the commits that we have made. This can be performed using git merge.

In the main branch,


```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git merge feature_branch
Merge made by the 'ort' strategy.
 third_file | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 third_file
```

We merge the feature branch. From the output, we can conclude that a successful 3-way merge has been performed to merge all the newly created files into main. We then push these changes to our remote repository.


Excercise-9 Public

main · 2 branches · 0 tags · Go to file · Add file · Code

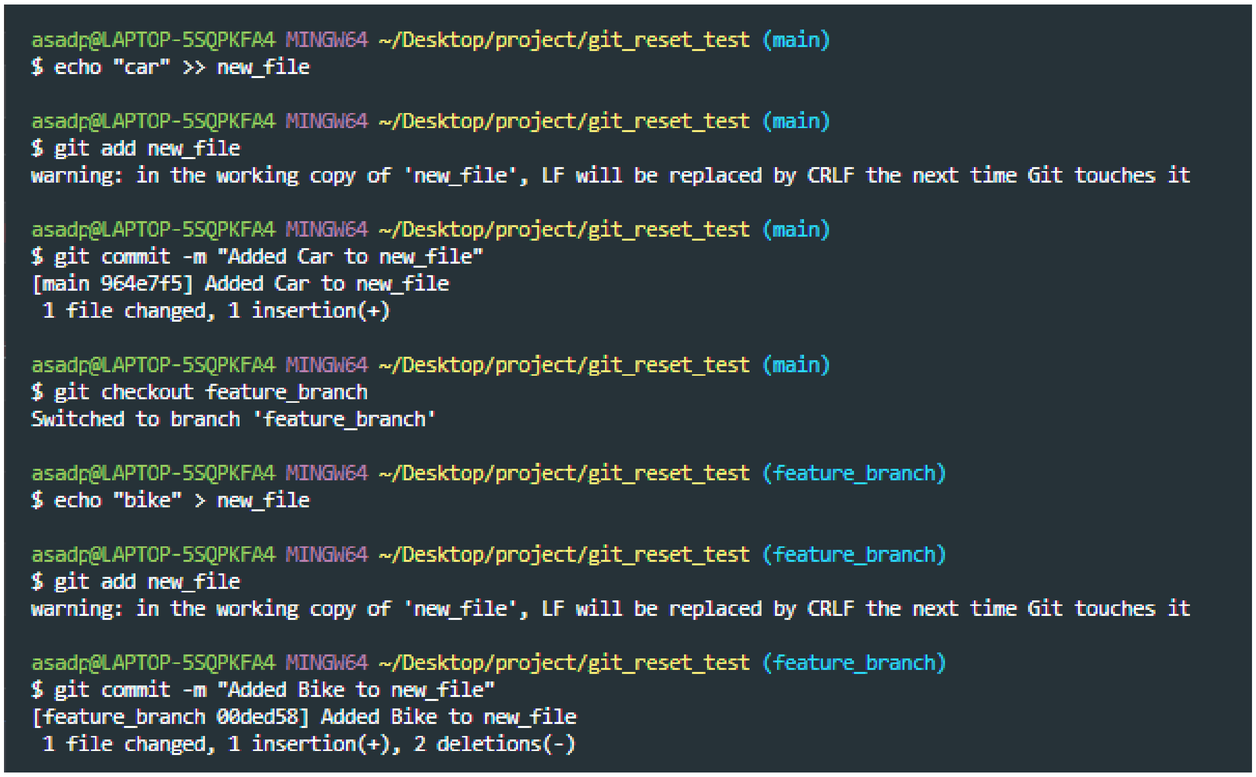
ArunPerkash Merge branch 'feature_branch' 86a6a9c 3 minutes ago 9 commits

fourth_file	Fourth file added	6 minutes ago
new_file	Added world to new_file	44 minutes ago
reset_lifecycle_file	Lifecycle file changed	13 minutes ago
third_file	Third file added	12 minutes ago

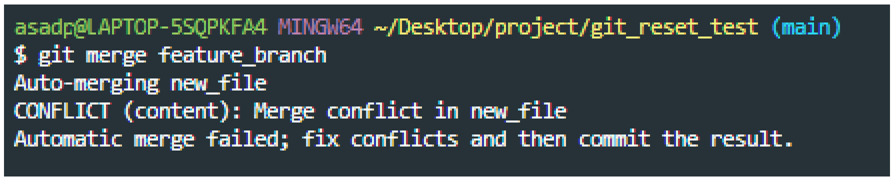
Help people interested in this repository understand your project by adding a README. Add a README

We can confirm that everything has worked perfectly. In the main branch, we have all the new files from branches that we created.

Q6

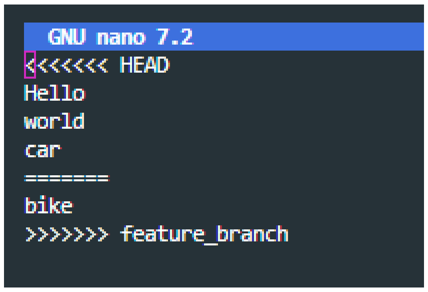

```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ echo "car" >> new_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git add new_file
warning: in the working copy of 'new_file', LF will be replaced by CRLF the next time Git touches it

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git commit -m "Added Car to new_file"
[main 964e7f5] Added Car to new_file
 1 file changed, 1 insertion(+)

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git checkout feature_branch
Switched to branch 'feature_branch'

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ echo "bike" > new_file

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git add new_file
warning: in the working copy of 'new_file', LF will be replaced by CRLF the next time Git touches it

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (feature_branch)
$ git commit -m "Added Bike to new_file"
[feature_branch 00ded58] Added Bike to new_file
 1 file changed, 1 insertion(+), 2 deletions(-)
```

In the main branch, we append a text “car” to the new_file and commit it. Whereas in the feature_branch, we append text “bike” to new_file and commit it.



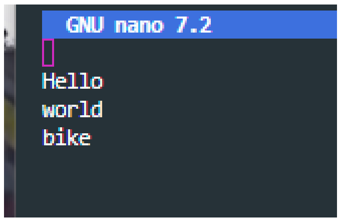
```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git merge feature_branch
Auto-merging new_file
CONFLICT (content): Merge conflict in new_file
Automatic merge failed; fix conflicts and then commit the result.
```

When we try to merge feature branch into main, we get a conflict due to 2 different lines being appended in both branches.



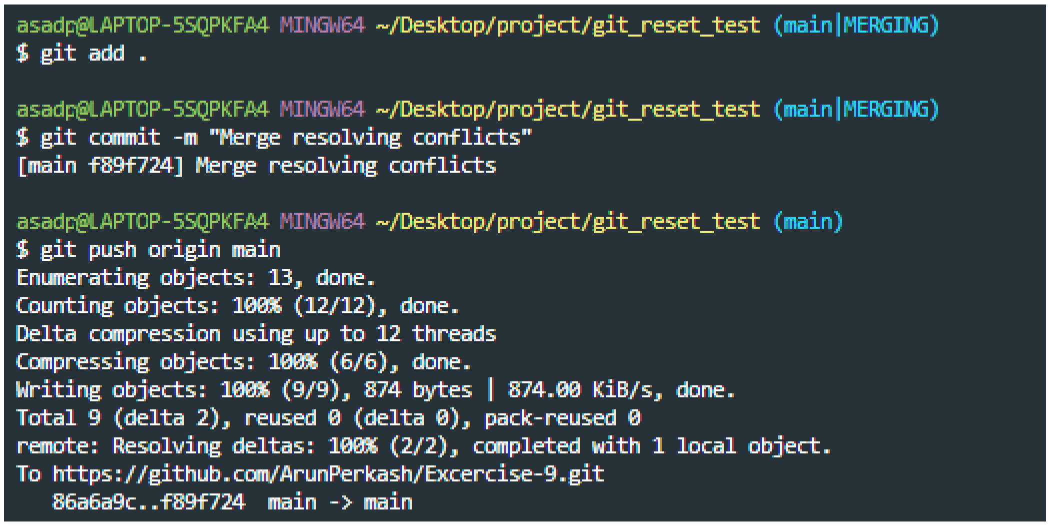
```
  GNU nano 7.2
<<<<<<< HEAD
Hello
world
car
=======
bike
>>>>>>> feature_branch
```

to



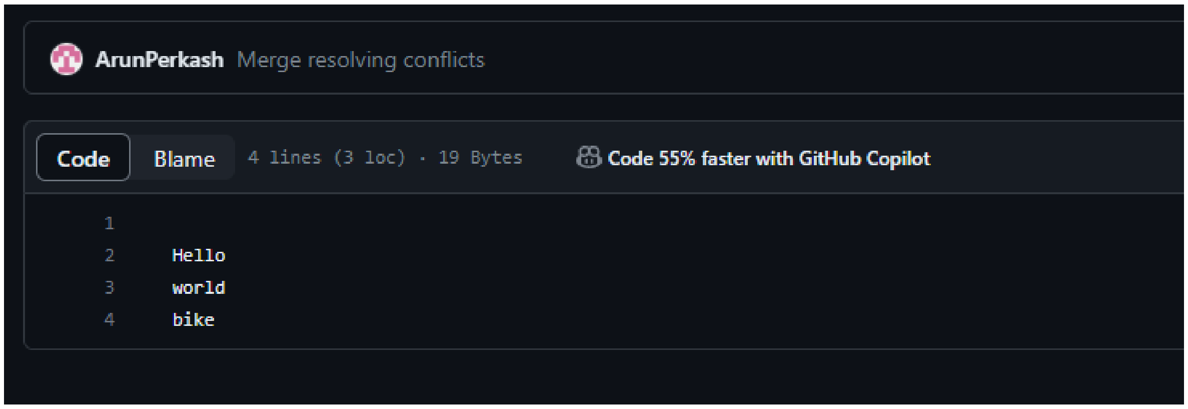
```
  GNU nano 7.2

Hello
world
bike
```

We then resolve the conflicts in a text editor. In our case, we want the “bike” instead of “car”.



```
asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main|MERGING)
$ git add .

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main|MERGING)
$ git commit -m "Merge resolving conflicts"
[main f89f724] Merge resolving conflicts

asadp@LAPTOP-5SQPKFA4 MINGW64 ~/Desktop/project/git_reset_test (main)
$ git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 874 bytes | 874.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/ArunPerkash/Excercise-9.git
   86a6a9c..f89f724  main -> main
```

We then finally push it to the remote repository.



ArunPerkash Merge resolving conflicts

Code
Blame
4 lines (3 loc) · 19 Bytes
Code 55% faster with GitHub Copilot

```

Hello
world
bike
```

We can confirm this operation by seeing the output on our github repository.

Hence we have executed and played with all the given git commands and that concludes this experiment.