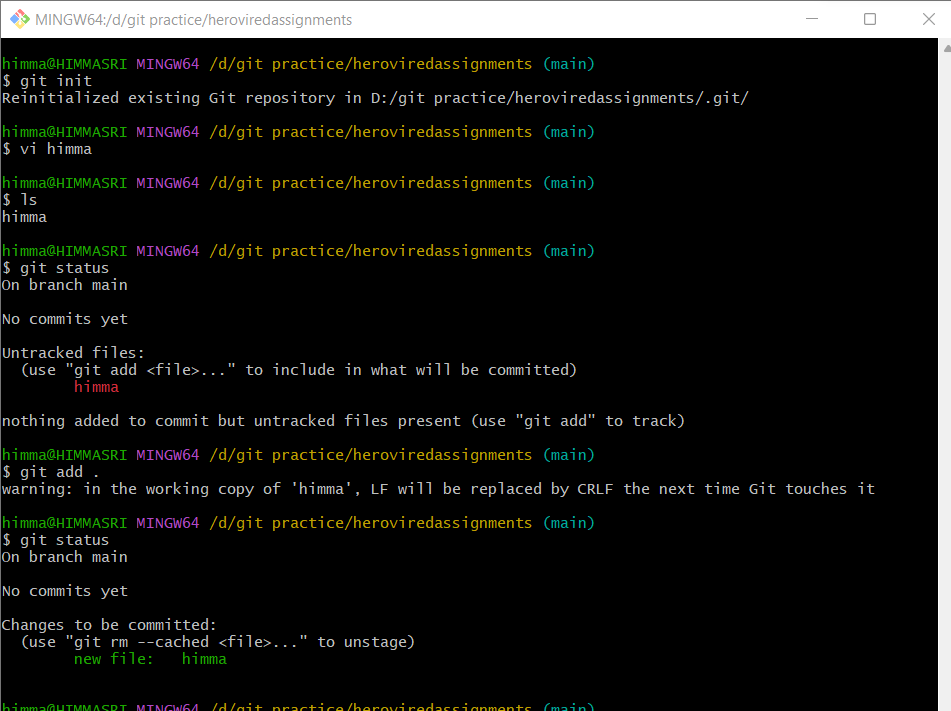
**Q1. Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.**

**Git stash** maintains the snapshot of the files. When you want to made changes to the code but you don’t want to commit those changes with the present code but you need those changes for the future reference we can use stash command to stash(save) the work aside.Whenever we need those changes we can retrieve them by referring to the index of the stash in the stash list.

We have created a file “himma” on the main branch and staged it.



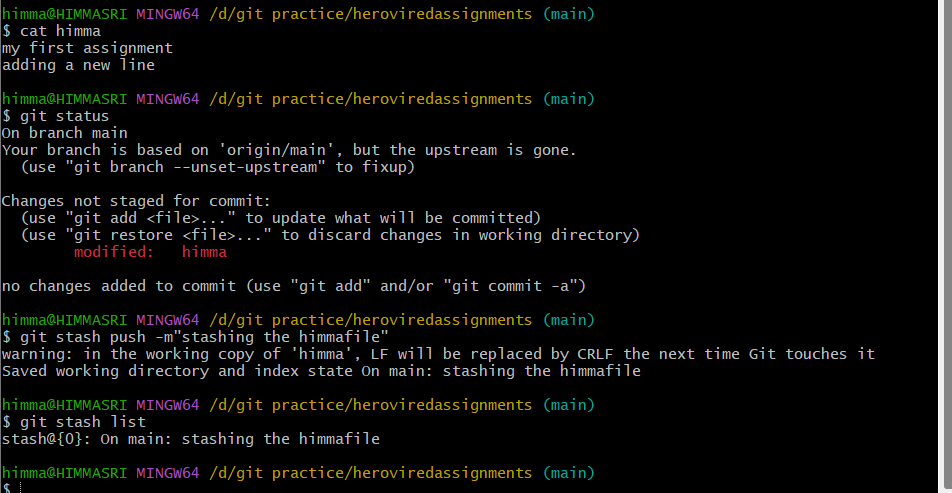
The file contains a single line statement.Now we commit this file. After committing the file we made changes to the file by adding another statement but I need to perform some operations on another branch but I don’t want to commit these changes as the file is not completed yet.

So what I do is stash these changes and move to the another branch. In that branch I will create a file and perform some operations on it and when I come back to the main branch I will retrieve the changes made to the file “himma” and keep on working on that file.

****

Now, I will add a new statement to the “himma” file and stash these changes then move to another branch without commiting in the main branch.

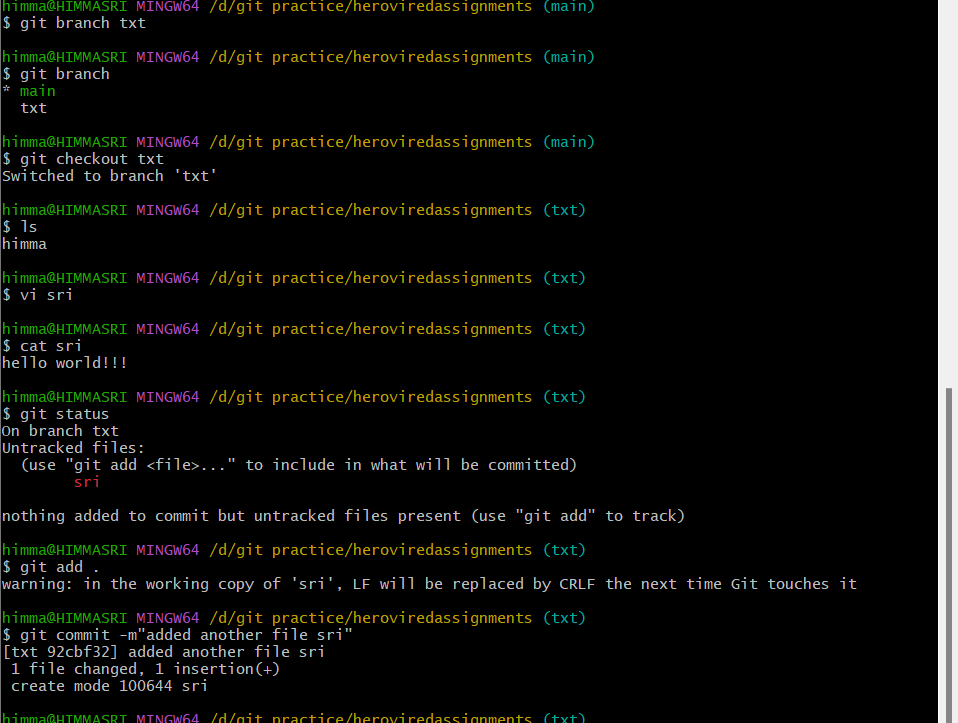
**git stash push -m “msg”** 🡪 to stash the changes.To view the stash, we can use git stash list that will the stashes performed with the index.



Now, I move to another branch called “txt” and add a file “sri” in that branch.

I will commit that file and return to the main branch.

To create a branch we use git branch “branchname”.



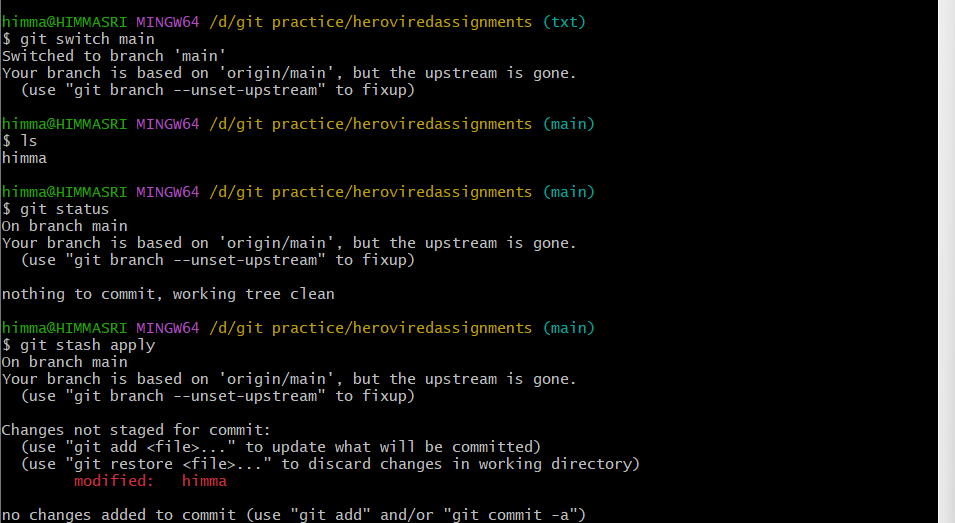
Now , I return to the main branch to keep on working with himma file.

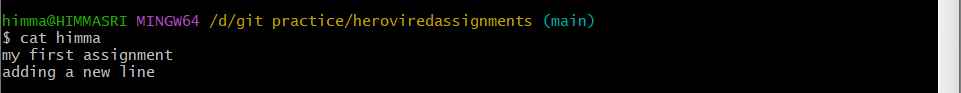
To retrieve the stash , use git stash apply command to get the recently stashed data.

To access a particular stash, we can specify the index of the stash.

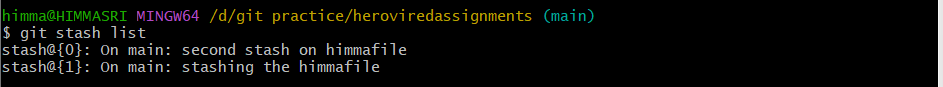
ex: **git stash apply [ -- index number ]**

After retrieving it , the new statement “adding a newline” which was not committed is back again.The status of the file is not staged as shown below.To apply these changes I can commit after some changes.





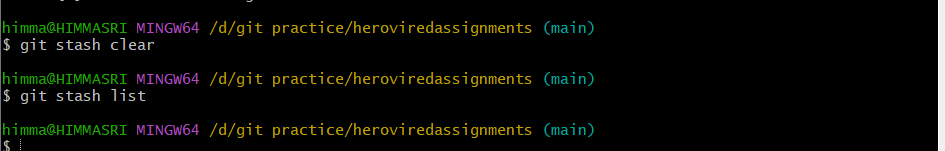
**Git stash list:** Lists the stashed performed.



**Git stash drop :** deletes a stash



**Git stash clear:** To clear all the stashes saved.



**Q2. By using a sample example of your choice, use the git fetch command and also use the git merge command and describe the whole process through a screenshot with all the commands and their output in git bash.**

**Git Fetch:**

The git fetch command is used to download the contents (commits,files) from the remote repository to the local repository.It doesn’t overwirte the existing commits in the local repository.It doesn’t integrate the any of the changes to the existing files.

We can use git fetch origin . When we run this command, a copy of our remote repository (located at “origin”) is downloaded and saved to our local machine. However, the local copy of our code has not yet been changed.

**Step 1:**Create a new repository and clone it in GitBash.

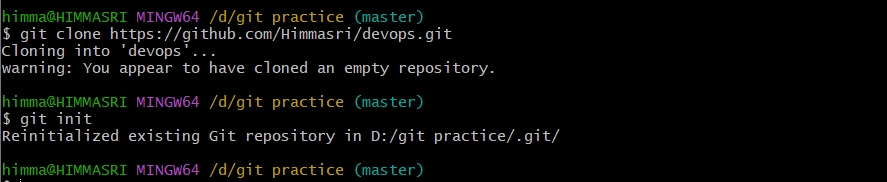
**Step 2:** Go to the Github,select that repository.

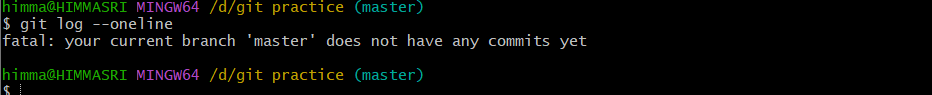
**Step 3:** Add / modify the repository .

**Step 4:** Use fetch command and fetch those changes to the local repository.

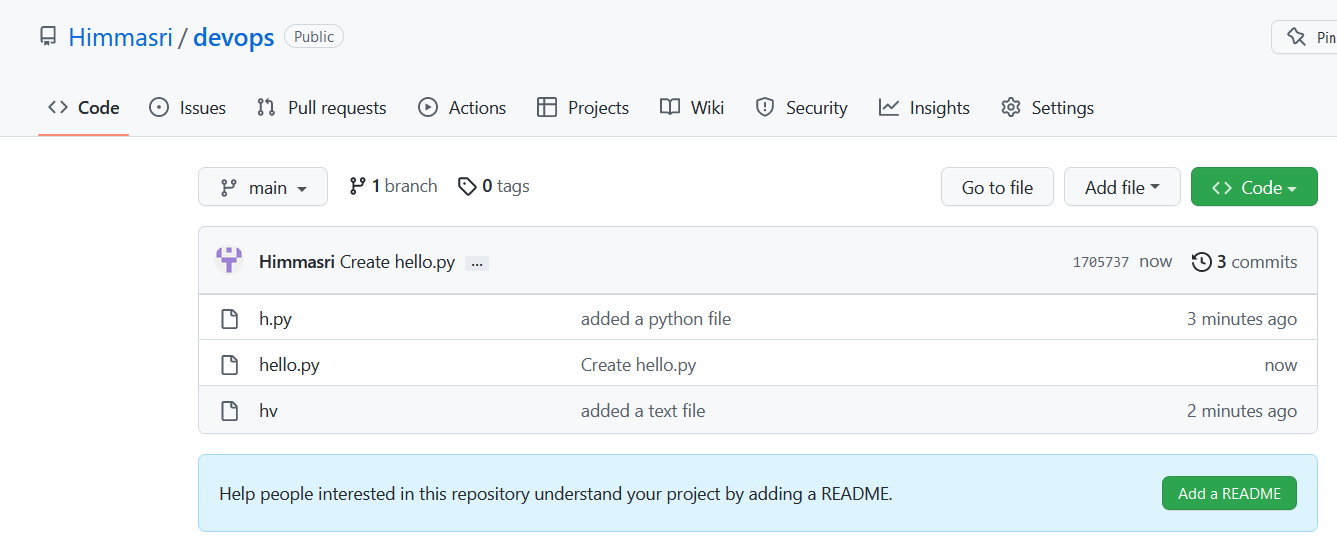
**Step 5:** We can verify by using git log command to view the fetched data.

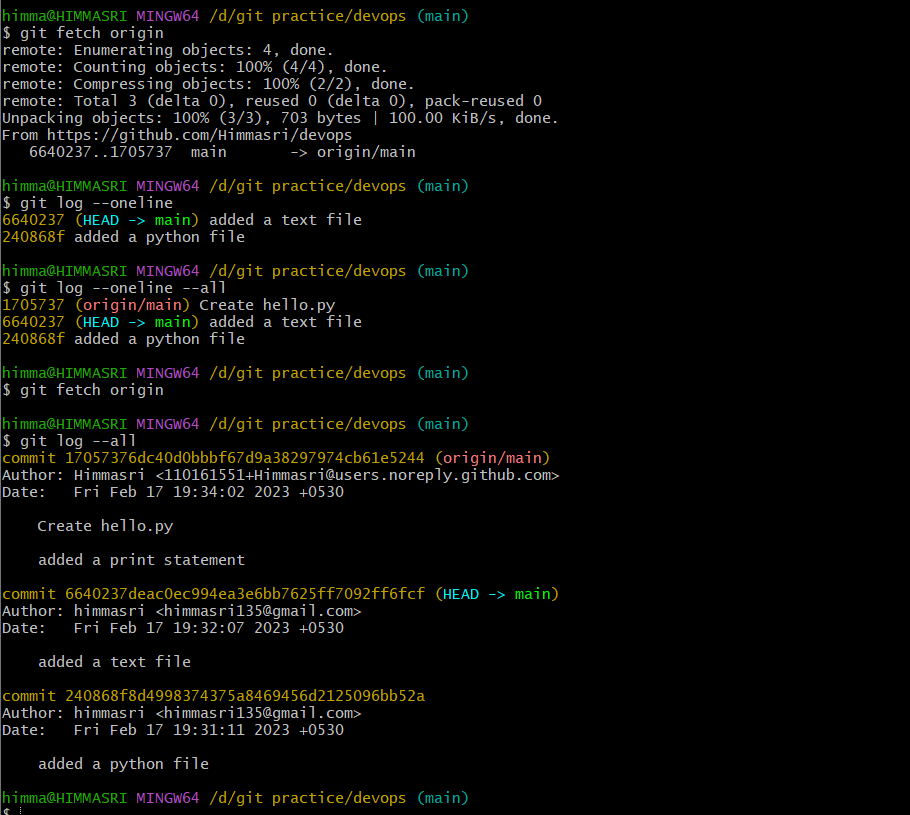
Since fetch downloads the files the changes are not applied to the local repository to do that, we can use git pull command.





Using git fetch to bring these changes to the local repository.



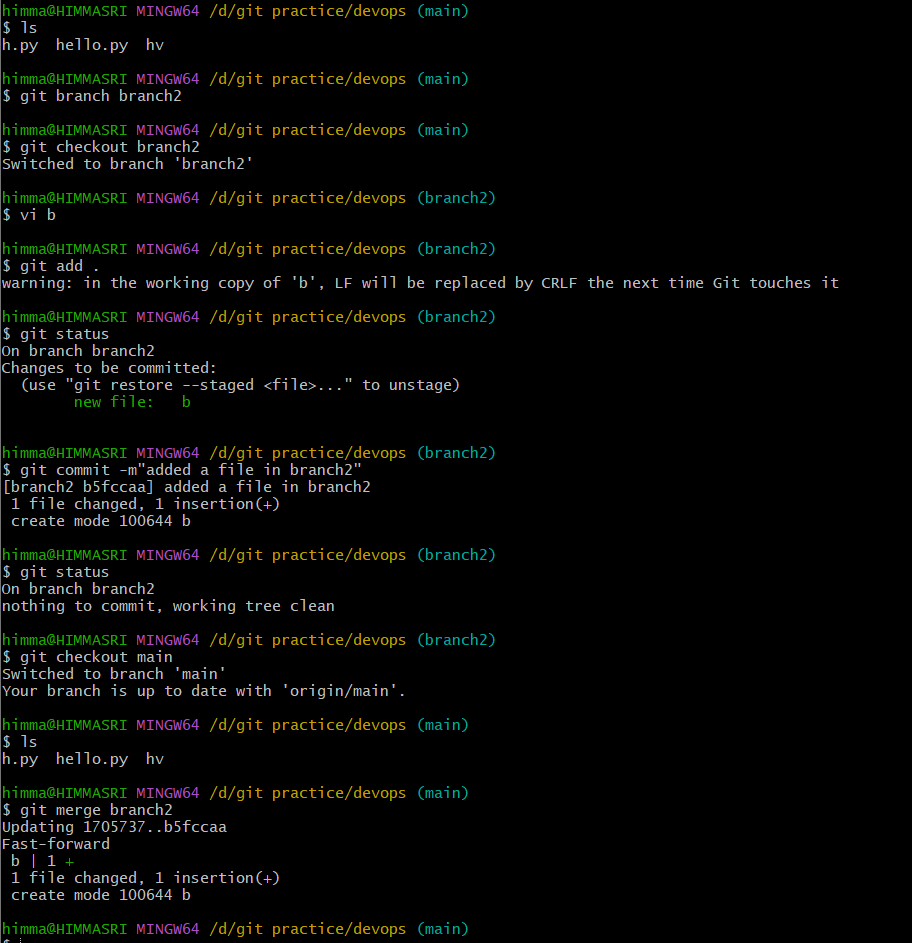


**Git Merge:**

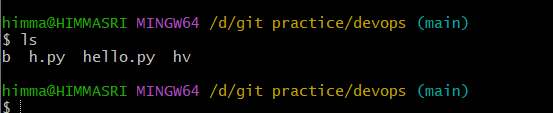
Git merge is used to merge(combine) and integrate multiple branches into a single branch.

It will combine the multiple commits in different branches into a single branch.

**Git merge branchname** 🡪 will merge the specified branch to the main.



Successfully, merged the branch2 to the main , when we list the files the file in branch2 will be displayed.

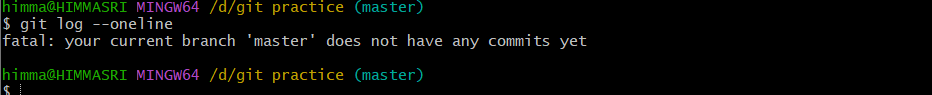


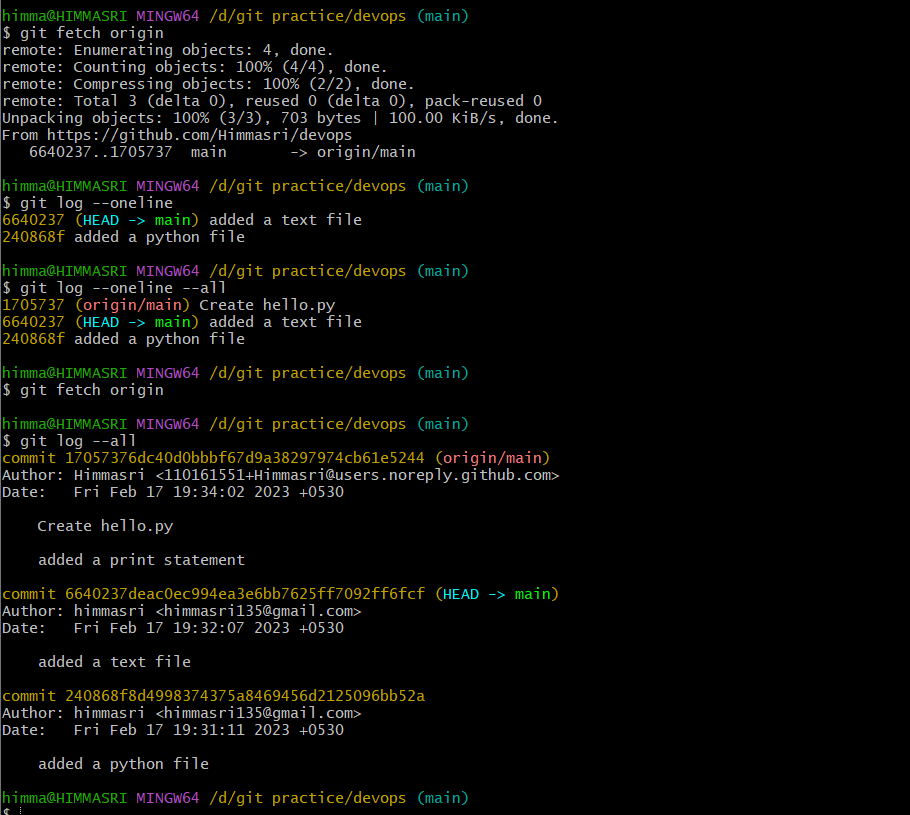
**Q3. State the difference between git fetch and git pull by doing a practical example in your git bash and attach a screenshot of all the processes.**

**Git Fetch:**

The git fetch command is used to download the contents (commits,files) from the remote repository to the local repository.It doesn’t overwirte the existing commits in the local repository.It doesn’t integrate the any of the changes to the existing files.

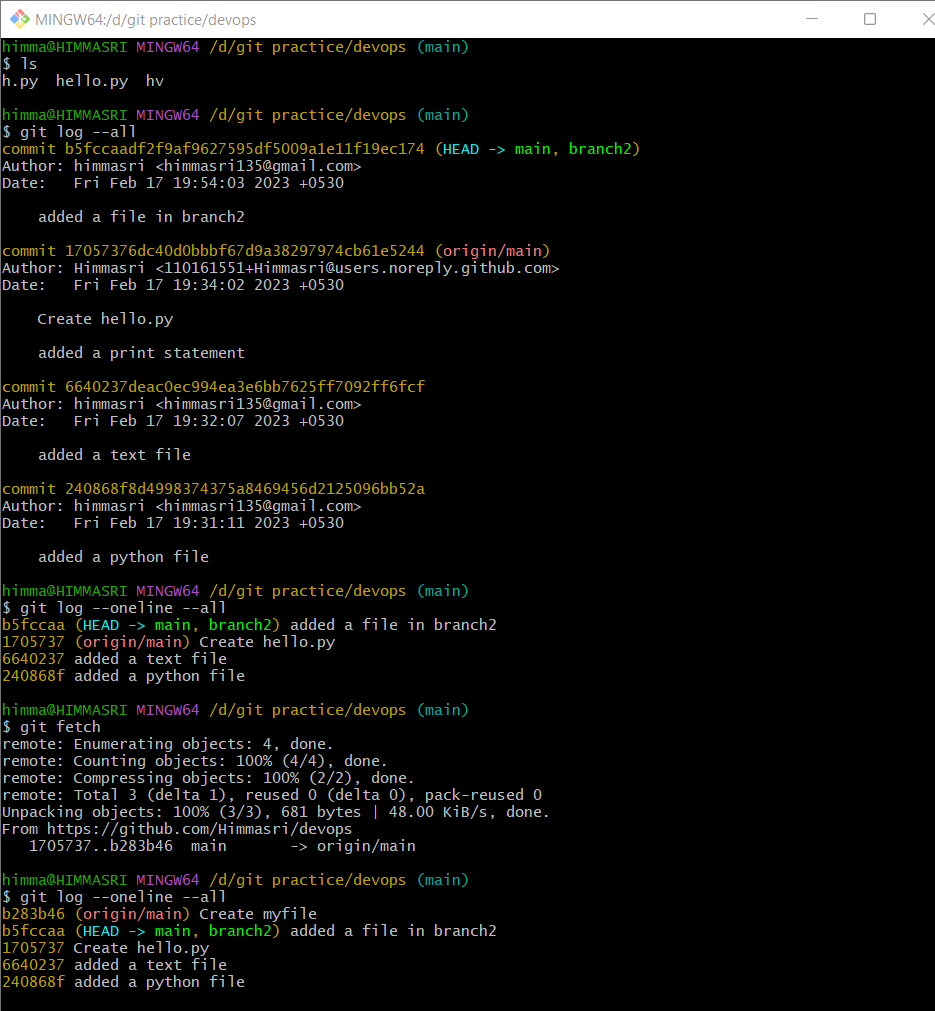
We can use **git fetch origin** . When we run this command, a copy of our remote repository (located at “origin”) is downloaded and saved to our local machine. However, the local copy of our code has not yet been changed.





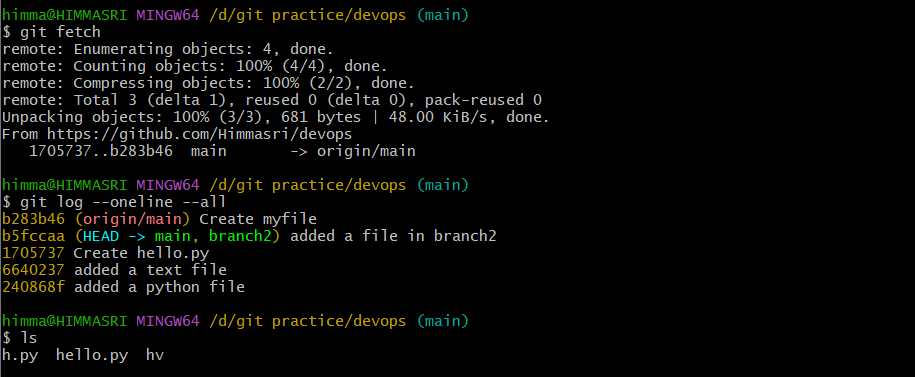
**Git Pull:**

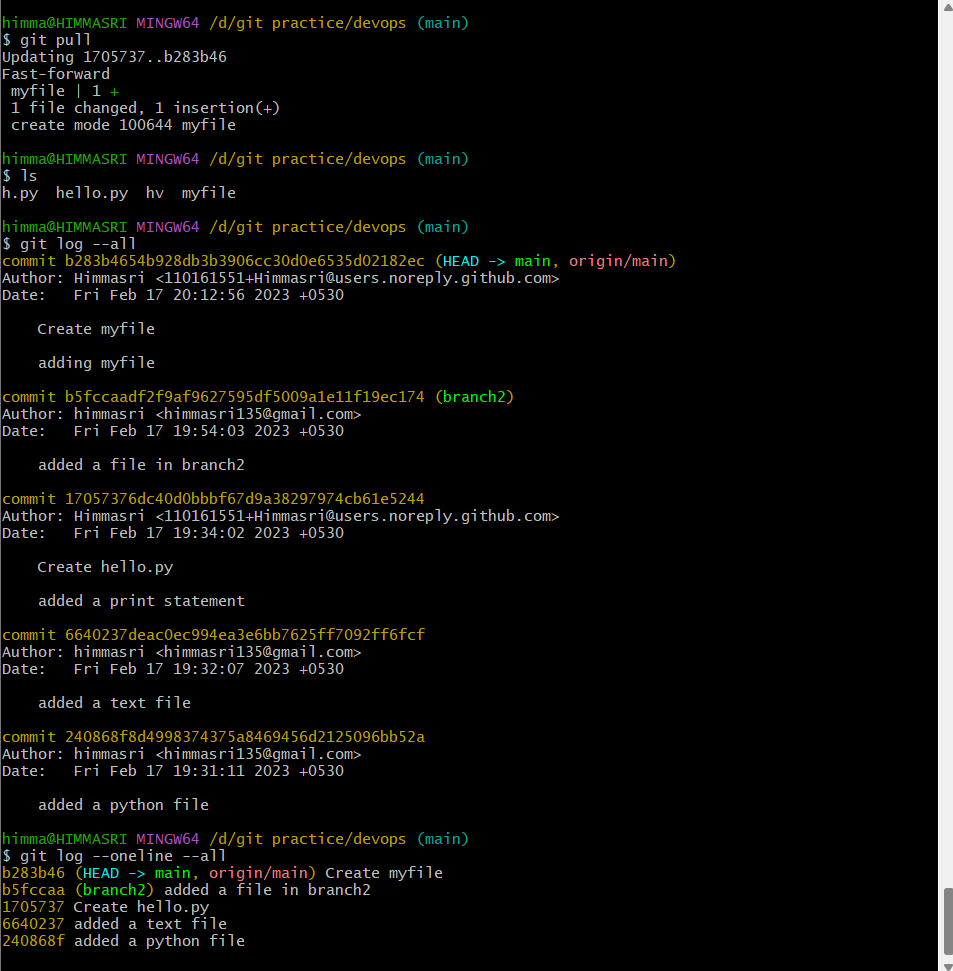
The pull command is used to access the commits from a remote repository to the local repository.Git fetch will only download the commits to the local repository.But git pull command will download and apply the changes to the local repository.It performs both fetch and merge.



The difference between git fetch and git pull is shown here.Even after performing git fetch the newly created file “myfile” doesn’t appear when we try to list the files using ls command.

But after performing pull command we can see that the “myfile” is listed.





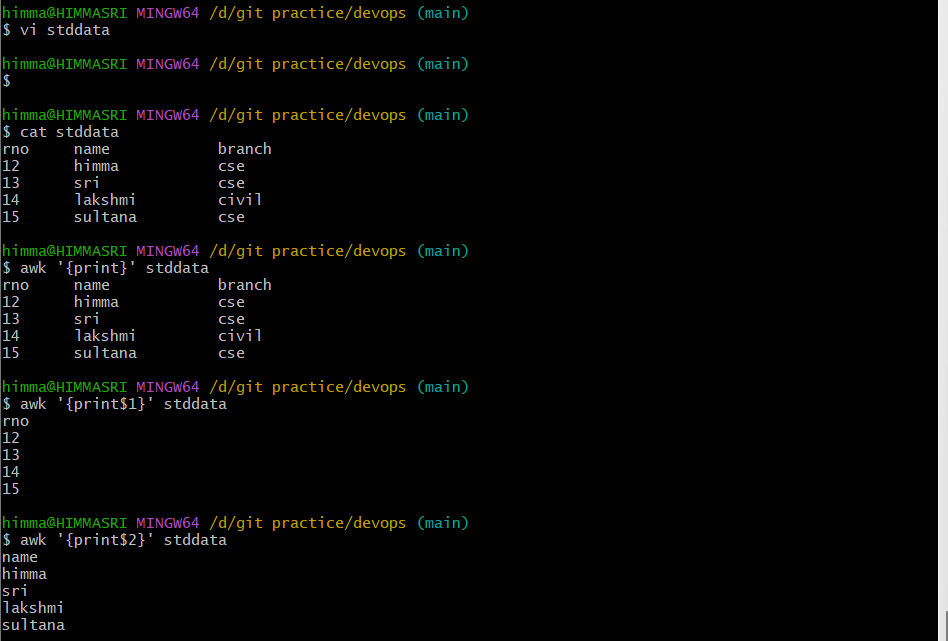
**Q4. Try to find out about the awk command and use it while reading a file created by yourself. Also, make a bash script file and try to find out the prime number from the range 1 to 20. The whole process should be carried out and by using the history command, give the screenshot of all the processes being carried out.**

**AWK command:**

The **awk** command's main purpose is to make **information retrieval and text manipulation** easy to perform in Linux.This command works by scanning a set of input lines and searches for lines matching the patterns specified by the user.

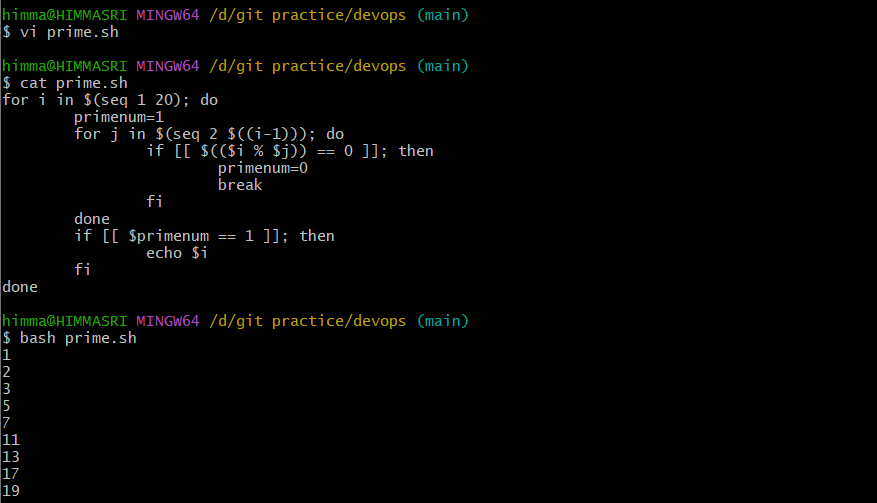
**awk ‘{print}’ filename** 🡪 will print the data in the file.

**awk ‘{print$columnnumber}’ filename** 🡪 will print the data in that specified column.



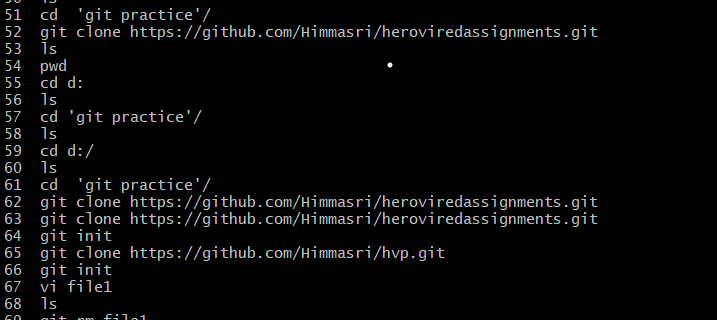
**Bash Scripting:**

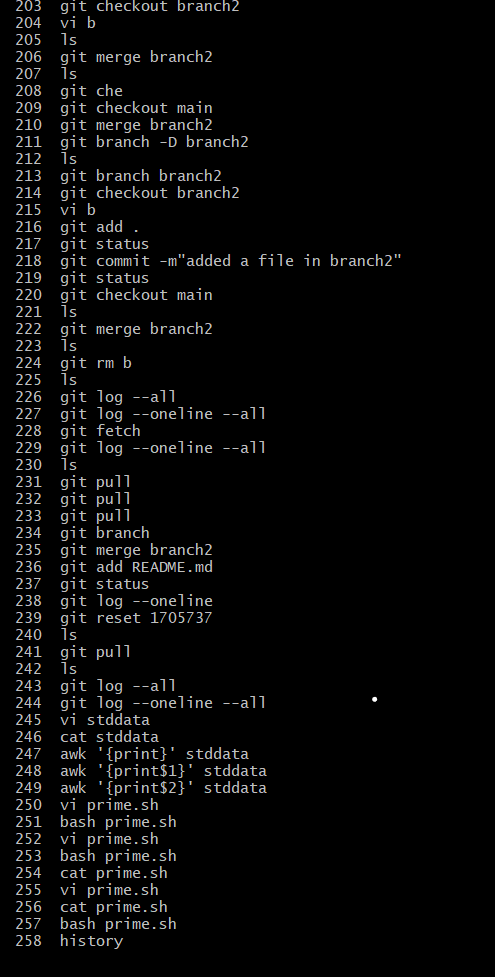
Program to print prime numbers between 1 to 20:



**History:**

History command lists all the previously executed commands.





**Q5. Set up a container and run a Ubuntu operating system. For this purpose, you can make use of the docker hub and run the container in interactive mode.**

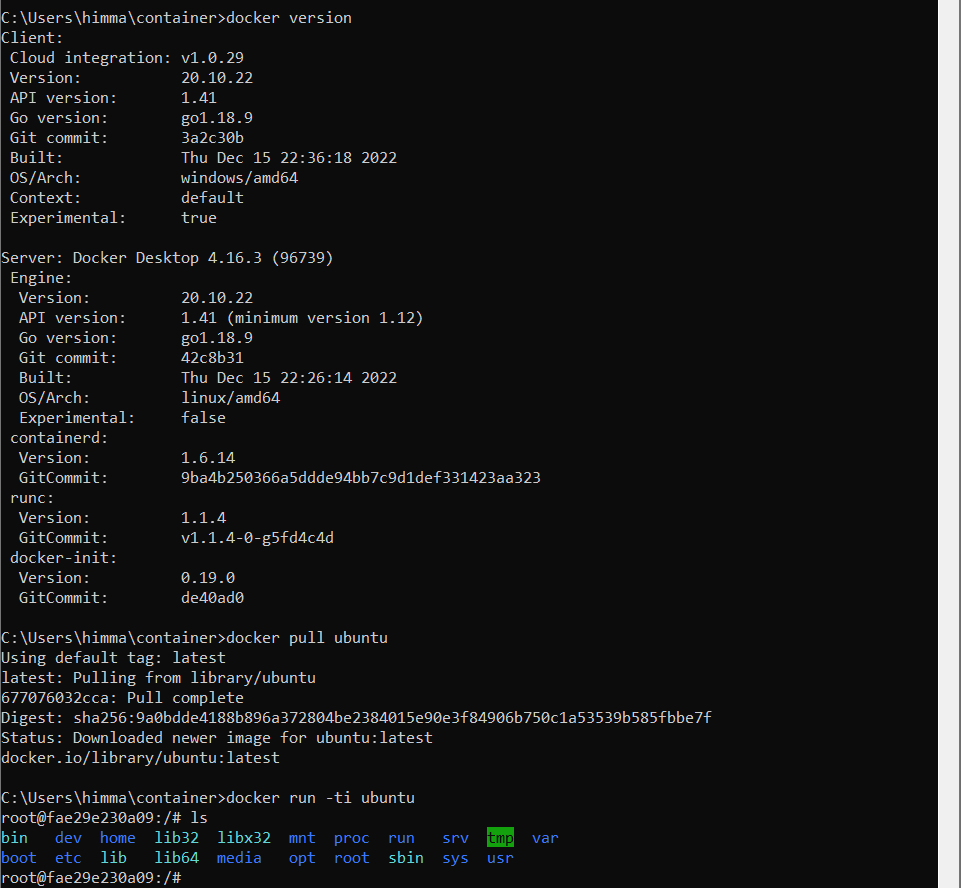
**CONTAINER** 🡪 The containers virtualize the operating system rather than virtualizing the hidden hardware so that each particular container can include both the libraries and dependencies of the corresponding application.The containers are fast, portable, and lightweight.

To set up a container we need to use :

**Docker pull:** This command will download the image from which the container is built.

**Docker run:** This command will run the container if the image already exists it will use it otherwise it will download and runs it.

To run the container in interactive mode use docker run -ti .



Github link: **https://github.com/Himmasri**