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

the stash's meaning is "store something safely in a hidden place."

The sense in Git is also the same for stash; Git temporarily saves your data safely without committing.

-It uses stack data structure which follows LIFO(Last In First Out) order.

-Following are stash commands:

$git stash

-Used to stash the change.

$ git stash list

-List out all the stashes.

$ git stash show

-Shows the changes made.

$ git stash apply [--index]

-Apply the stash with given index.

$ git stash pop [--index]

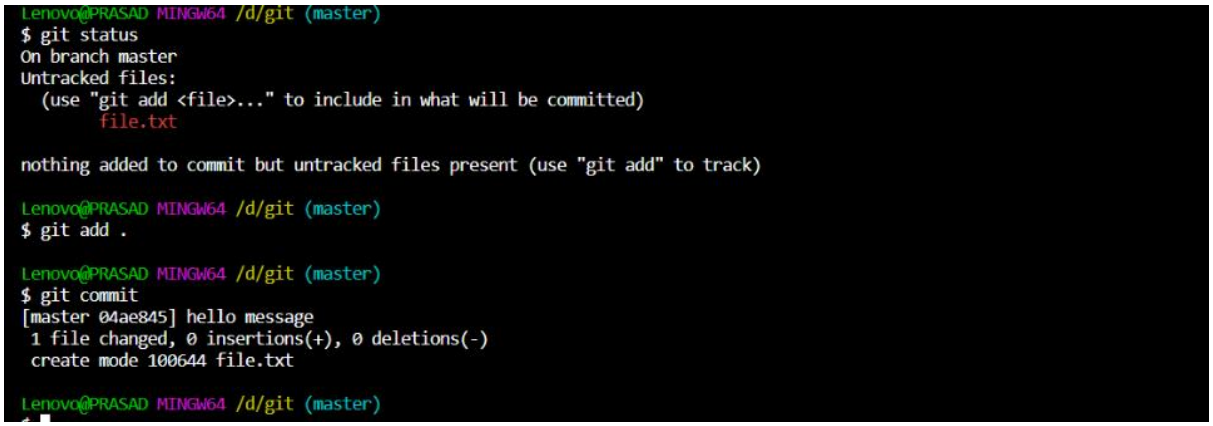
-Apply the top stash if index is not specified.

$ git stash drop

-Drops a stash

$ git stash clear

-Clears the stash



```
Lenovo@PRASAD MINGW64 /d/git (master)
$ git add .

Lenovo@PRASAD MINGW64 /d/git (master)
$ git commit
[master 04ae845] hello message
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt

Lenovo@PRASAD MINGW64 /d/git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
```

```
$ git commit
[master 04ae845] hello message
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt

Lenovo@PRASAD MINGW64 /d/git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash
Saved working directory and index state WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$
```

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash
Saved working directory and index state WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash lsit
fatal: subcommand wasn't specified; 'push' can't be assumed due to unexpected token 'lsit'

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash list
stash@{0}: WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
```

```
Saved working directory and index state WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash lsit
fatal: subcommand wasn't specified; 'push' can't be assumed due to unexpected token 'lsit'

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash list
stash@{0}: WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash
Saved working directory and index state WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash list
stash@{0}: WIP on master: 04ae845 hello message
stash@{1}: WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$
```

```
$ git stash list
stash@{0}: WIP on master: 04ae845 hello message
stash@{1}: WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$ git apply



Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash apply
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
```

```
Lenovo@PRASAD MINGW64 /d/git (master)
$ git add .

Lenovo@PRASAD MINGW64 /d/git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file.txt


Lenovo@PRASAD MINGW64 /d/git (master)
$ git commit
[master 45692f4] stashed 1 commit
 1 file changed, 2 insertions(+)

Lenovo@PRASAD MINGW64 /d/git (master)
$ git log --oneline
45692f4 (HEAD -> master) stashed 1 commit
04ae845 hello message
3fbf393 1stash
```

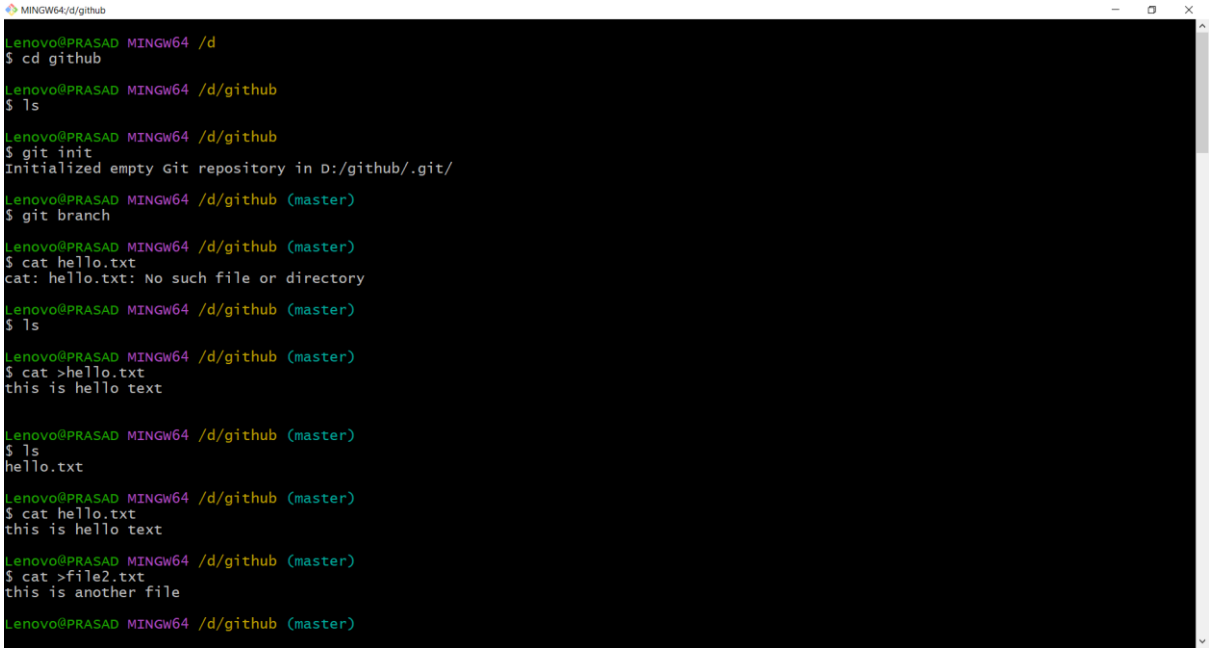
```
$ git commit
[master 45692f4] stashed 1 commit
 1 file changed, 2 insertions(+)

Lenovo@PRASAD MINGW64 /d/git (master)
$ git log --oneline
45692f4 (HEAD -> master) stashed 1 commit
04ae845 hello message
3fbf393 1stash
4653bb7 hello

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash drop
Dropped refs/stash@{0} (bd941c6d6e7efa4fe223acb3ae3a2d93c935fcda)

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash list
stash@{0}: WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$
```

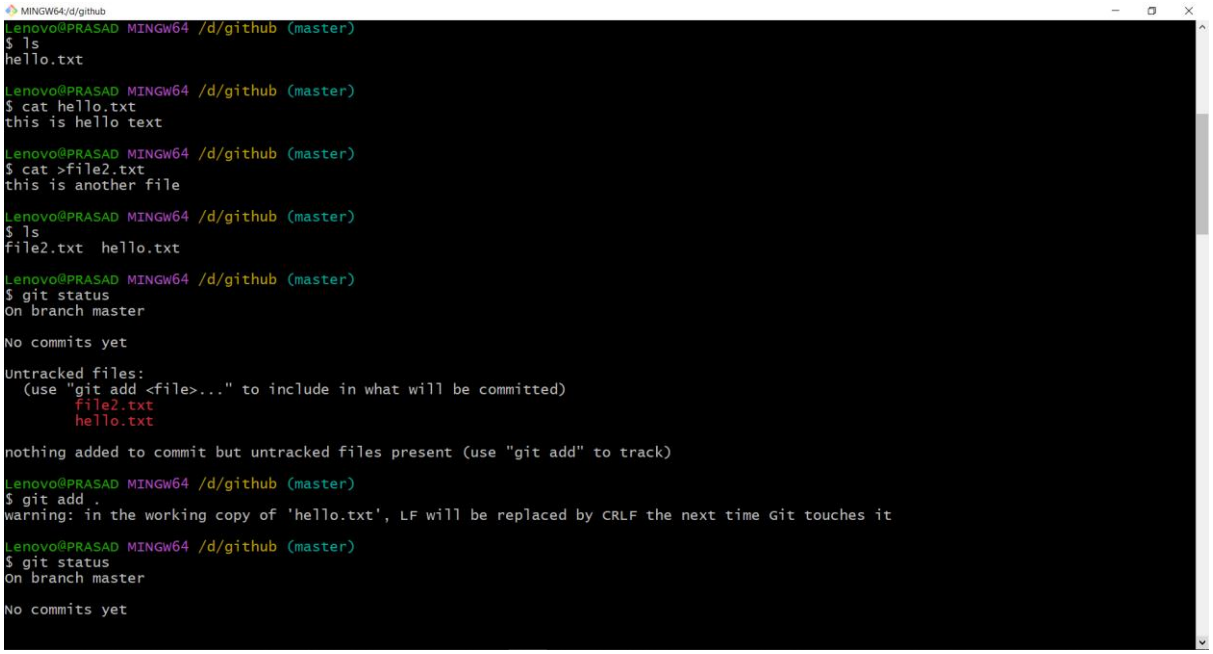
```
$ git stash drop
Dropped refs/stash@{0} (bd941c6d6e7efa4fe223acb3ae3a2d93c935fcda)

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash list
stash@{0}: WIP on master: 04ae845 hello message

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash apply
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
On branch master
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
$
```

```
$ git stash apply
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
On branch master
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
$ git add .

Lenovo@PRASAD MINGW64 /d/git (master)
$ git commit
[master 8603256] second stash
 1 file changed, 6 insertions(+), 1 deletion(-)

Lenovo@PRASAD MINGW64 /d/git (master)
$
```

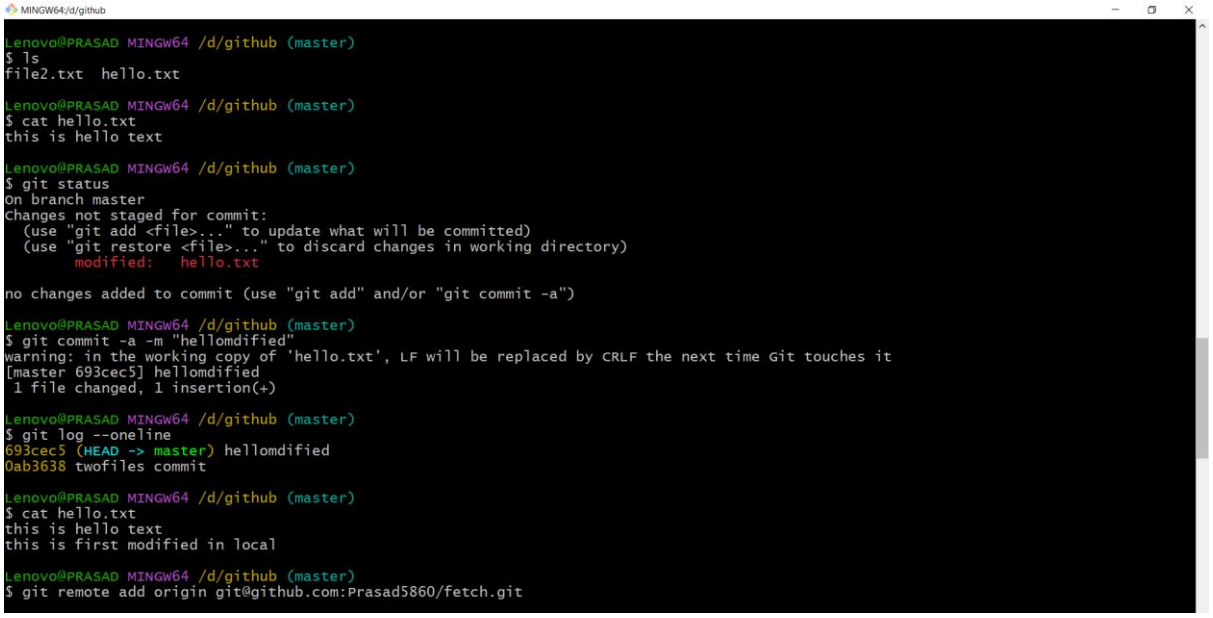
```
        both modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/git (master)
$ git add .

Lenovo@PRASAD MINGW64 /d/git (master)
$ git commit
[master 8603256] second stash
 1 file changed, 6 insertions(+), 1 deletion(-)

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash drop
Dropped refs/stash@{0} (4419b6e86d11cd44fdda23b6924ecad2e168e954)

Lenovo@PRASAD MINGW64 /d/git (master)
$ git stash list

Lenovo@PRASAD MINGW64 /d/git (master)
$
```

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.

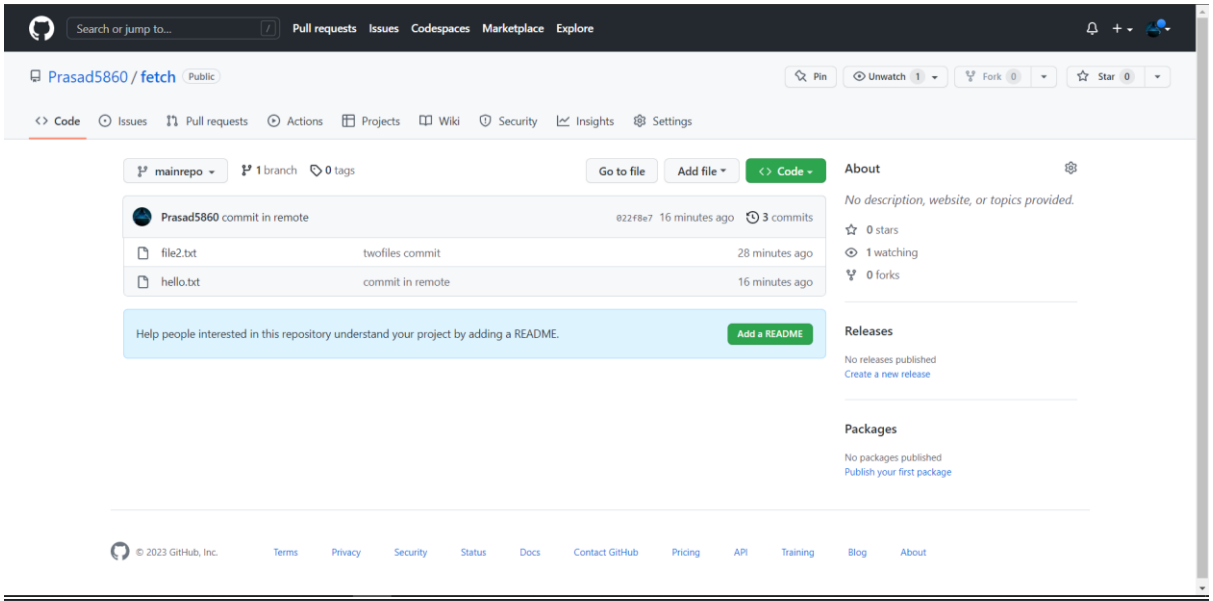


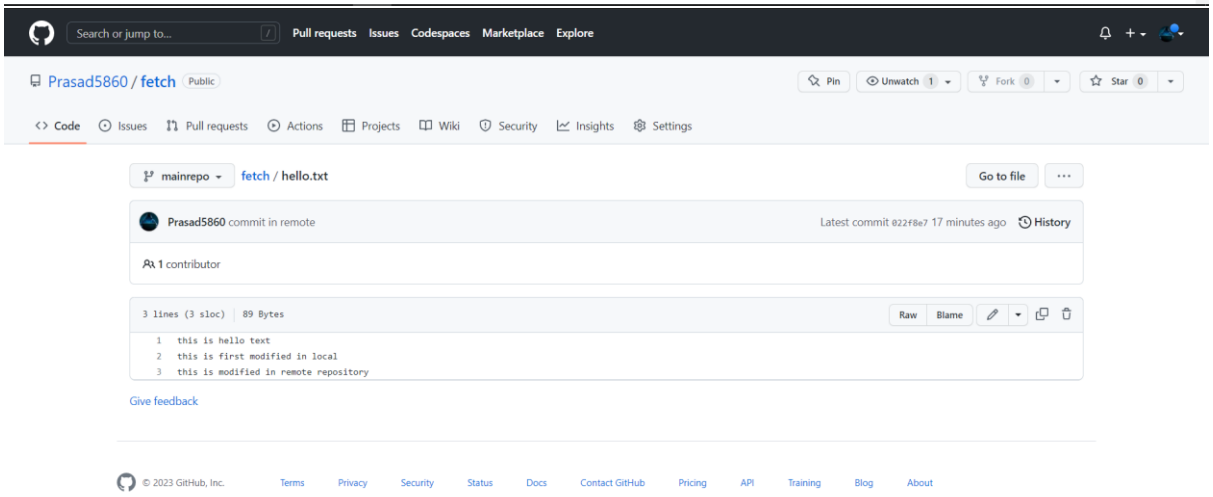


MINGW64:/d/github
Lenovo@PRASAD MINGW64 /d/github (master)
$ git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: file2.txt
new file: hello.txt
Lenovo@PRASAD MINGW64 /d/github (master)
$ cat hello.txt
this is hello text
Lenovo@PRASAD MINGW64 /d/github (master)
$ git commit -m "twofiles commit"
[master (root-commit) 0ab3638] twofiles commit
2 files changed, 1 insertion(+)
create mode 100644 file2.txt
create mode 100644 hello.txt
Lenovo@PRASAD MINGW64 /d/github (master)
$ git status
On branch master
nothing to commit, working tree clean
Lenovo@PRASAD MINGW64 /d/github (master)
$ git log --oneline
0ab3638 (HEAD -> master) twofiles commit
Lenovo@PRASAD MINGW64 /d/github (master)
$ cat >>hello.txt
this first modified line
Lenovo@PRASAD MINGW64 /d/github (master)
$ git status
On branch master


MINGW64:/d/github
Lenovo@PRASAD MINGW64 /d/github (master)
$ ls
file2.txt hello.txt
Lenovo@PRASAD MINGW64 /d/github (master)
$ cat hello.txt
this is hello text
Lenovo@PRASAD MINGW64 /d/github (master)
$ git status
On branch master
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: hello.txt
no changes added to commit (use "git add" and/or "git commit -a")
Lenovo@PRASAD MINGW64 /d/github (master)
$ git commit -a -m "hellomdified"
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it
[master 693cec5] hellomdified
1 file changed, 1 insertion(+)
Lenovo@PRASAD MINGW64 /d/github (master)
$ git log --oneline
693cec5 (HEAD -> master) hellomdified
0ab3638 twofiles commit
Lenovo@PRASAD MINGW64 /d/github (master)
$ cat hello.txt
this is hello text
this is first modified in local
Lenovo@PRASAD MINGW64 /d/github (master)
$ git remote add origin git@github.com:Prasad5860/fetch.git

MINGW64:/d/github
Lenovo@PRASAD MINGW64 /d/github (master)
$ git branch -m mainrepo
Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git push -u origin mainrepo
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 545 bytes | 272.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Prasad5860/fetch.git
* [new branch] mainrepo -> mainrepo
branch 'mainrepo' set up to track 'origin/mainrepo'.

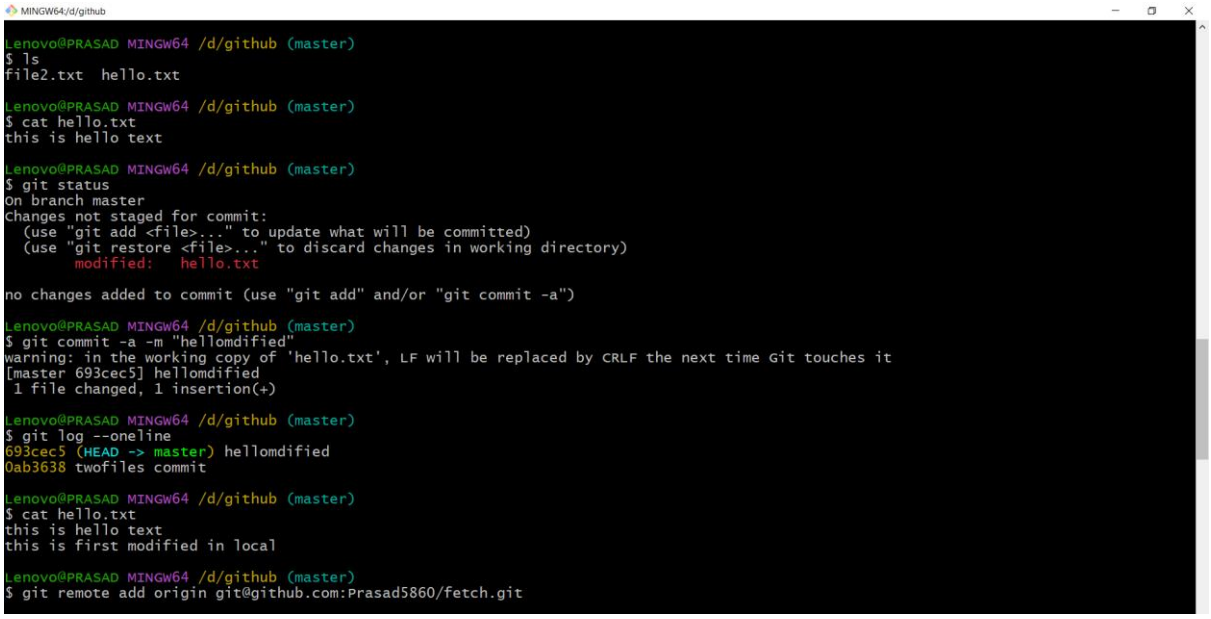


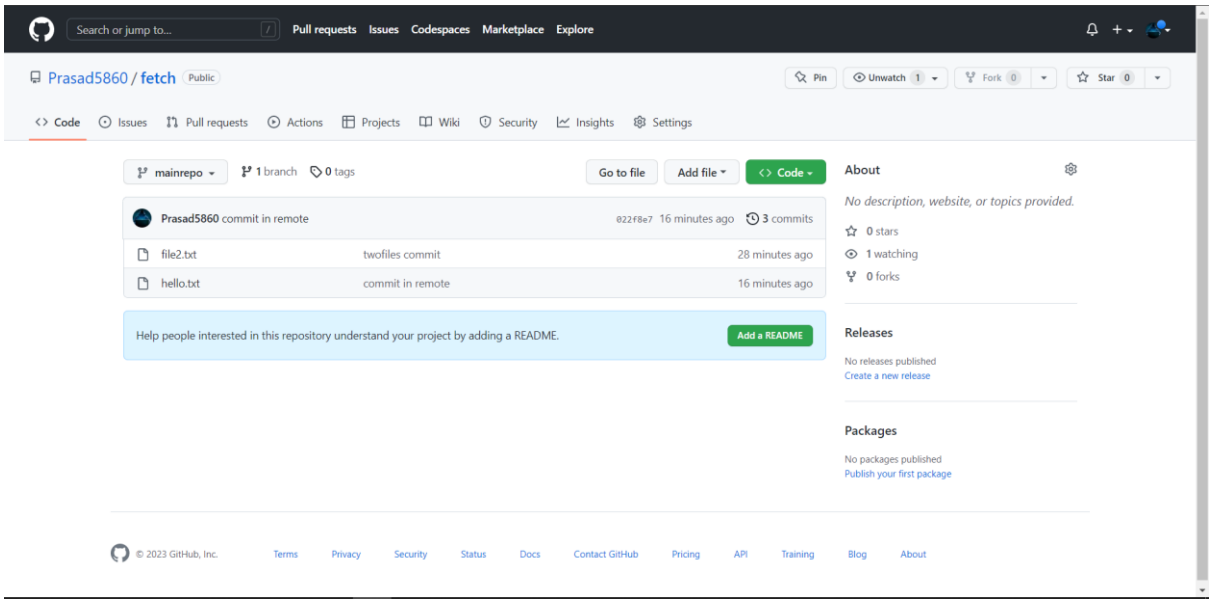


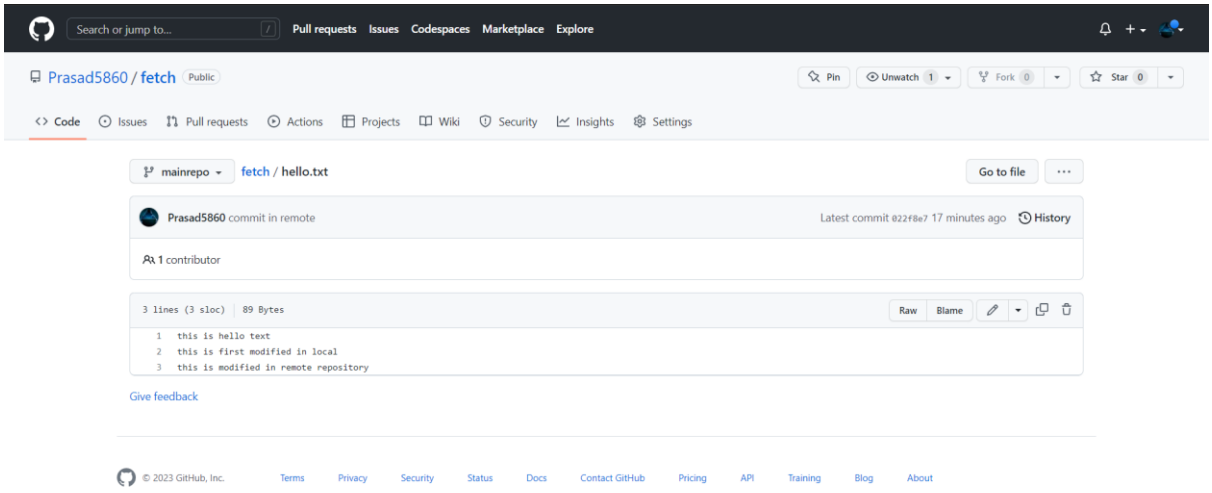

```
Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git switch master
fatal: invalid reference: master

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 717 bytes | 6.00 KiB/s, done.
From github.com:Prasad5860/fetch
   693cec5..022f8e7  mainrepo   -> origin/mainrepo

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git merge
Updating 693cec5..022f8e7
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
```

```
Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git branch -M master

Lenovo@PRASAD MINGW64 /d/github (master)
$ git log --oneline
022f8e7 (HEAD -> master, origin/mainrepo) commit in remote
693cec5 hellomdified
0ab3638 twofiles commit
```

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.








```
Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git switch master
fatal: invalid reference: master

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 717 bytes | 6.00 KiB/s, done.
From github.com:Prasad5860/fetch
   693cec5..022f8e7  mainrepo    -> origin/mainrepo

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git merge
Updating 693cec5..022f8e7
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
```

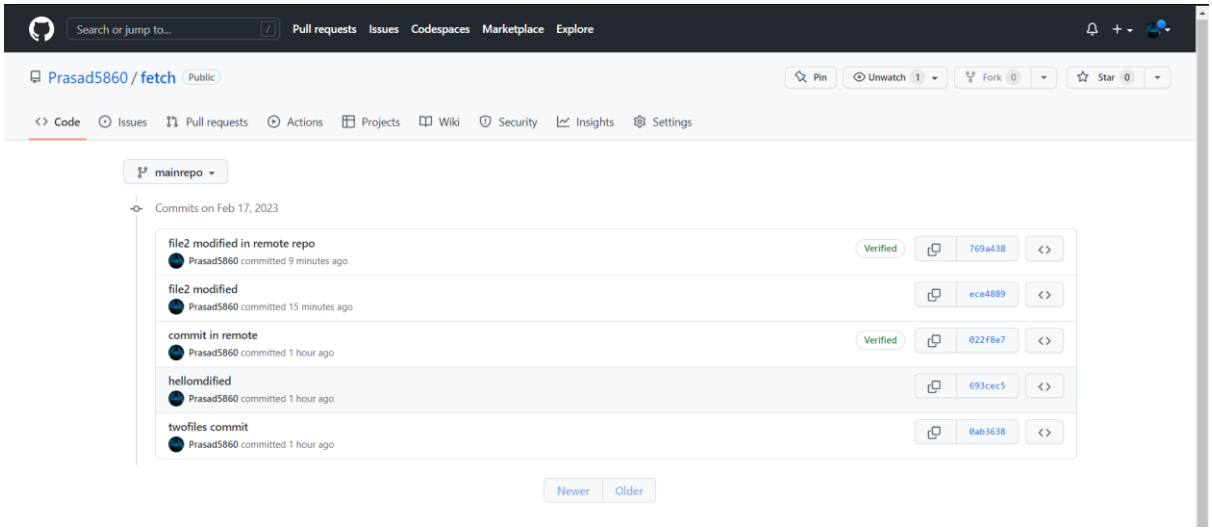
```
Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git branch -M master

Lenovo@PRASAD MINGW64 /d/github (master)
$ git log --oneline
022f8e7 (HEAD -> master, origin/mainrepo) commit in remote
693cec5 hellomdified
0ab3638 twofiles commit
```

MINGW64:/d/github

```
Lenovo@PRASAD MINGW64 /d/github (master)
$ git status
On branch master
Your branch is up to date with 'origin/mainrepo'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file2.txt

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@PRASAD MINGW64 /d/github (master)
$ git add .

Lenovo@PRASAD MINGW64 /d/github (master)
$ git status
On branch master
Your branch is up to date with 'origin/mainrepo'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file2.txt


Lenovo@PRASAD MINGW64 /d/github (master)
$ git commit -m "file2 modified"
[master ece4889] file2 modified
 1 file changed, 1 insertion(+)
```

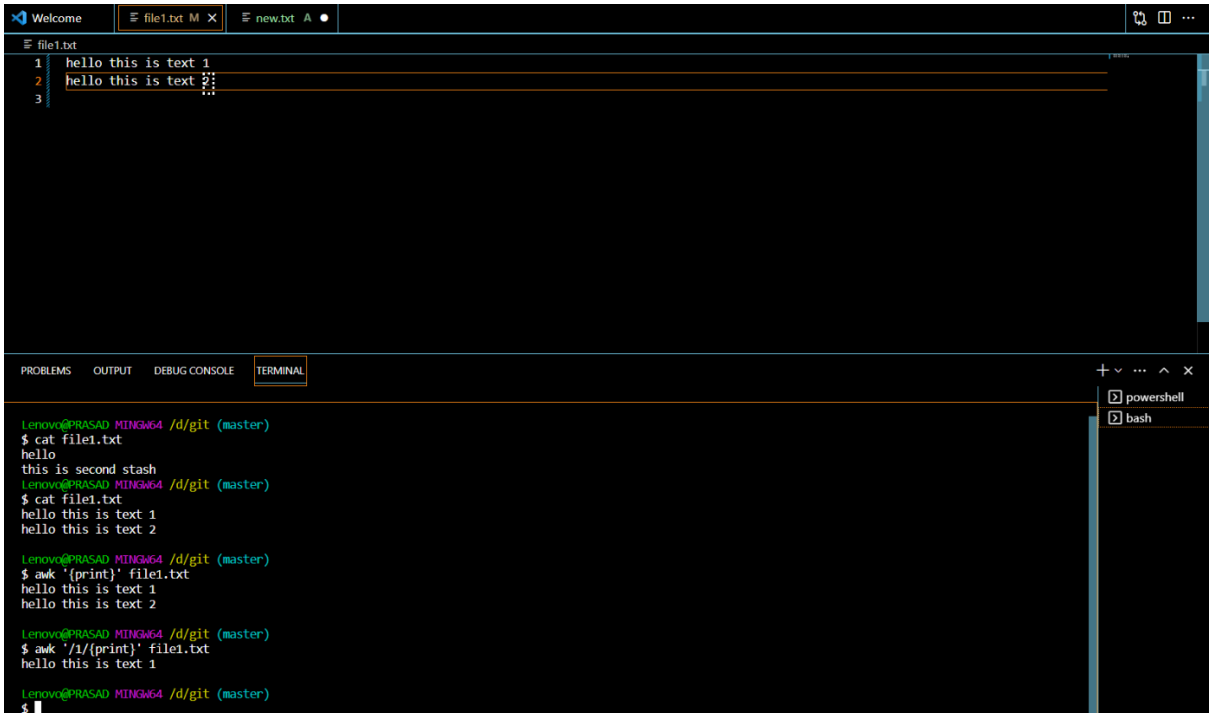
```
MINGW64:/d/github

Lenovo@PRASAD MINGW64 /d/github (master)
$ git log
commit ece488956ce42f23ff43eb9b17909b9847bf4b48 (HEAD -> master)
Author: ppr5860 <ppr58403@gmail.com>
Date:   Fri Feb 17 10:52:31 2023 +0530

    file2 modified

commit 022f8e7e636a40497901283aa88f3675bc107565 (origin/mainrepo)
Author: prasad reddy <84456097+Prasad5860@users.noreply.github.com>
Date:   Fri Feb 17 10:11:31 2023 +0530

    commit in remote

commit 693cec55da7c4f526b36bc69c78282d684641264
Author: ppr5860 <ppr58403@gmail.com>
Date:   Fri Feb 17 10:02:01 2023 +0530

    hellomdified

commit 0ab3638947e758b0f8c6c16bf444ddc3c447a30e
Author: ppr5860 <ppr58403@gmail.com>
Date:   Fri Feb 17 09:58:58 2023 +0530

    twofiles commit
```

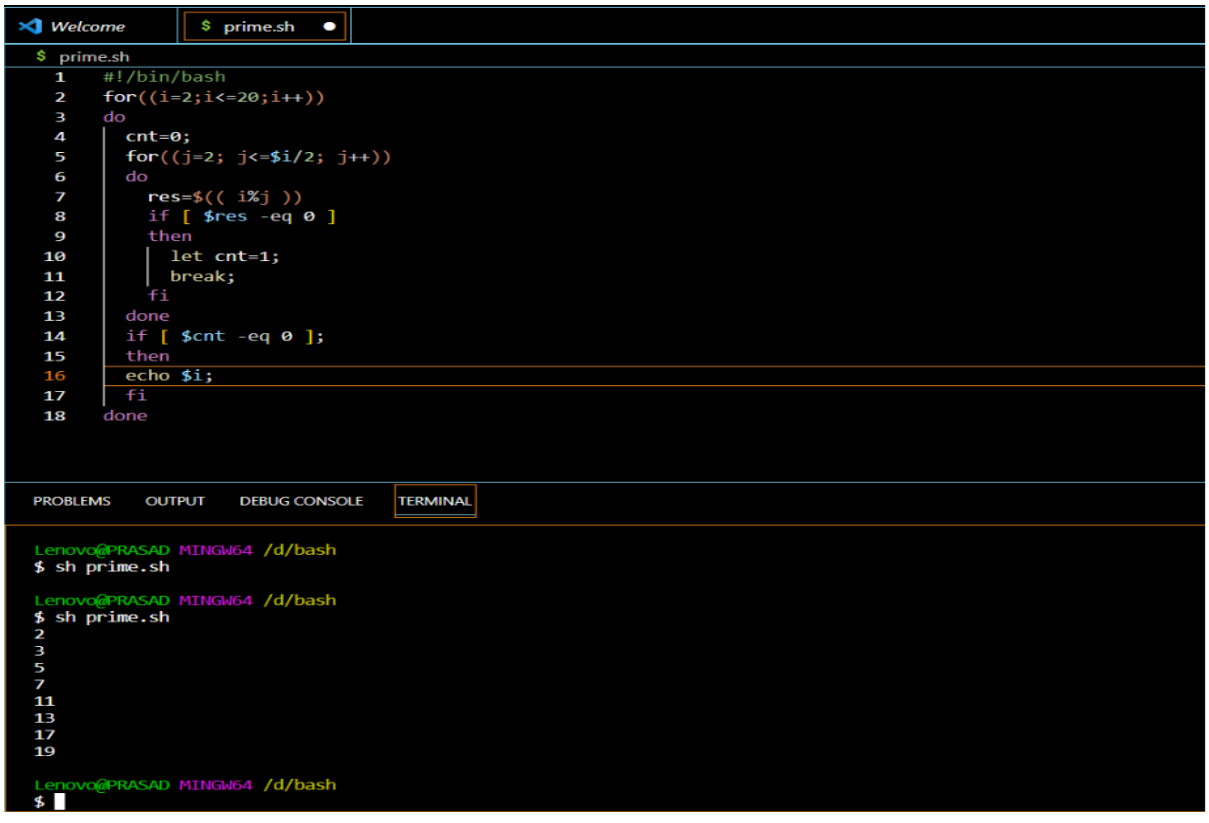
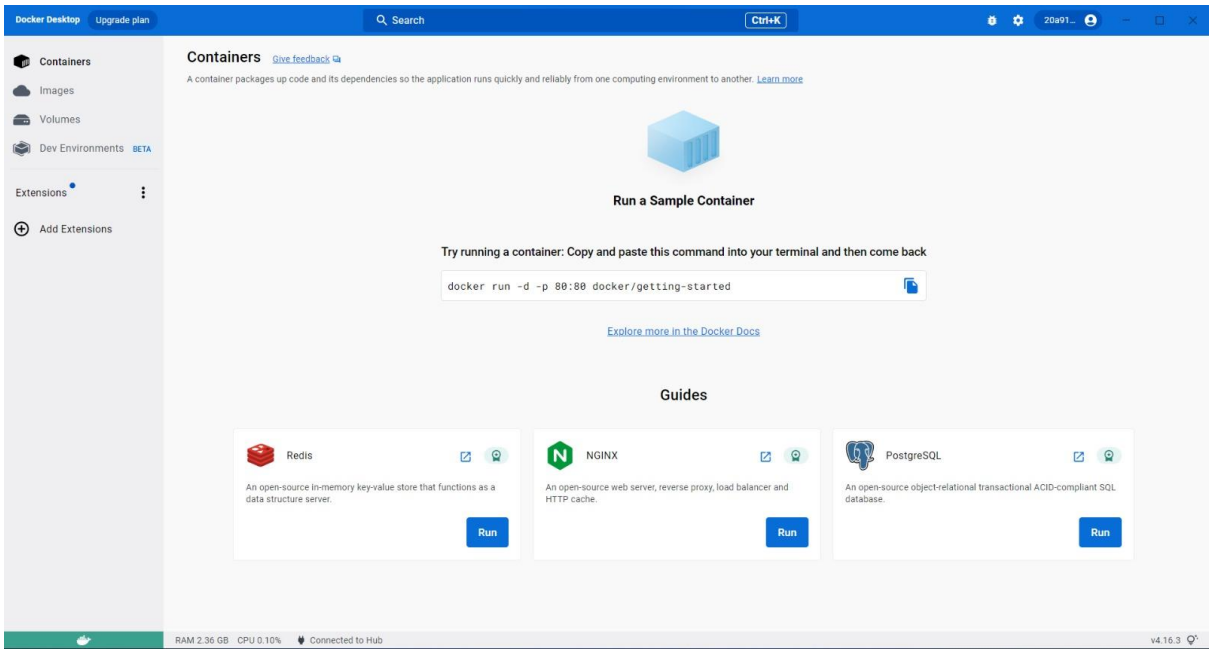
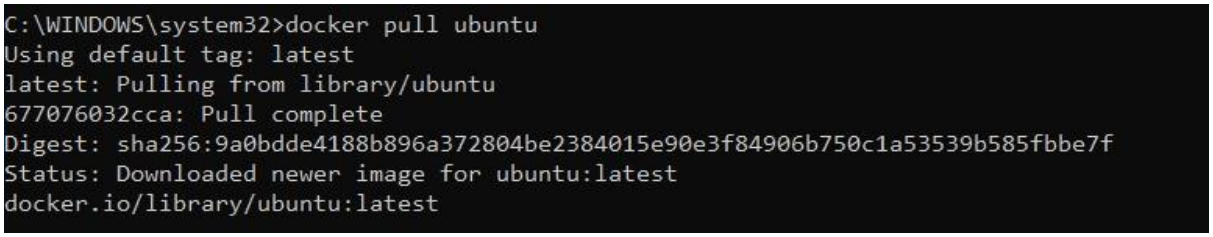
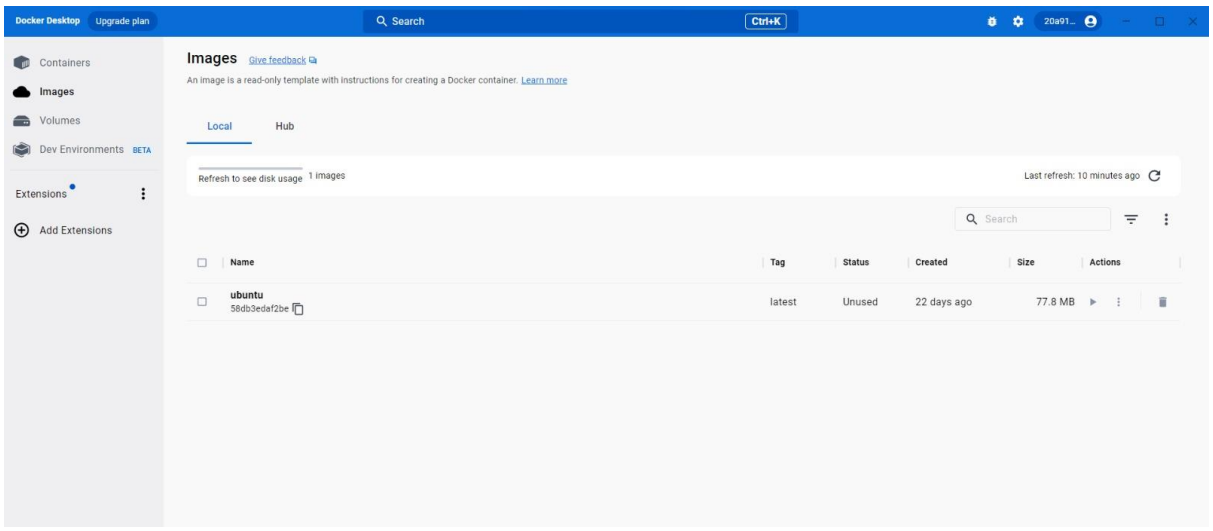
```
MINGW64:/d/github

Lenovo@PRASAD MINGW64 /d/github (master)
$ git branch -m mainrepo

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Prasad5860/fetch.git
   022f8e7..ece4889  mainrepo -> mainrepo

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git pull origin master
fatal: couldn't find remote ref master

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git pull
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), 712 bytes | 7.00 KiB/s, done.
From github.com:Prasad5860/fetch
   ece4889..769a438  mainrepo   -> origin/mainrepo
Updating ece4889..769a438
Fast-forward
 file2.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git log --oneline
769a438 (HEAD -> mainrepo, origin/mainrepo) file2 modified in  remote repo
ece4889 file2 modified
022f8e7 commit in remote
693cec5 hellomdified
```



```
Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git pull
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), 712 bytes | 7.00 KiB/s, done.
From github.com:Prasad5860/fetch
   ece4889..769a438  mainrepo   -> origin/mainrepo
Updating ece4889..769a438
Fast-forward
 file2.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git log --oneline
769a438 (HEAD -> mainrepo, origin/mainrepo) file2 modified in  remote repo
ece4889 file2 modified
022f8e7 commit in remote
693cec5 hellomdified
0ab3638 twofiles commit

Lenovo@PRASAD MINGW64 /d/github (mainrepo)
$ git branch -m master

Lenovo@PRASAD MINGW64 /d/github (master)
$ git log --oneline
769a438 (HEAD -> master, origin/mainrepo) file2 modified in  remote repo
ece4889 file2 modified
022f8e7 commit in remote
693cec5 hellomdified
0ab3638 twofiles commit

Lenovo@PRASAD MINGW64 /d/github (master)
$
```

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.





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.
All the processes pertaining to this should be provided in a screenshot for grading.