Assignment 1

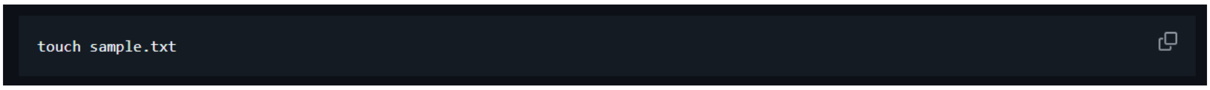
Submitted by: Ishika Dutta (CT_CSI_DV_5572)

Task 1.1

Create a file, assign permissions (read, write, execute) to different user categories (owner, group, others), and practice changing permissions using chmod.

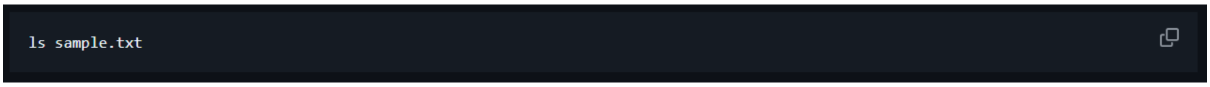
Commands Used:

To create a file



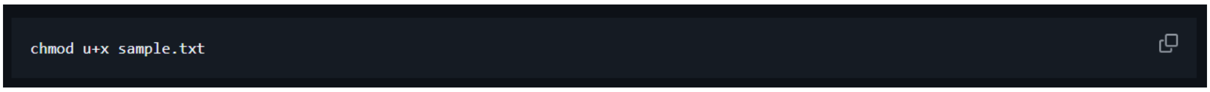
```
touch sample.txt
```

Listing the contents of directory to check whether the file has been created



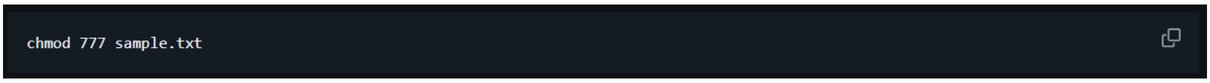
```
ls sample.txt
```

Assigning permissions using chmod: Add execute permission for the owner



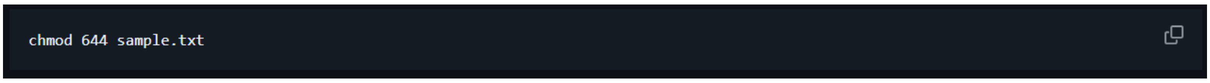
```
chmod u+x sample.txt
```

Assigning permissions using chmod: Give everyone full access (read, write, execute)



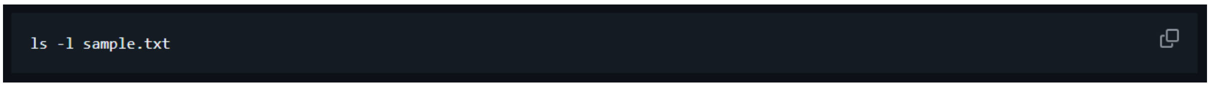
```
chmod 777 sample.txt
```

Assigning permissions using chmod: Give read & write to owner, read-only to rest



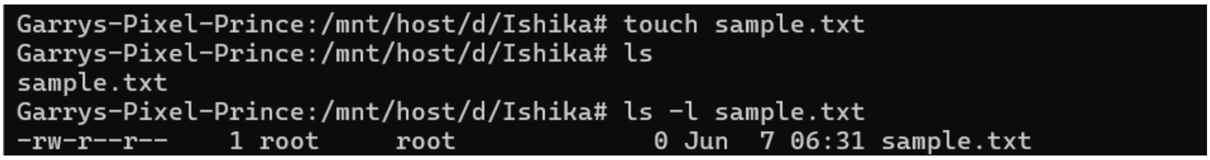
```
chmod 644 sample.txt
```

Checking the file's permissions using -l flag

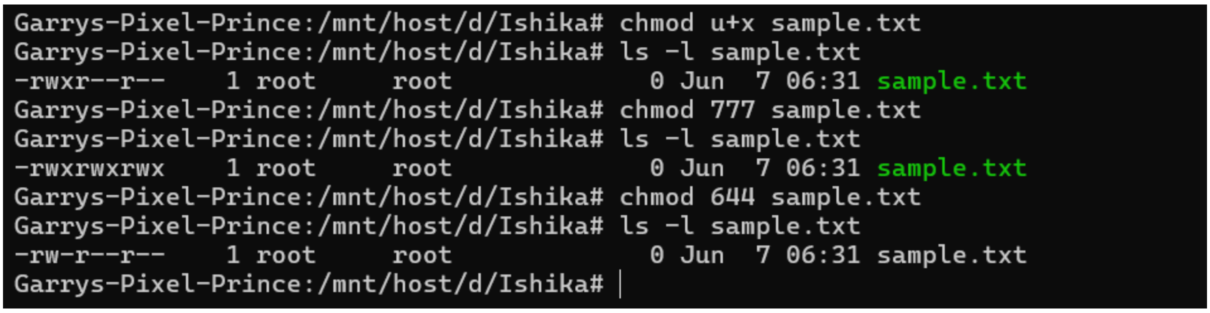


```
ls -l sample.txt
```

Exemplary Output:



```
Garrys-Pixel-Prince:/mnt/host/d/Ishika# touch sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls
sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls -l sample.txt
-rw-r--r--    1 root     root             0 Jun  7 06:31 sample.txt
```



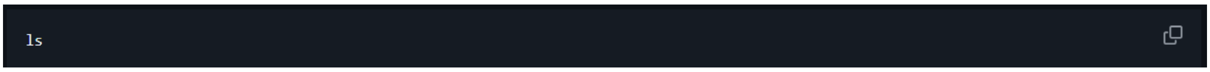
```
Garrys-Pixel-Prince:/mnt/host/d/Ishika# chmod u+x sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls -l sample.txt
-rwxr--r--    1 root     root             0 Jun  7 06:31 sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# chmod 777 sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls -l sample.txt
-rwxrwxrwx    1 root     root             0 Jun  7 06:31 sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# chmod 644 sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls -l sample.txt
-rw-r--r--    1 root     root             0 Jun  7 06:31 sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika#
```

Task 1.2

Execute basic Linux commands (e.g., ls, cd, mkdir, rm, touch) to manipulate files and directories, with an emphasis on understanding their usage.

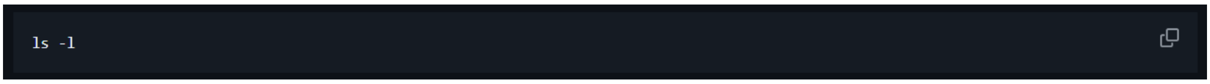
Commands Used:

List files and directories



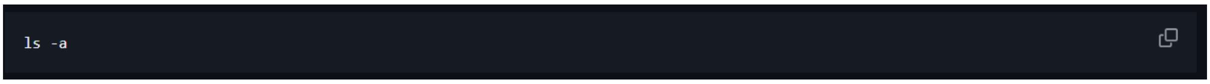
```
ls
```

Detailed (long) listing with permissions, ownership, and timestamps



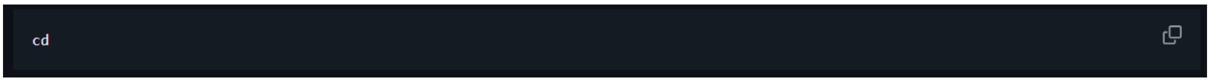
```
ls -l
```

Shows hidden files (starting with .)



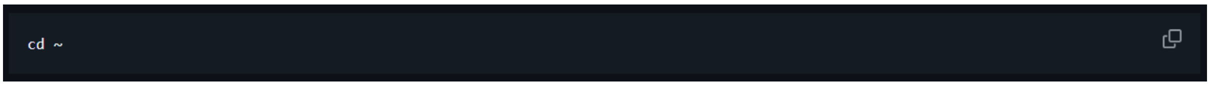
```
ls -a
```

Change directory



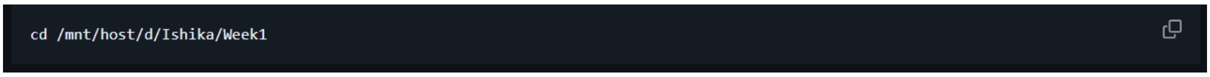
```
cd
```

Moves up one level (to the parent directory)



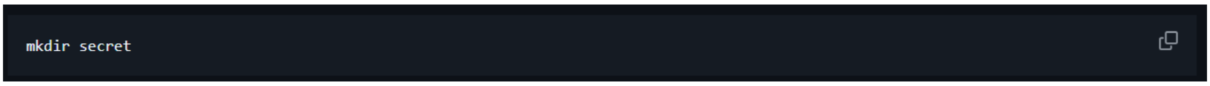
```
cd ~
```

Moving to a particular directory using path (specific to WSL)



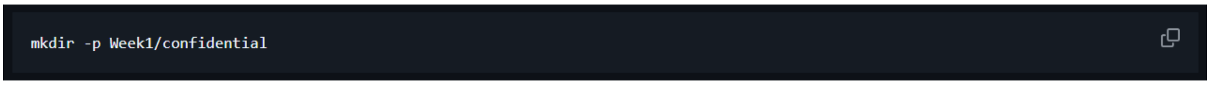
```
cd /mnt/host/d/Ishika/Week1
```

Create directory



```
mkdir secret
```

Make parent directory



```
mkdir -p Week1/confidential
```

List directories and subdirectories

```
ls -R
```

To create a file

```
touch welcome.txt
```

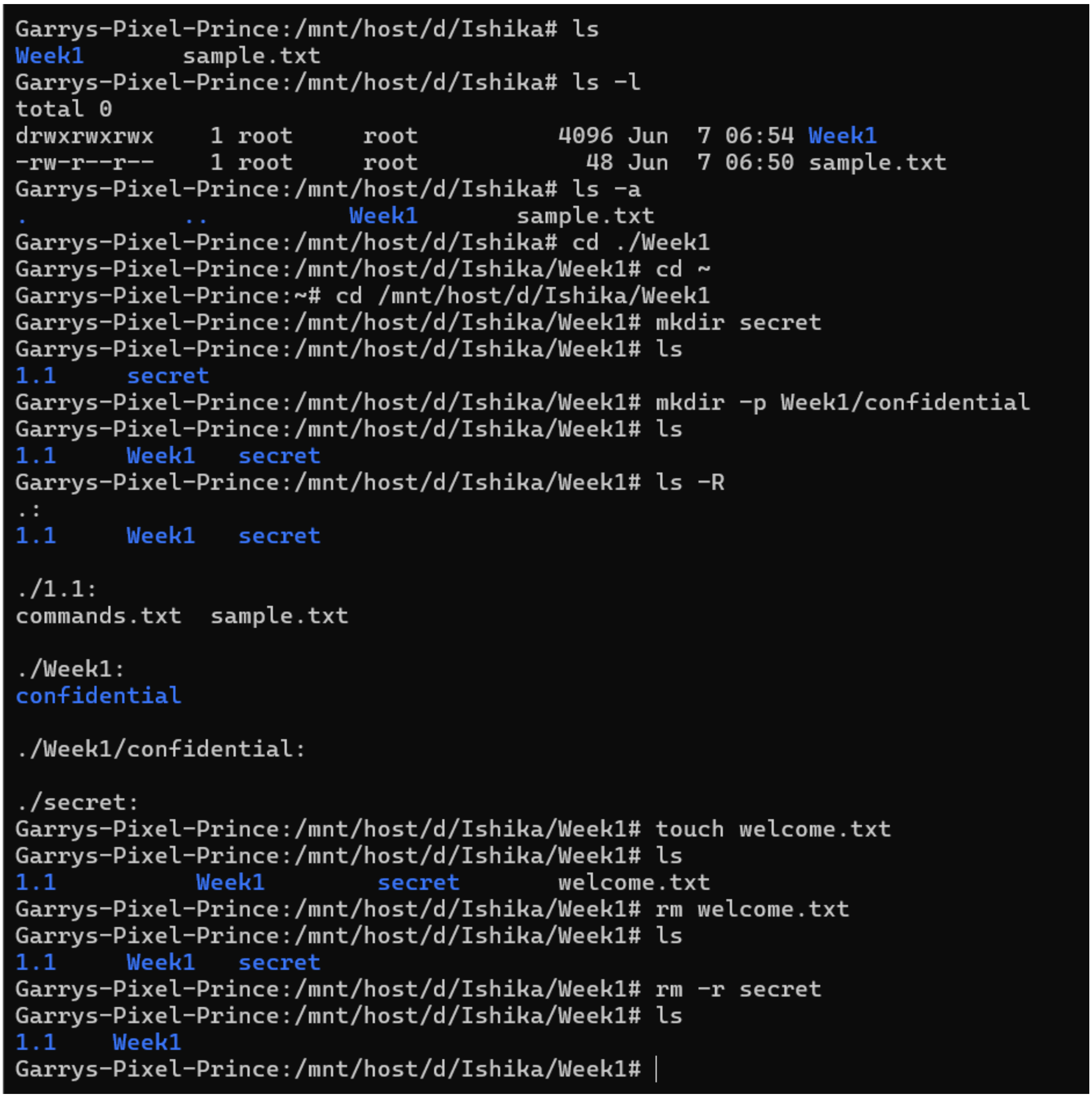
Delete a file

```
rm welcome.txt
```

Delete a folder

```
rm -r secret
```

Exemplary Output:



```
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls
Week1       sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls -l
total 0
drwxrwxrwx   1 root    root          4096 Jun  7 06:54 Week1
-rw-r--r--   1 root    root            48 Jun  7 06:50 sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls -a
.           ..          Week1       sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# cd ./Week1
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# cd ~
Garrys-Pixel-Prince:~# cd /mnt/host/d/Ishika/Week1
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# mkdir secret
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# ls
1.1     secret
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# mkdir -p Week1/confidential
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# ls
1.1     Week1   secret
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# ls -R
.:
1.1     Week1   secret

./1.1:
commands.txt  sample.txt

./Week1:
confidential

./Week1/confidential:

./secret:
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# touch welcome.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# ls
1.1          Week1        secret       welcome.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# rm welcome.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# ls
1.1     Week1   secret
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# rm -r secret
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# ls
1.1    Week1
Garrys-Pixel-Prince:/mnt/host/d/Ishika/Week1# 
```

Task 1.3

Using the terminal, practice navigating through directories, listing file contents, and moving files to different locations.

Commands Used:

Go to your home directory

```
cd ~
```

Print working/current directory

```
pwd
```

Moving to a particular directory using path (specific to WSL)

```
cd /mnt/host/d/Ishika
```

Make a directory

```
mkdir practice
```

Change directory

```
cd practice
```

Make few directories at once

```
mkdir file001 file007
```

Create a file

```
touch 001.txt
```

List files and directories

```
ls
```

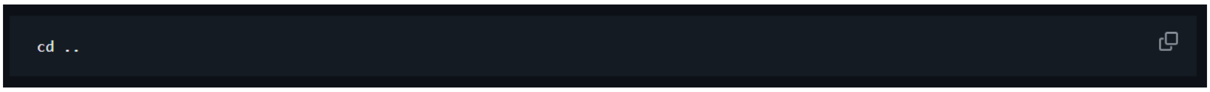
Detailed (long) listing with permissions, ownership, and timestamps

```
ls -l
```

Move file to another folder

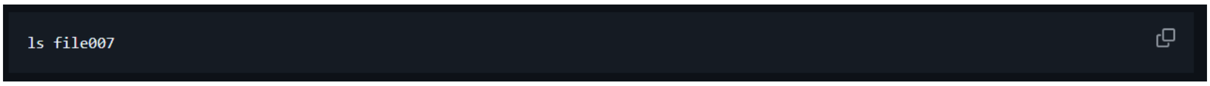
```
mv 001.txt ../file007/
```

Change directory to parent



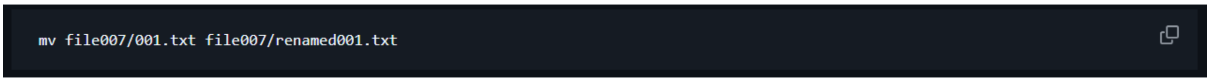
```
cd ..
```

List a subdirectory



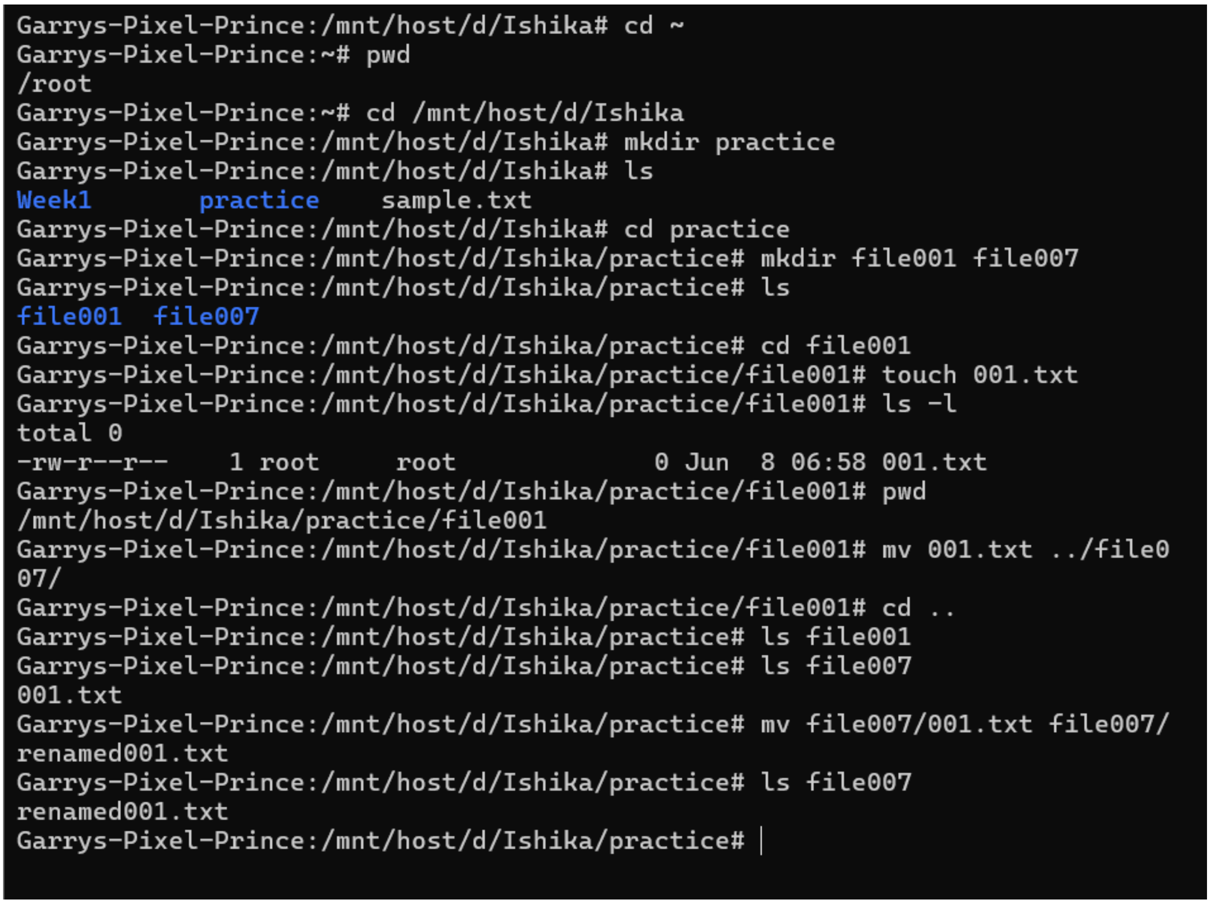
```
ls file007
```

Rename a file



```
mv file007/001.txt file007/renamed001.txt
```

Exemplary Output:

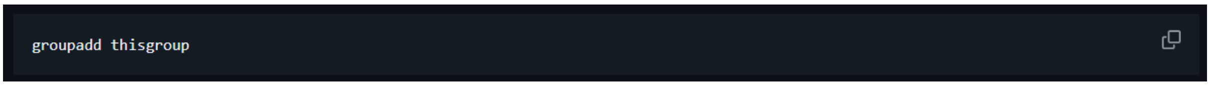


```
Garrys-Pixel-Prince:/mnt/host/d/Ishika# cd ~
Garrys-Pixel-Prince:~# pwd
/root
Garrys-Pixel-Prince:~# cd /mnt/host/d/Ishika
Garrys-Pixel-Prince:/mnt/host/d/Ishika# mkdir practice
Garrys-Pixel-Prince:/mnt/host/d/Ishika# ls
Week1       practice    sample.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika# cd practice
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# mkdir file001 file007
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# ls
file001  file007
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# cd file001
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice/file001# touch 001.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice/file001# ls -l
total 0
-rw-r--r--    1 root     root             0 Jun  8 06:58 001.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice/file001# pwd
/mnt/host/d/Ishika/practice/file001
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice/file001# mv 001.txt ../file0
07/
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice/file001# cd ..
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# ls file001
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# ls file007
001.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# mv file007/001.txt file007/
renamed001.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# ls file007
renamed001.txt
Garrys-Pixel-Prince:/mnt/host/d/Ishika/practice# |
```

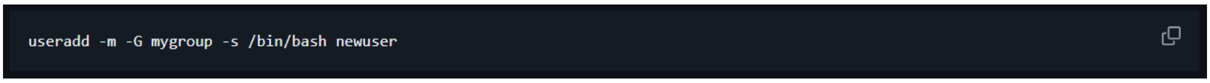
Task 1.4

Create a new user and group, set their permissions, and explore user management commands like useradd, usermod, and userdel.

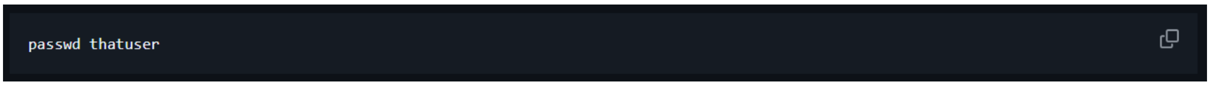
Commands Used:



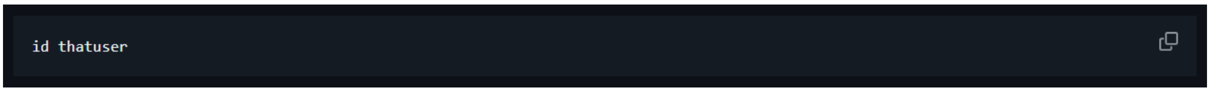
```
groupadd thisgroup
```



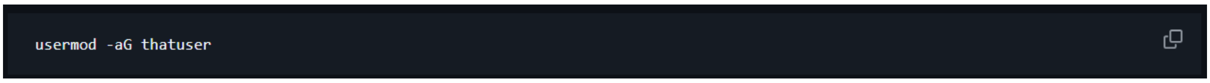
```
useradd -m -G mygroup -s /bin/bash newuser
```



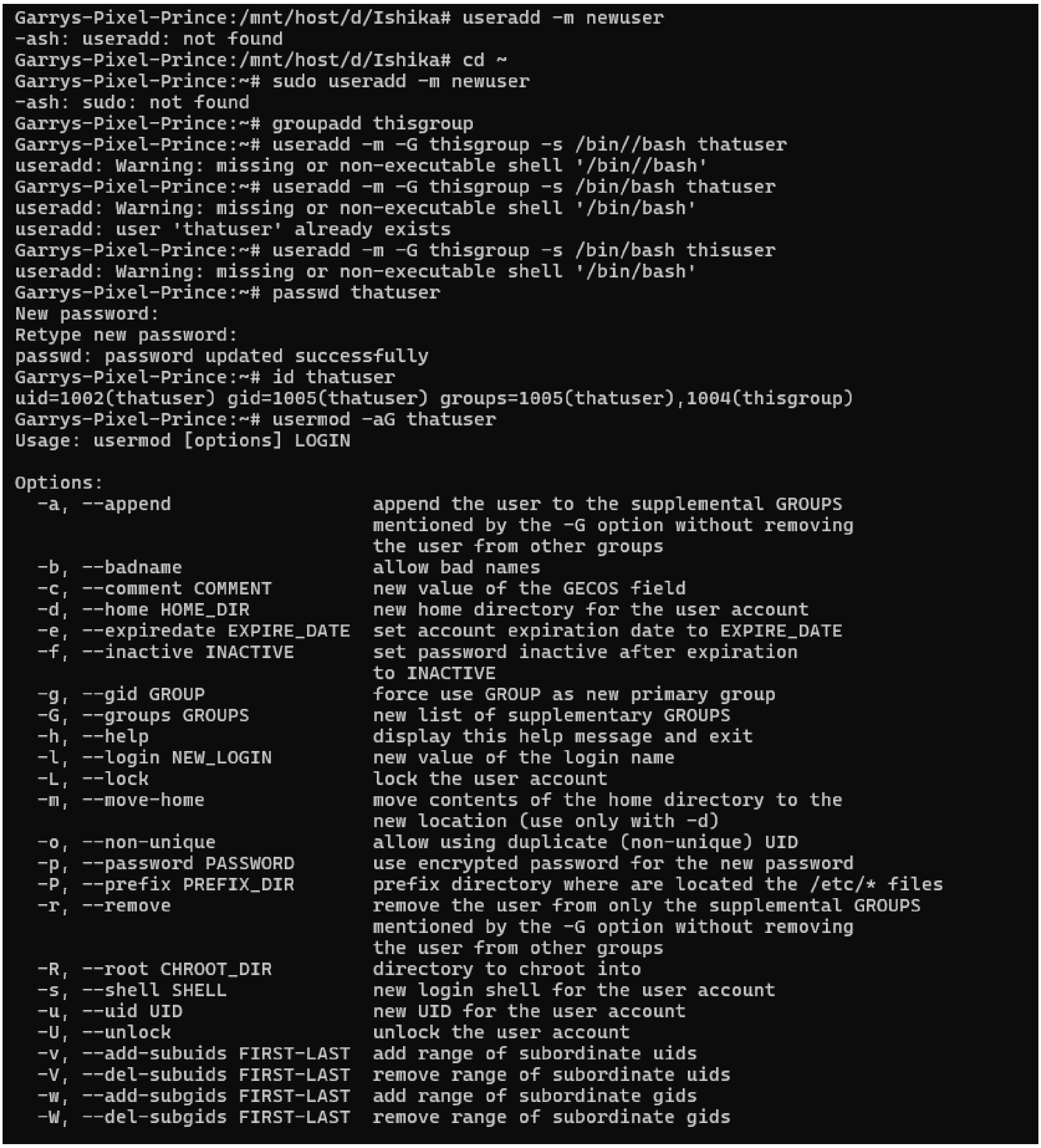
```
passwd thatuser
```



```
id thatuser
```



```
usermod -aG thatuser
```



```
Garrys-Pixel-Prince:/mnt/host/d/Ishika# useradd -m newuser
-ash: useradd: not found
Garrys-Pixel-Prince:/mnt/host/d/Ishika# cd ~
Garrys-Pixel-Prince:~# sudo useradd -m newuser
-ash: sudo: not found
Garrys-Pixel-Prince:~# groupadd thisgroup
Garrys-Pixel-Prince:~# useradd -m -G thisgroup -s /bin//bash thatuser
useradd: Warning: missing or non-executable shell '/bin//bash'
Garrys-Pixel-Prince:~# useradd -m -G thisgroup -s /bin/bash thatuser
useradd: Warning: missing or non-executable shell '/bin/bash'
useradd: user 'thatuser' already exists
Garrys-Pixel-Prince:~# useradd -m -G thisgroup -s /bin/bash thisuser
useradd: Warning: missing or non-executable shell '/bin/bash'
Garrys-Pixel-Prince:~# passwd thatuser
New password:
Retype new password:
passwd: password updated successfully
Garrys-Pixel-Prince:~# id thatuser
uid=1002(thatuser) gid=1005(thatuser) groups=1005(thatuser),1004(thisgroup)
Garrys-Pixel-Prince:~# usermod -aG thatuser
Usage: usermod [options] LOGIN

Options:
  -a, --append                  append the user to the supplemental GROUPS
                                mentioned by the -G option without removing
                                the user from other groups
  -b, --badname                 allow bad names
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new password
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -r, --remove                  remove the user from only the supplemental GROUPS
                                mentioned by the -G option without removing
                                the user from other groups
  -R, --root CHROOT_DIR         directory to chroot into
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -v, --add-subuids FIRST-LAST  add range of subordinate uids
  -V, --del-subuids FIRST-LAST  remove range of subordinate uids
  -w, --add-subgids FIRST-LAST  add range of subordinate gids
  -W, --del-subgids FIRST-LAST  remove range of subordinate gids
```

Task 1.5

More Linux commands.

Command	Description
uname -a	Show all system info
top	Live view of processes (like Task Manager)
htop	Enhanced version of top (if installed)
uptime	Show how long the system has been running
whoami	Show current user
id	Show user ID and group info
free -h	Show available memory
df -h	Show disk usage of file systems
du -sh <dir>	Show size of a specific directory
Command	**Description**
`cat file`	View file contents
`tac file`	View file contents in reverse
`head -n 10 file`	Show first 10 lines
`tail -n 10 file`	Show last 10 lines
`less file` / `more file`	Paginate through file
`wc -l file.txt`	Count lines in file
`cut -d':' -f1 /etc/passwd`	Extract column data using delimiter
`sort file.txt`	Sort contents of file
`uniq file.txt`	Show unique lines (usually used with `sort`)
`grep "text" file.txt`	Search for text in file
`find . -name "*.txt"`	Find all `.txt` files in current dir
`xargs`	Run commands on input from `stdin` (ex: `find ...
Command	**Description**
`sudo apt update`	Refresh package list
`sudo apt upgrade`	Upgrade installed packages
`sudo apt install package-name`	Install a package
`dpkg -l`	List installed packages
`dpkg -i package.deb`	Install a local `.deb` package
Command	**Description**
`who`	Show who is logged in

`groups`	Show groups of current user
`addgroup groupname`	Add a new group
`usermod -aG groupname username`	Add user to a group
`passwd username`	Set/change a password for a user

Command	Description
`tar -cvf archive.tar folder`	Create `.tar` file
`tar -xvf archive.tar`	Extract `.tar` file
`tar -czvf file.tar.gz folder`	Create compressed archive
`gzip file.txt`	Compress file
`gunzip file.txt.gz`	Decompress file

Command	Description
`ping google.com`	Check network connectivity
`curl http://example.com`	Fetch content of a URL
`wget http://example.com/file.zip`	Download a file
`ifconfig` or `ip a`	Show IP and network config
`netstat -tulnp`	Show listening ports
`ss -tulwn`	Faster alternative to `netstat`

Command	Description
`chmod +x script.sh`	Make script executable
`chown user:group file`	Change file ownership
`umask`	Show default permissions for new files

Command	Description
`ps aux`	List running processes
`kill <PID>`	Kill process by PID
`killall <name>`	Kill all processes with a given name
`bg` / `fg`	Send process to background/foreground
`jobs`	List background jobs

Command	Description
`history`	Show command history
`clear` or `Ctrl+L`	Clear terminal screen
`alias ll='ls -la'`	Create shortcut/alias
`crontab -e`	Schedule tasks (cron jobs)

Task 1.6

Introduction to version control and Git basics, Git installation and configuration, Basic Git commands (init, add, commit, push, pull)

Commands:

Create a new Git repository

```
git init
```

Copy a remote repository to local machine

```
git clone https://github.com/username/project.git
```

Check the Status of Files

```
git status
```

Add Files to Staging Area

```
git add index.html
```

Commit Changes

```
git commit -m "initial commit"
```

View Commit History

```
git log
```

Push Changes

```
git push origin main
```

Pull Changes

```
git pull origin main
```

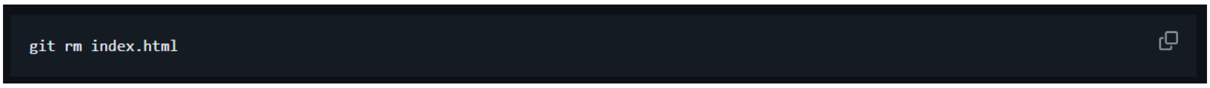
Create a new branch

```
git branch feature-branch
```

Switch to that branch

```
git checkout feature-branch
```

Remove a File



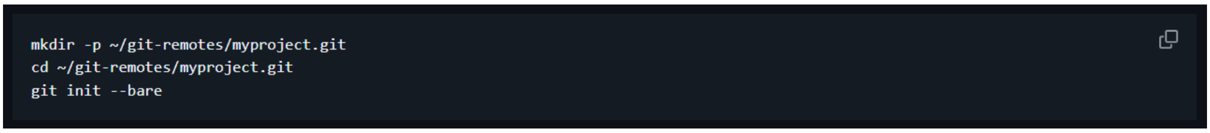
```
git rm index.html
```

Task 1.7

Setup a remote repository in Local , Add a file and commit or save the changes and push to master branch

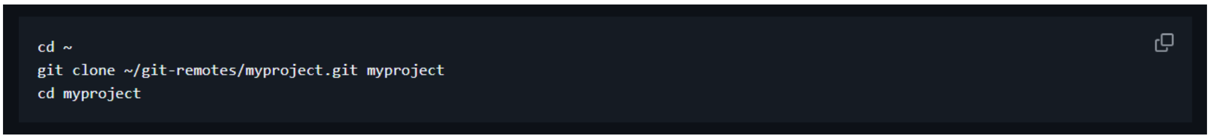
Commands:

Create a bare remote repository



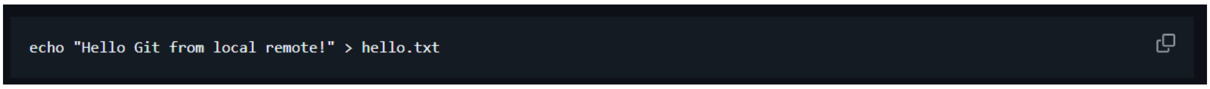
```
mkdir -p ~/git-remotes/myproject.git
cd ~/git-remotes/myproject.git
git init --bare
```

Clone the remote to start working locally



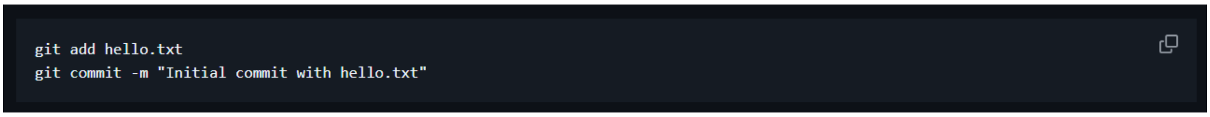
```
cd ~
git clone ~/git-remotes/myproject.git myproject
cd myproject
```

Add a file



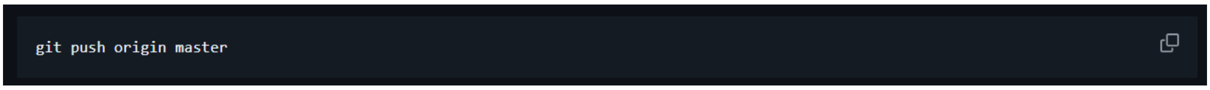
```
echo "Hello Git from local remote!" > hello.txt
```

Stage and commit the file



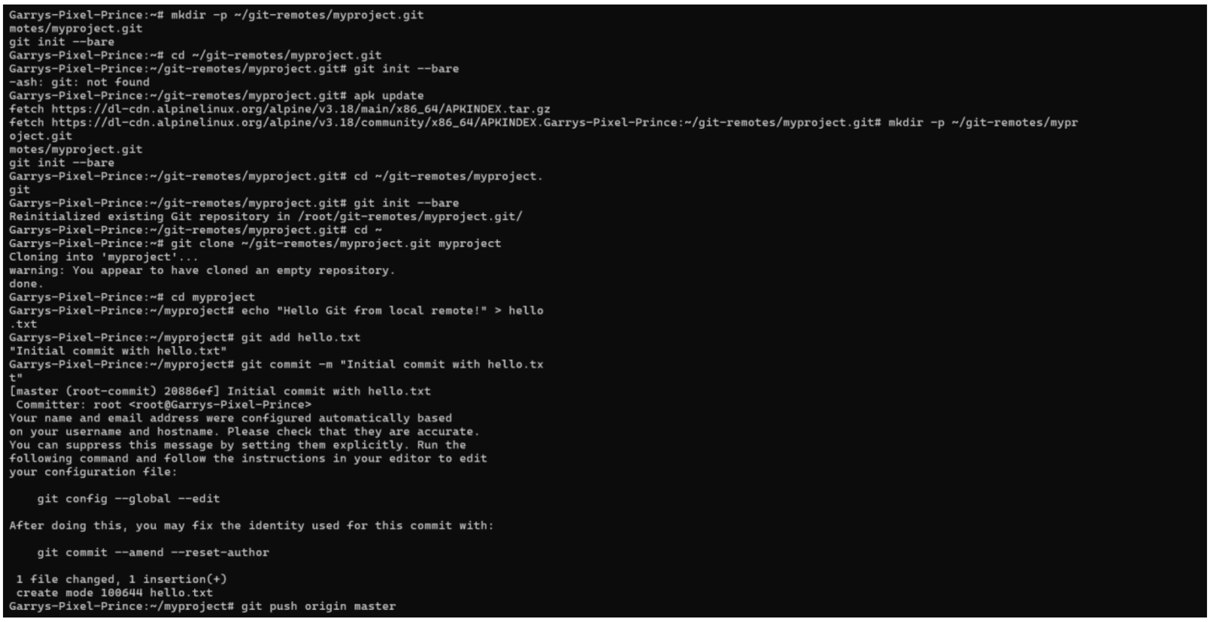
```
git add hello.txt
git commit -m "Initial commit with hello.txt"
```

Push to the master branch



```
git push origin master
```

Output:



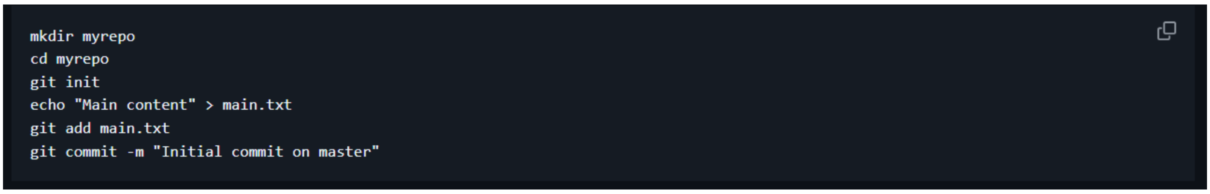
```
Garrys-Pixel-Prince:~# mkdir -p ~/git-remotes/myproject.git
motes/myproject.git
git init --bare
Garrys-Pixel-Prince:~# cd ~/git-remotes/myproject.git
Garrys-Pixel-Prince:~/git-remotes/myproject.git# git init --bare
-ash: git: not found
Garrys-Pixel-Prince:~/git-remotes/myproject.git# apk update
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/community/x86_64/APKINDEX.Garrys-Pixel-Prince:~/git-remotes/myproject.git# mkdir -p ~/git-remotes/mypr
oject.git
motes/myproject.git
git init --bare
Garrys-Pixel-Prince:~/git-remotes/myproject.git# cd ~/git-remotes/myproject.
git
Garrys-Pixel-Prince:~/git-remotes/myproject.git# git init --bare
Reinitialized existing Git repository in /root/git-remotes/myproject.git/
Garrys-Pixel-Prince:~/git-remotes/myproject.git# cd ~
Garrys-Pixel-Prince:~# git clone ~/git-remotes/myproject.git myproject
Cloning into 'myproject'...
warning: You appear to have cloned an empty repository.
done.
Garrys-Pixel-Prince:~# cd myproject
Garrys-Pixel-Prince:~/myproject# echo "Hello Git from local remote!" > hello
.txt
Garrys-Pixel-Prince:~/myproject# git add hello.txt
"Initial commit with hello.txt"
Garrys-Pixel-Prince:~/myproject# git commit -m "Initial commit with hello.tx
t"
[master (root-commit) 20886ef] Initial commit with hello.txt
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
Garrys-Pixel-Prince:~/myproject# git push origin master
```

Task 1.8

Merge types, Create a new branch then commit and push the changes to new branch and merge it with the master branch using pull request.

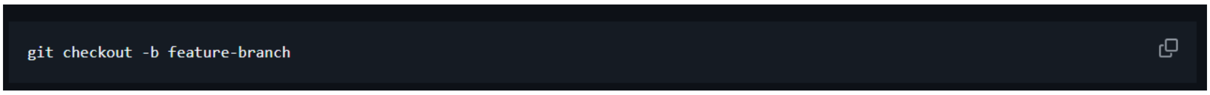
Commands:

Setup and Initialization



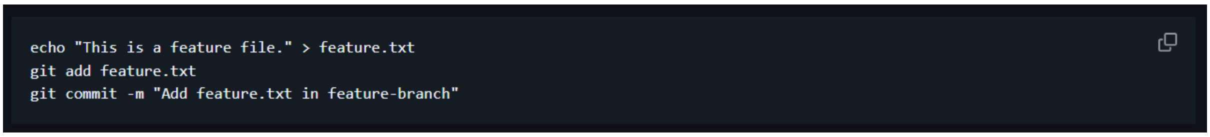
```
mkdir myrepo
cd myrepo
git init
echo "Main content" > main.txt
git add main.txt
git commit -m "Initial commit on master"
```

Create a New Branch



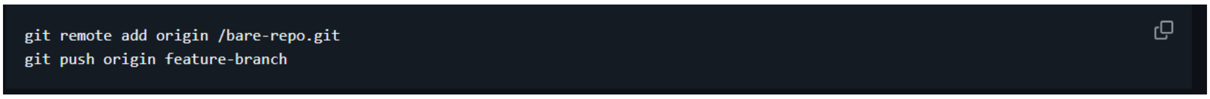
```
git checkout -b feature-branch
```

Add and Commit Changes



```
echo "This is a feature file." > feature.txt
git add feature.txt
git commit -m "Add feature.txt in feature-branch"
```

Push the Branch



```
git remote add origin /bare-repo.git
git push origin feature-branch
```

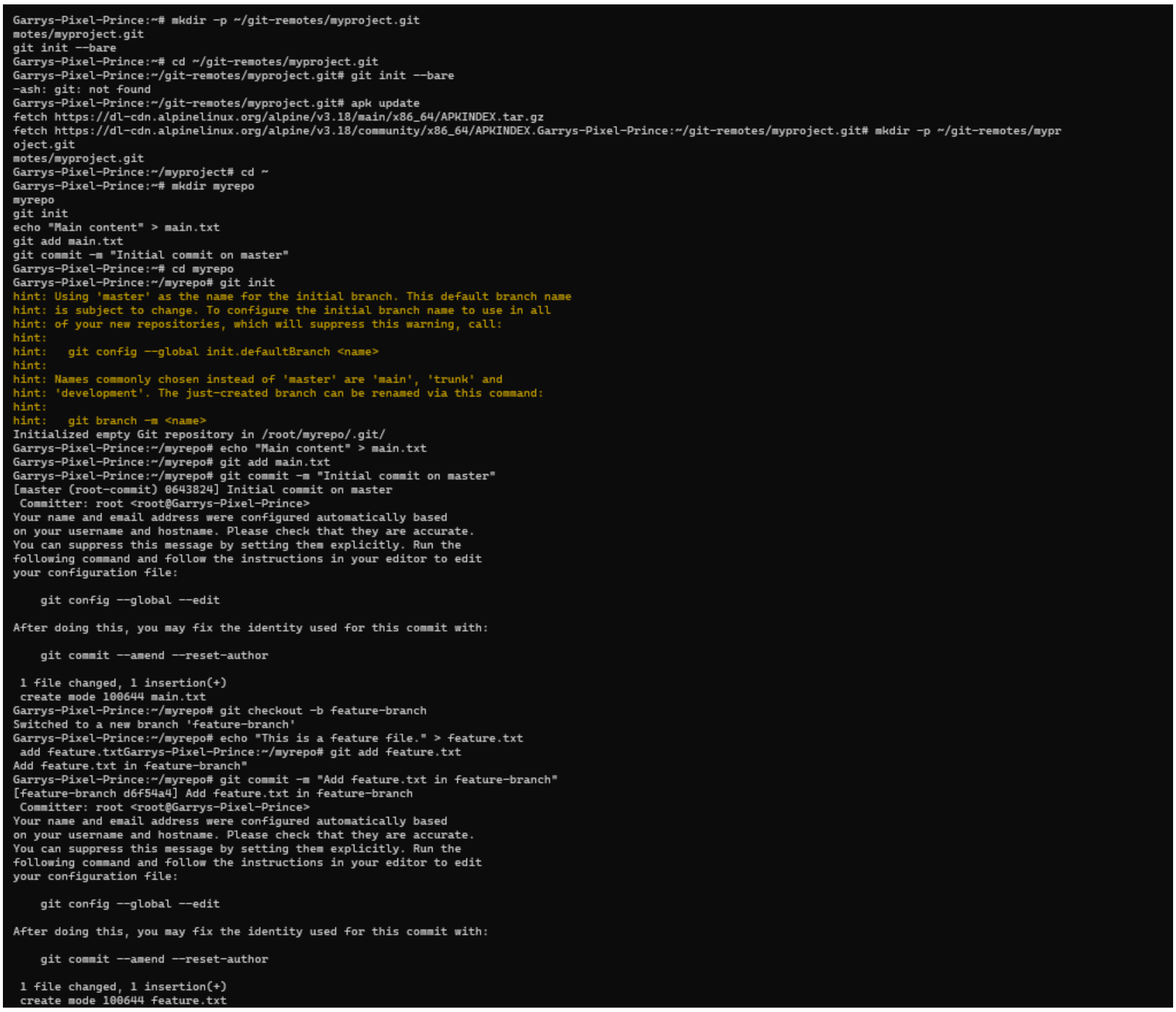
Switch Back to master and Merge

```
git checkout master
git pull origin master      # make sure master is up to date
git merge feature-branch    # merges the branch
```

Push Merged Master

```
git push origin master
```

Output:



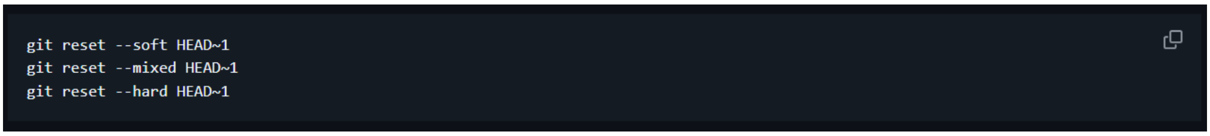
```
Garrys-Pixel-Prince:~# mkdir -p ~/git-remotes/myproject.git
motes/myproject.git
git init --bare
Garrys-Pixel-Prince:~# cd ~/git-remotes/myproject.git
Garrys-Pixel-Prince:~/git-remotes/myproject.git# git init --bare
-ash: git: not found
Garrys-Pixel-Prince:~/git-remotes/myproject.git# apk update
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/community/x86_64/APKINDEX.Garrys-Pixel-Prince:~/git-remotes/myproject.git# mkdir -p ~/git-remotes/mypr
oject.git
motes/myproject.git
Garrys-Pixel-Prince:~/myproject# cd ~
Garrys-Pixel-Prince:~# mkdir myrepo
myrepo
git init
echo "Main content" > main.txt
git add main.txt
git commit -m "Initial commit on master"
Garrys-Pixel-Prince:~# cd myrepo
Garrys-Pixel-Prince:~/myrepo# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/myrepo/.git/
Garrys-Pixel-Prince:~/myrepo# echo "Main content" > main.txt
Garrys-Pixel-Prince:~/myrepo# git add main.txt
Garrys-Pixel-Prince:~/myrepo# git commit -m "Initial commit on master"
[master (root-commit) 0643824] Initial commit on master
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 main.txt
Garrys-Pixel-Prince:~/myrepo# git checkout -b feature-branch
Switched to a new branch 'feature-branch'
Garrys-Pixel-Prince:~/myrepo# echo "This is a feature file." > feature.txt
 add feature.txtGarrys-Pixel-Prince:~/myrepo# git add feature.txt
Add feature.txt in feature-branch"
Garrys-Pixel-Prince:~/myrepo# git commit -m "Add feature.txt in feature-branch"
[feature-branch d6f54a4] Add feature.txt in feature-branch
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
```

Task 1.9

Undo the last commit or remove the last created file from remote repo using CLI

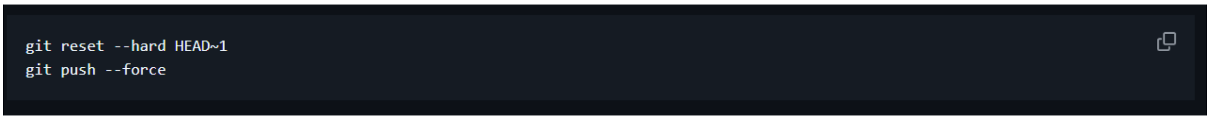
Commands:

Undo Last Commit (Before Pushing)



```
git reset --soft HEAD~1
git reset --mixed HEAD~1
git reset --hard HEAD~1
```

Undo Last Commit (After Push)



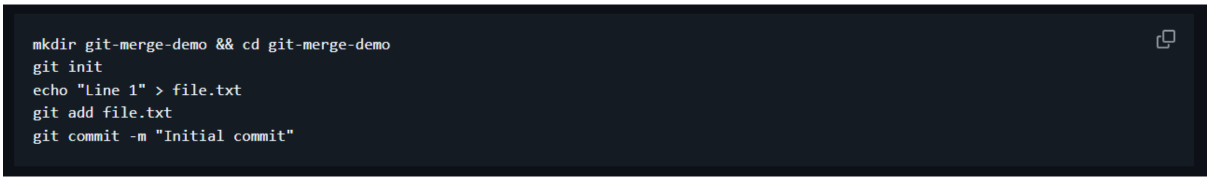
```
git reset --hard HEAD~1
git push --force
```

Task 1.10

Branching and merging in Git, Resolve a merge conflicts

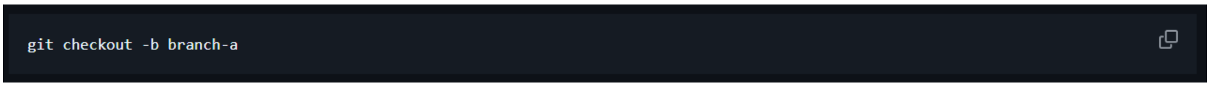
Commands:

Initialize a repo and commit something



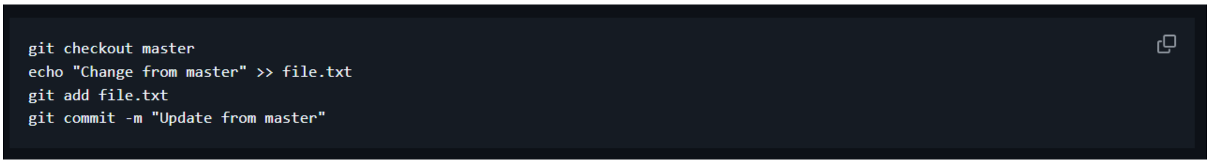
```
mkdir git-merge-demo && cd git-merge-demo
git init
echo "Line 1" > file.txt
git add file.txt
git commit -m "Initial commit"
```

Create and switch to a new branch



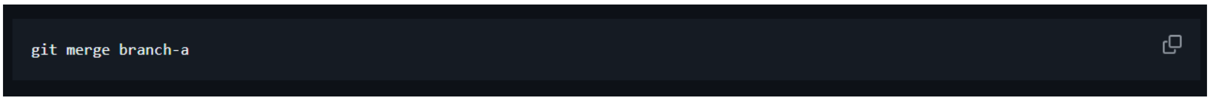
```
git checkout -b branch-a
```

Switch to master and create a conflicting change



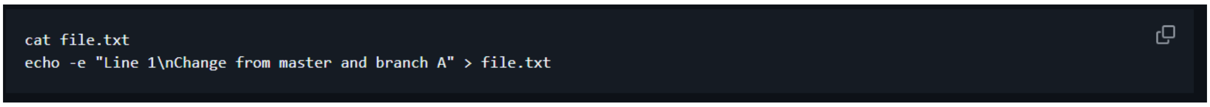
```
git checkout master
echo "Change from master" >> file.txt
git add file.txt
git commit -m "Update from master"
```

Merge branch-a into master



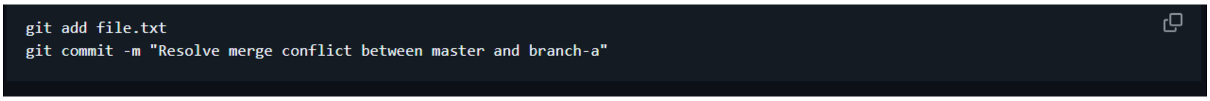
```
git merge branch-a
```

Resolve the Merge Conflict



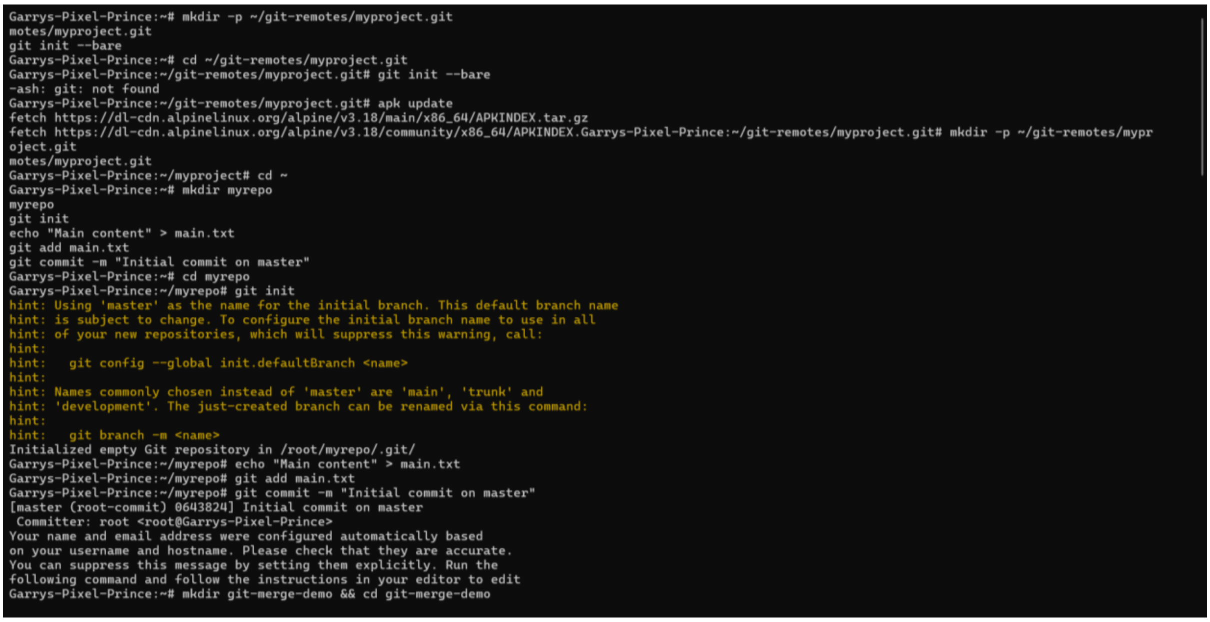
```
cat file.txt
echo -e "Line 1\nChange from master and branch A" > file.txt
```

Add the resolved file and commit

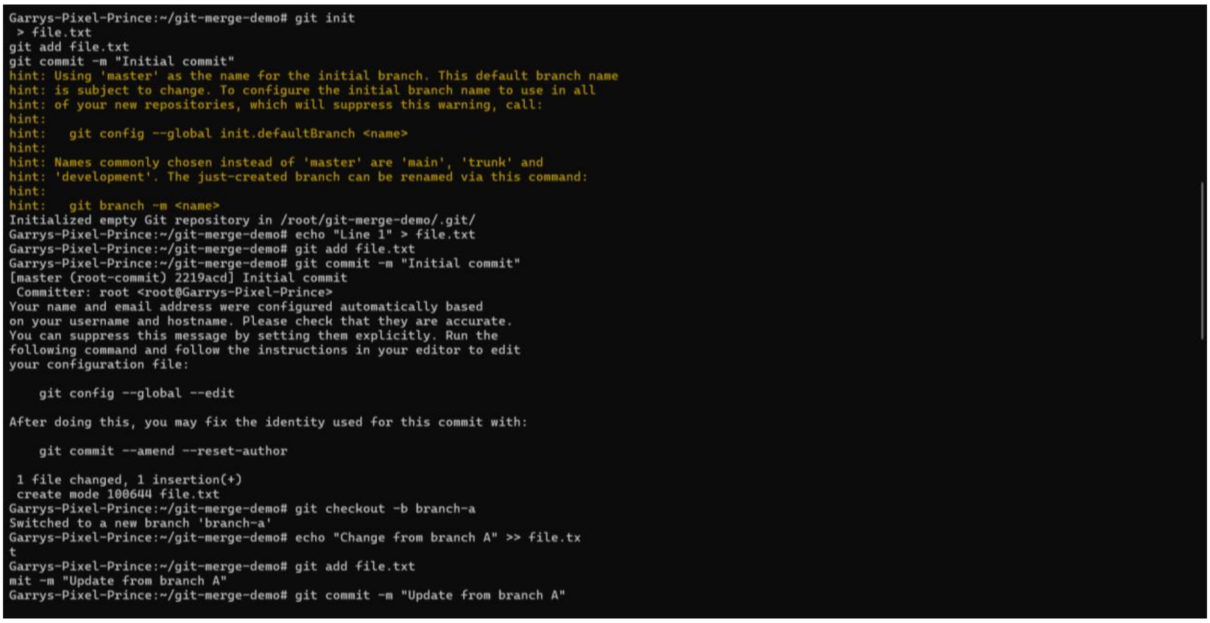


```
git add file.txt
git commit -m "Resolve merge conflict between master and branch-a"
```

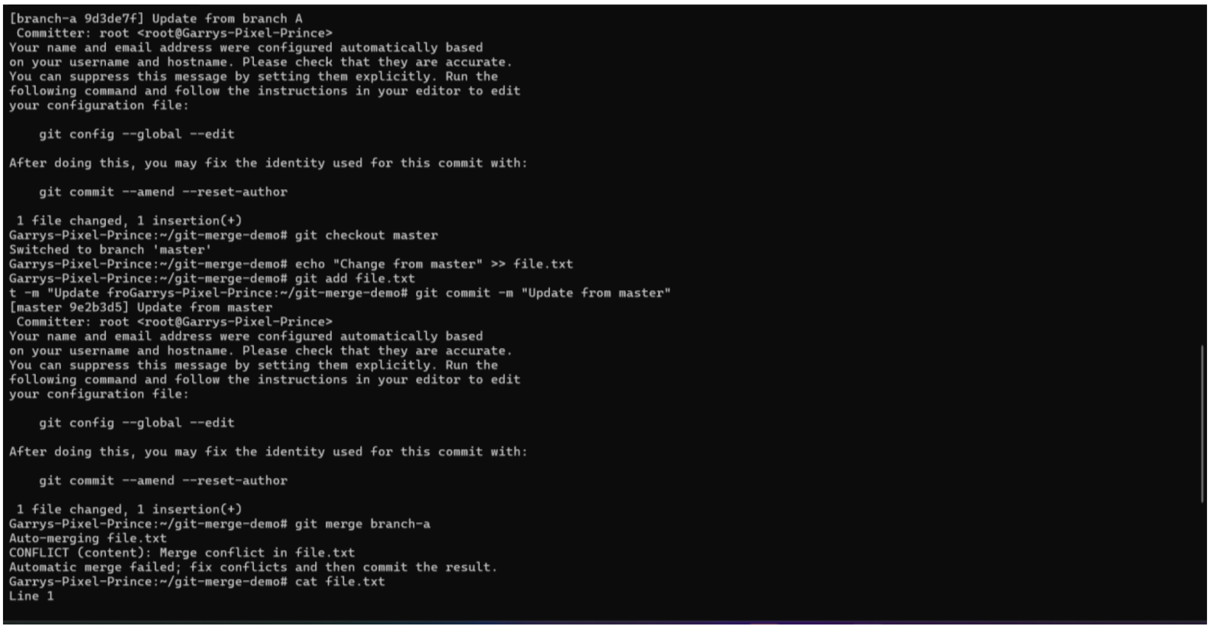
Output:



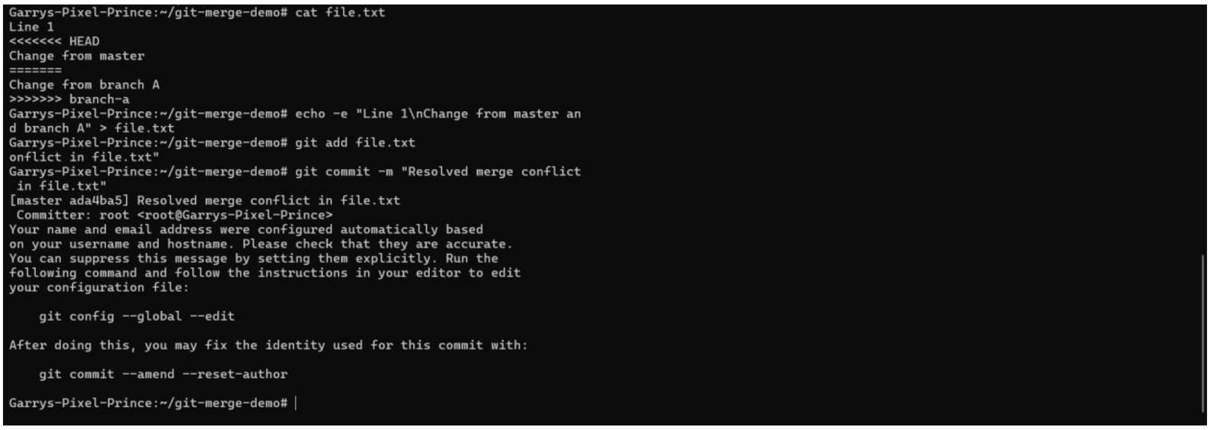
```
Garrys-Pixel-Prince:~# mkdir -p ~/git-remotes/myproject.git
motes/myproject.git
git init --bare
Garrys-Pixel-Prince:~# cd ~/git-remotes/myproject.git
Garrys-Pixel-Prince:~/git-remotes/myproject.git# git init --bare
-ash: git: not found
Garrys-Pixel-Prince:~/git-remotes/myproject.git# apk update
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/community/x86_64/APKINDEX.Garrys-Pixel-Prince:~/git-remotes/myproject.git# mkdir -p ~/git-remotes/mypr
oject.git
motes/myproject.git
Garrys-Pixel-Prince:~/myproject# cd ~
Garrys-Pixel-Prince:~# mkdir myrepo
myrepo
git init
echo "Main content" > main.txt
git add main.txt
git commit -m "Initial commit on master"
Garrys-Pixel-Prince:~# cd myrepo
Garrys-Pixel-Prince:~/myrepo# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/myrepo/.git/
Garrys-Pixel-Prince:~/myrepo# echo "Main content" > main.txt
Garrys-Pixel-Prince:~/myrepo# git add main.txt
Garrys-Pixel-Prince:~/myrepo# git commit -m "Initial commit on master"
[master (root-commit) 0643824] Initial commit on master
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
Garrys-Pixel-Prince:~# mkdir git-merge-demo && cd git-merge-demo
```



```
Garrys-Pixel-Prince:~/git-merge-demo# git init
 > file.txt
git add file.txt
git commit -m "Initial commit"
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/git-merge-demo/.git/
Garrys-Pixel-Prince:~/git-merge-demo# echo "Line 1" > file.txt
Garrys-Pixel-Prince:~/git-merge-demo# git add file.txt
Garrys-Pixel-Prince:~/git-merge-demo# git commit -m "Initial commit"
[master (root-commit) 2219acd] Initial commit
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
Garrys-Pixel-Prince:~/git-merge-demo# git checkout -b branch-a
Switched to a new branch 'branch-a'
Garrys-Pixel-Prince:~/git-merge-demo# echo "Change from branch A" >> file.tx
t
Garrys-Pixel-Prince:~/git-merge-demo# git add file.txt
mit -m "Update from branch A"
Garrys-Pixel-Prince:~/git-merge-demo# git commit -m "Update from branch A"
```



```
[branch-a 9d3de7f] Update from branch A
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
Garrys-Pixel-Prince:~/git-merge-demo# git checkout master
Switched to branch 'master'
Garrys-Pixel-Prince:~/git-merge-demo# echo "Change from master" >> file.txt
Garrys-Pixel-Prince:~/git-merge-demo# git add file.txt
t -m "Update froGarrys-Pixel-Prince:~/git-merge-demo# git commit -m "Update from master"
[master 9e2b3d5] Update from master
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
Garrys-Pixel-Prince:~/git-merge-demo# git merge branch-a
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
Garrys-Pixel-Prince:~/git-merge-demo# cat file.txt
Line 1
```



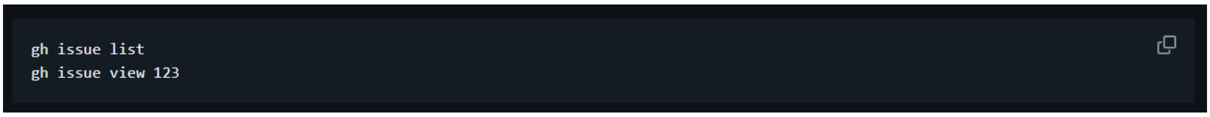
```
Garrys-Pixel-Prince:~/git-merge-demo# cat file.txt
Line 1
<<<<<<< HEAD
Change from master
=======
Change from branch A
>>>>>>> branch-a
Garrys-Pixel-Prince:~/git-merge-demo# echo -e "Line 1\nChange from master an
d branch A" > file.txt
Garrys-Pixel-Prince:~/git-merge-demo# git add file.txt
onflict in file.txt"
Garrys-Pixel-Prince:~/git-merge-demo# git commit -m "Resolved merge conflict
 in file.txt"
[master ada4ba5] Resolved merge conflict in file.txt
 Committer: root <root@Garrys-Pixel-Prince>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

Garrys-Pixel-Prince:~/git-merge-demo# |
```

Task 1.11

More git commands

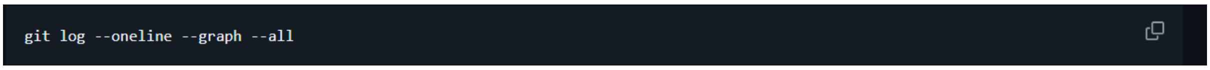
Commands:

View issues



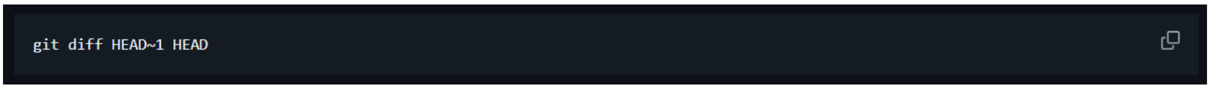
```
gh issue list
gh issue view 123
```

View detailed commit history



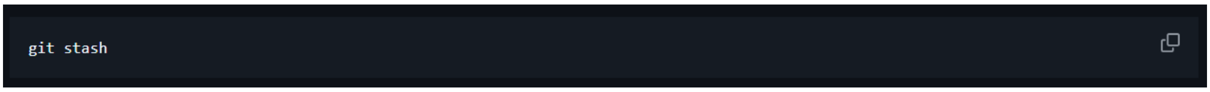
```
git log --oneline --graph --all
```

See changes across commits

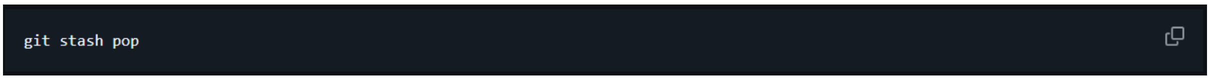


```
git diff HEAD~1 HEAD
```

Stash uncommitted changes

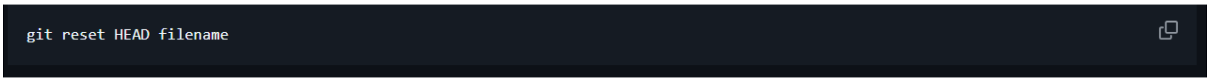


```
git stash
```



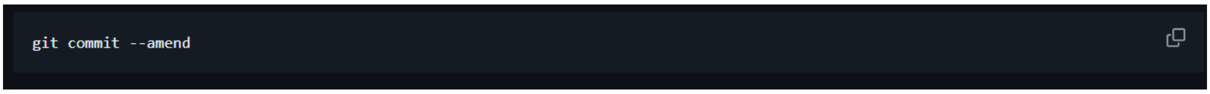
```
git stash pop
```

Temporarily save dirty (uncommitted) changes



```
git reset HEAD filename
```

Rename the latest commit message



```
git commit --amend
```