1)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.

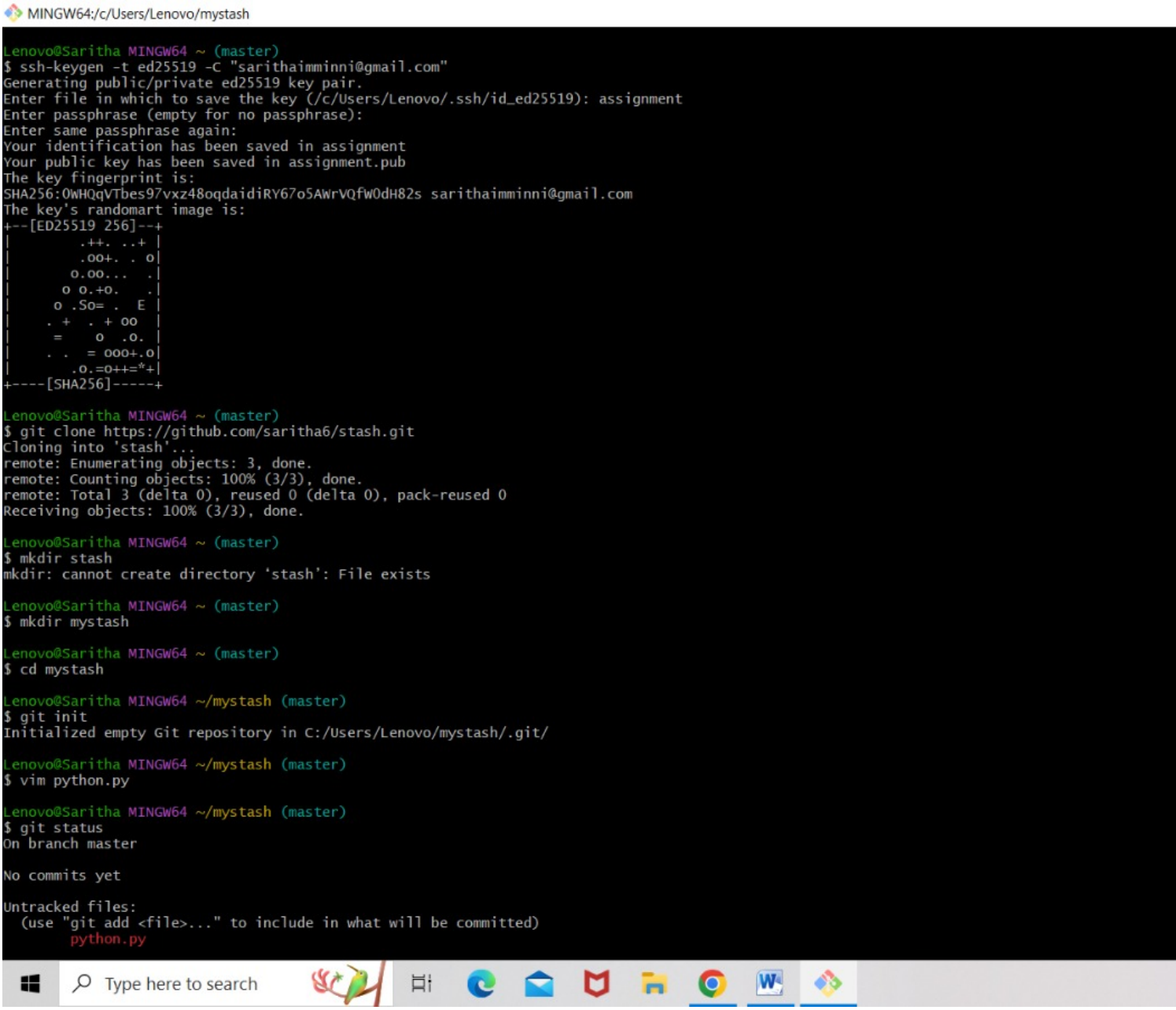
Usage of the git statsh:-

The git stash commad is used to temporarily save changes in a Git repository that are not yet ready to be committed.

The git stash command enables you to switch branches without committing the current branch.

Generally, 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.

Git stash uses STACK data structure.



MINGW64:/c/Users/Lenovo/mystash

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        python.py

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git add .
warning: in the working copy of 'python.py', LF will be replaced by CRLF the next time Git touches

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   python.py


Lenovo@Saritha MINGW64 ~/mystash (master)
$ git commit -m "multiplication"
[master (root-commit) f7c0bde] multiplication
 1 file changed, 4 insertions(+)
 create mode 100644 python.py

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git log --oneline
f7c0bde (HEAD -> master) multiplication

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git branch assignbranch

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git checkout assignbranch
Switched to branch 'assignbranch'

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ vim python.py

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ vim hello.java

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git add .
warning: in the working copy of 'python.py', LF will be replaced by CRLF the next time Git touches
warning: in the working copy of 'hello.java', LF will be replaced by CRLF the next time Git touches

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git status
On branch assignbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.java
```

MINGW64:/c/Users/Lenovo/mystash

```
Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git stash save "first assignment"
Saved working directory and index state On assignbranch: first assignment

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git stash pop
On branch assignbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.java

Dropped refs/stash@{0} (dc4ebf74266325419db6ac401671444cefca8bf0)

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git commit -m "assignment"
[assignbranch 87f01a7] assignment
 1 file changed, 6 insertions(+)
 create mode 100644 hello.java

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git checkout master
Switched to branch 'master'

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git push origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git remote add origin^C

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git remote add origin  https://github.com/saritha6/stash.git

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/saritha6/stash/pull/new/master
remote:
To https://github.com/saritha6/stash.git
 * [new branch]      master -> master

Lenovo@Saritha MINGW64 ~/mystash (master)
$ checkout assignbranch
bash: checkout: command not found

Lenovo@Saritha MINGW64 ~/mystash (master)
```

MINGW64:/c/Users/Lenovo/mystash

```
Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git checkout master
Switched to branch 'master'

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git push origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git remote add originAC

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git remote add origin  https://github.com/saritha6/stash.git

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/saritha6/stash/pull/new/master
remote:
To https://github.com/saritha6/stash.git
 * [new branch]      master -> master

Lenovo@Saritha MINGW64 ~/mystash (master)
$ checkout assignbranch
bash: checkout: command not found

Lenovo@Saritha MINGW64 ~/mystash (master)
$ git checkout assignbranch
Switched to branch 'assignbranch'

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ git push origin assignbranch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 387 bytes | 387.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'assignbranch' on GitHub by visiting:
remote:      https://github.com/saritha6/stash/pull/new/assignbranch
remote:
To https://github.com/saritha6/stash.git
 * [new branch]      assignbranch -> assignbranch

Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$
```

2)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 downloads commits, files, and refs from a remote repository into your local repository.Fetching allows us to download changes from remote repository.But those changes will not be automatically integrated to our working files.

Syntax:- git fetch<remote>

This command fetches branches and history from a specific remote repository.it only updates the remote tracking branchesEg:- git fetch origin

git fetch origin would fetch all changes from the origin remote repository

We can also fetch a specific branch from a remote using the following command.

Syntax:-git fetch<remote><branch>

Eg:-git fetch origin master

Retrive the latest information from the master branch on the origin remote repository.

Process of git fetch command:-

1.Check the status of your local repository with the command git status.this will show you the current state of your local repository.

2.Create a new repository named as git-fetch in github

3.In that repository create two branches named as newBranch, branch2

Git Merge:-

Git merging is basically to merge multiple sequences of commits, stored in multiple branches.

When you merge one branch into the another,Git takes the changes that were made on the source and applies them to the destination branch

Syntax:-git merge <filename>

MINGW64:/c/Users/Lenovo/fetchandmerge

```
Lenovo@Saritha MINGW64 ~/mystash (assignbranch)
$ cd ..

Lenovo@Saritha MINGW64 ~ (master)
$ ssh-keygen -t ed25519 -C "sarithaimminni@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Lenovo/.ssh/id_ed25519): merge
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in merge
Your public key has been saved in merge.pub
The key fingerprint is:
SHA256:qFOs1IMIWClXFgAwqNEEzBlasjMS6GN+sXOPxWSVk7k sarithaimminni@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|&*B++.     ..+   |
|B@+.      o =    |
|@+       +   o   |
|oB o + . o E     |
|o o * * S        |
| . + = +         |
|  . + . o        |
|     .   .       |
|                 |
+----[SHA256]-----+

Lenovo@Saritha MINGW64 ~ (master)
$ git clone https://github.com/saritha6/fetch.git
Cloning into 'fetch'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
Receiving objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Lenovo@Saritha MINGW64 ~ (master)
$ mkdir fetchandmerge

Lenovo@Saritha MINGW64 ~ (master)
$ cd fetchandmerge

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git init
Initialized empty Git repository in C:/Users/Lenovo/fetchandmerge/.git/

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git remote add origin https://github.com/saritha6/fetch.git

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ vim file1

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git add .
warning: in the working copy of 'file1', LF will be replaced by CRLF the next time Git touches it

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git status
On branch master
```

MINGW64:/c/Users/Lenovo/fetchandmerge

```
warning: in the working copy of 'file2', LF will be replaced by CRLF the next time Git touches it

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git status
On branch DevOps
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file2


Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git commit -m "class.py"
[DevOps f258b17] class.py
 1 file changed, 19 insertions(+)
 create mode 100644 file2

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git checkout fetchandmerge
error: pathspec 'fetchandmerge' did not match any file(s) known to git

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git checkout master
Switched to branch 'master'

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 585 bytes | 1024 bytes/s, done.
From https://github.com/saritha6/fetch
 * [new branch]      main       -> origin/main

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git merge DevOps
Updating 16e9a36..f258b17
Fast-forward
 file2 | 19 ++++++++++++++++++
 1 file changed, 19 insertions(+)
 create mode 100644 file2

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 660 bytes | 660.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/saritha6/fetch/pull/new/master
remote:
To https://github.com/saritha6/fetch.git
 * [new branch]      master -> master

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$
```

MINGW64:/c/Users/Lenovo/fetchandmerge

```
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1


Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git commit -m "feach&merge"
[master (root-commit) 16e9a36] feach&merge
 1 file changed, 3 insertions(+)
 create mode 100644 file1

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git branch DevOps

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ checkout DevOps
bash: checkout: command not found

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git checkout DevOps
Switched to branch 'DevOps'

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ vim file2

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git add .
warning: in the working copy of 'file2', LF will be replaced by CRLF the next time Git touches it

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git status
On branch DevOps
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file2


Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git commit -m "class.py"
[DevOps f258b17] class.py
 1 file changed, 19 insertions(+)
 create mode 100644 file2

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git checkout fetchandmerge
error: pathspec 'fetchandmerge' did not match any file(s) known to git

Lenovo@Saritha MINGW64 ~/fetchandmerge (DevOps)
$ git checkout master
Switched to branch 'master'

Lenovo@Saritha MINGW64 ~/fetchandmerge (master)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
```

3)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:-Git fetch downloads the changes from a remote repository to your local repository, but it does not apply those changes to your current working branch.

Instead, it updates your remote tracking branch to reflect any changes that have occurred on the remote repository.

Git fetch is useful when you want to check for changes in a remote repository without merging them into your current working branch.

Git Pull:-

git pull, on the other hand, does both a git fetch and a git merge in one step.

It downloads the changes from a remote repository to your local repository and immediately applies those changes to your current working branch. git pull is useful when you want to update your local branch to the latest changes in a remote branch and immediately see those changes in your working copy.

Steps for git pull:-

Step1:-

At first, check the status of the git repository and check what files are present in.

Step2:-

Add a file in git repository and commit the file

MINGW64:/c/Users/Lenovo/fetching

```
Lenovo@Saritha MINGW64 ~ (master)
$ mkdir fetching

Lenovo@Saritha MINGW64 ~ (master)
$ cd fetching

Lenovo@Saritha MINGW64 ~/fetching (master)
$ git init
Initialized empty Git repository in C:/Users/Lenovo/fetching/.git/

Lenovo@Saritha MINGW64 ~/fetching (master)
$ git remote add origin https://github.com/saritha6/git-fetch-and-git-pull.git

Lenovo@Saritha MINGW64 ~/fetching (master)
$ vim text.txt

Lenovo@Saritha MINGW64 ~/fetching (master)
$ git add .
warning: in the working copy of 'text.txt', LF will be replaced by CRLF the next time Git touches i

Lenovo@Saritha MINGW64 ~/fetching (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   text.txt


Lenovo@Saritha MINGW64 ~/fetching (master)
$ git commit -m "text"
[master (root-commit) 2614b6d] text
 1 file changed, 2 insertions(+)
 create mode 100644 text.txt

Lenovo@Saritha MINGW64 ~/fetching (master)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 601 bytes | 2.00 KiB/s, done.
From https://github.com/saritha6/git-fetch-and-git-pull
 * [new branch]      main       -> origin/main

Lenovo@Saritha MINGW64 ~/fetching (master)
$
```

MINGW64:/c/Users/Lenovo/mydir

```
Lenovo@Saritha MINGW64 ~ (master)
$ mkdir mydir

Lenovo@Saritha MINGW64 ~ (master)
$ cd mydir

Lenovo@Saritha MINGW64 ~/mydir (master)
$ git init
Initialized empty Git repository in C:/Users/Lenovo/mydir/.git/

Lenovo@Saritha MINGW64 ~/mydir (master)
$ git remote add origin https://github.com/saritha6/git-fetch-and-git-pull.git

Lenovo@Saritha MINGW64 ~/mydir (master)
$ vim file1.txt

Lenovo@Saritha MINGW64 ~/mydir (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches

Lenovo@Saritha MINGW64 ~/mydir (master)
$ git commit -m "File2"
[master (root-commit) c4a9172] File2
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

Lenovo@Saritha MINGW64 ~/mydir (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 223 bytes | 223.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/saritha6/git-fetch-and-git-pull/pull/new/master
remote:
To https://github.com/saritha6/git-fetch-and-git-pull.git
 * [new branch]      master -> master

Lenovo@Saritha MINGW64 ~/mydir (master)
$ git pull origin master
From https://github.com/saritha6/git-fetch-and-git-pull
 * branch            master     -> FETCH_HEAD
Already up to date.

Lenovo@Saritha MINGW64 ~/mydir (master)
$
```

4)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:-

awk is a powerful command-line tool used for processing and manipulating text files , especially when dealing with large amounts of data.

In the below image, created a file and named as Data.txt and print the same data

Command on awk:-

Syntax:- awk '{print}' filename

This commad prints the data present in the file

Command:- awk '{print$column_number}' filename

Eg:-awk '{print$2}' Data.txt

This command prints the second column data in a data.txt file

Command:- awk '{print$1,$4}' Data.txt

This command prints the first and fourth column data in a Data.txt filek

BASH SCRIPTING:-

Steps to follow in bash Scripting

Step1:

Create a file with extension .sh

Step2:-

Give the permissions of read, write and excute

Step3:-

open the shell and write the script

Step4:-

Save the code and run the code

The command to run a code is

Syntax:- bash filename

root@fbb0b3f32c3a: /

```
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>docker run -it ubuntu
root@fbb0b3f32c3a:/# cat >> AWKfile.txt
Hello,
     This is my AWK file.
^C
root@fbb0b3f32c3a:/# cat >> PrimeNumbers.sh
^V#!/bin/bash

for i in $(seq 1 20); do
    is_prime=1
    for j in $(seq 2 $((i-1))); do
        if [[ $(($i % $j)) == 0 ]]; then
            is_prime=0
            break
        fi
    done
    if [[ $is_prime == 1 ]]; then
        echo $i
    fi
done^C
root@fbb0b3f32c3a:/# bash PrimeNumbers.sh
PrimeNumbers.sh: line 1: ▬#!/bin/bash: No such file or directory
1
2
3
5
7
11
13
17
19
root@fbb0b3f32c3a:/# history
    1  cat >> AWKfile.txt
    2  cat >> PrimeNumbers.sh
    3  bash PrimeNumbers.sh
    4  history
root@fbb0b3f32c3a:/# _
```

5)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.

Steps to set up a container and run a ubuntu operating system.

Step1:- Install docker image from a google and set the docker image according to your machine

Step2:-To check the weather the docker installed correctly in your machine excute the below command in the command prompt.

Command:- docker version

If you get the description and version about the docker then it installed correctly.

Otherwise again install the docker in your machine

Step3:-Pull the ubuntu image from the docker image by running the below commad:

Command:- docker pull ubuntu

Step4:-After the image was downloaded, run the container using the following commad

Command:- docker run -it ubuntu

Select root@1a07f276bb73: /

```
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
677076032cca: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\Lenovo>docker run -it ubuntu
root@1a07f276bb73:/#
```