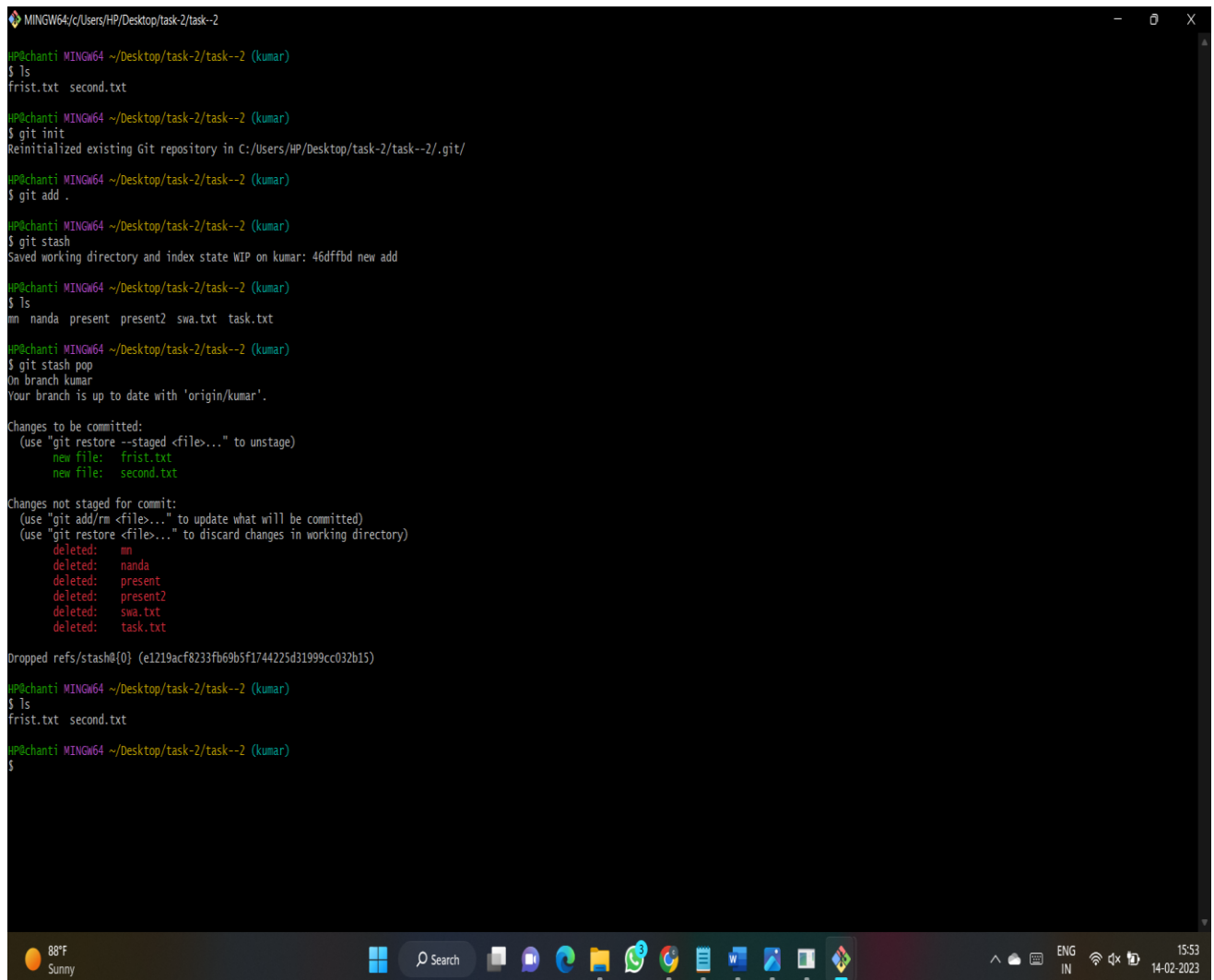


Q1. Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.

Git stash: git stash is used to store the data temporarily and safely without committing.

when we are working on a branch sometimes we need to switch the branch without completion on current work and we don't want to commit incomplete work. so here the "git stash" come to action and stores the work in hidden place without any commit.



```
MINGW64~/c/Users/HP/Desktop/task-2/task--2
HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ ls
frist.txt  second.txt

HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ git init
Reinitialized existing Git repository in C:/Users/HP/Desktop/task-2/task--2/.git/

HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ git add .

HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ git stash
Saved working directory and index state WIP on kumar: 46dffb new add

HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ ls
mn  nanda  present  present2  swa.txt  task.txt

HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ git stash pop
On branch kumar
Your branch is up to date with 'origin/kumar'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   frist.txt
    new file:   second.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    mn
    deleted:    nanda
    deleted:    present
    deleted:    present2
    deleted:    swa.txt
    deleted:    task.txt

Dropped refs/stash@{0} (e1219acf8233fb69b5f1744225d31999cc032b15)

HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$ ls
frist.txt  second.txt

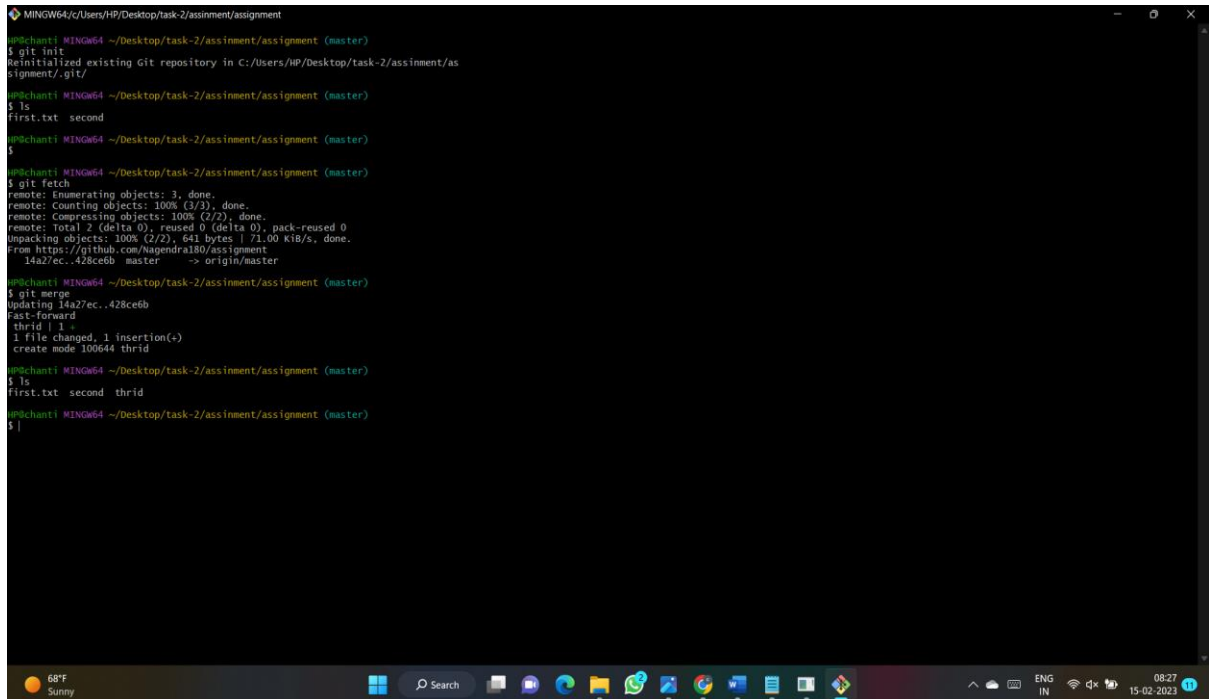
HP@chanti MINGW64 ~/Desktop/task-2/task--2 (kumar)
$
```

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>> git fetch is used to pull the all updates form remote repository

by using git fetch command we pull the all updates from remote repository

>>frist in “git fetch” tells the updates are in remote repository.



```
MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ git init
Reinitialized existing Git repository in C:/Users/HP/Desktop/task-2/assinment/assignment/.git/

MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ ls
first.txt  second

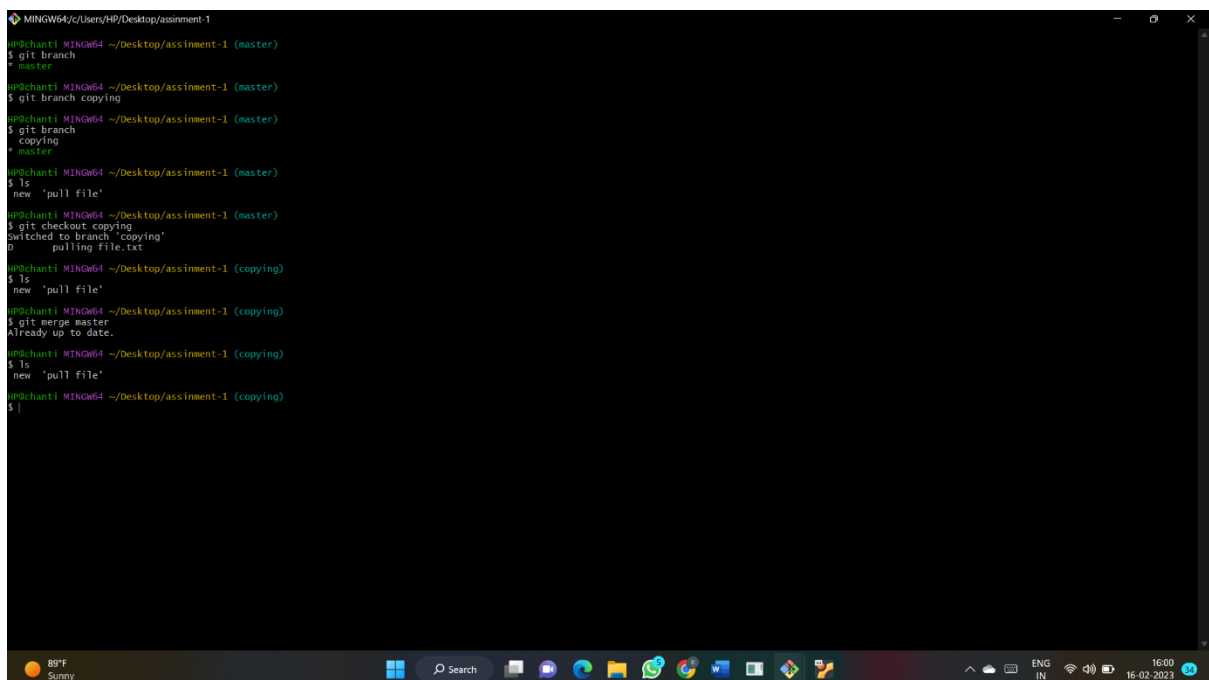
MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ git fetch
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 641 bytes | 71.00 KiB/s, done.
From https://github.com/nagendra180/assignment
 14a27ec..428ce6b  master    -> origin/master

MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ git merge
Updating 14a27ec..428ce6b
Fast-forward
 thrid | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 thrid

MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ ls
first.txt  second  thrid

MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ |
```

>>Git merge is used to copying the data from one branch to another branch



```
MINGW64 ~/Desktop/assinment-1 (master)
$ git branch
* master

MINGW64 ~/Desktop/assinment-1 (master)
$ git branch copying

MINGW64 ~/Desktop/assinment-1 (master)
$ git branch
* master
  copying

MINGW64 ~/Desktop/assinment-1 (master)
$ ls
new 'pull file'

MINGW64 ~/Desktop/assinment-1 (master)
$ git checkout copying
Switched to branch 'copying'
0   pulling file.txt

MINGW64 ~/Desktop/assinment-1 (copying)
$ ls
new 'pull file'

MINGW64 ~/Desktop/assinment-1 (copying)
$ git merge master
Already up to date.

MINGW64 ~/Desktop/assinment-1 (copying)
$ ls
new 'pull file'

MINGW64 ~/Desktop/assinment-1 (copying)
$ |
```

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 pull==git fetch+ git merge

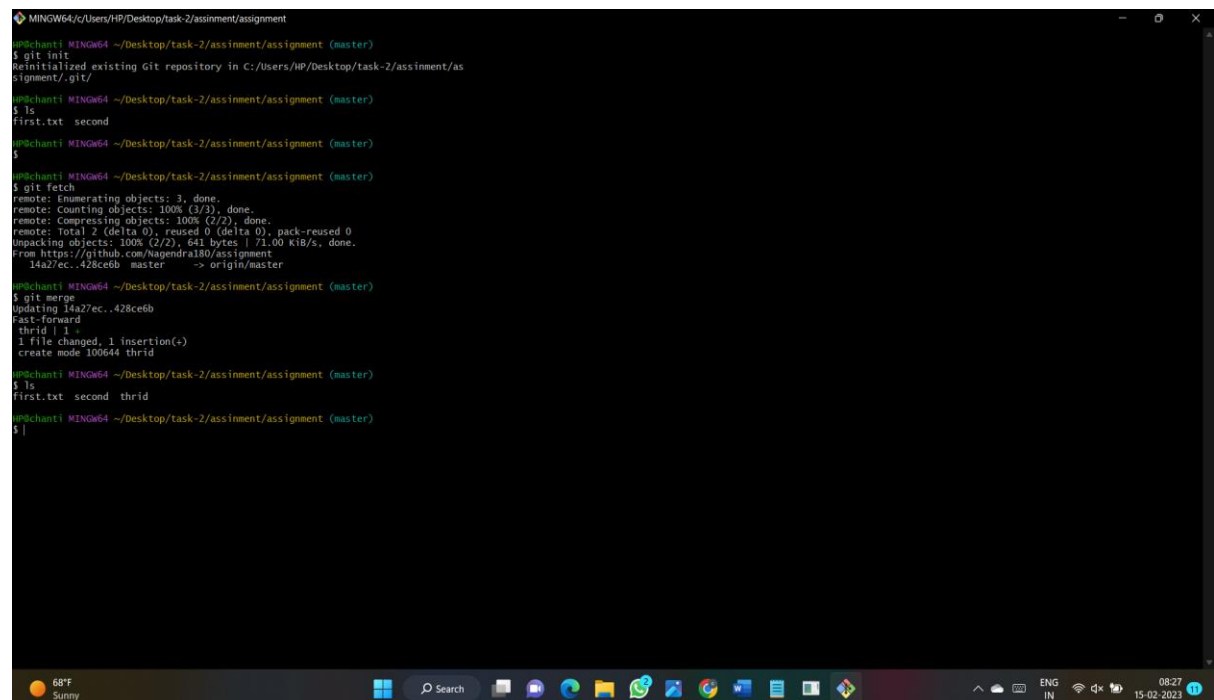
git fetch tells the there are changes available in remote repository without bringing the changes in local repository

But git pull brings the copy of files in remote repository to local repository

after the git fetch git merge command is used to copying the files form remote repository to local repository

Fetch never manipulates or integrates data. Pull downloads the data and integrates it with the current data.

Git fetch + git merge



```
MINGW64 ~/Desktop/task-2/assinment/assignment
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ git init
Reinitialized existing Git repository in C:/Users/HP/Desktop/task-2/assinment/as
signment/.git/
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ ls
first.txt  second
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ git fetch
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 641 bytes | 71.00 KiB/s, done.
From https://github.com/nagendra180/assignment
14a27ec..428ceb master -> origin/master
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ git merge
Updating 14a27ec..428ceb
Fast-forward
 thrid | 1
 1 file changed, 1 insertion(+)
 create mode 100644 thrid
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ ls
first.txt  second  thrid
@chanti MINGW64 ~/Desktop/task-2/assinment/assignment (master)
$ |
```

Git pull

```
MINGW64/c/Users/HP/Desktop/assiment-1
git@chanti: MINGW64 ~/Desktop/assiment-1 (master)
$ ls
new

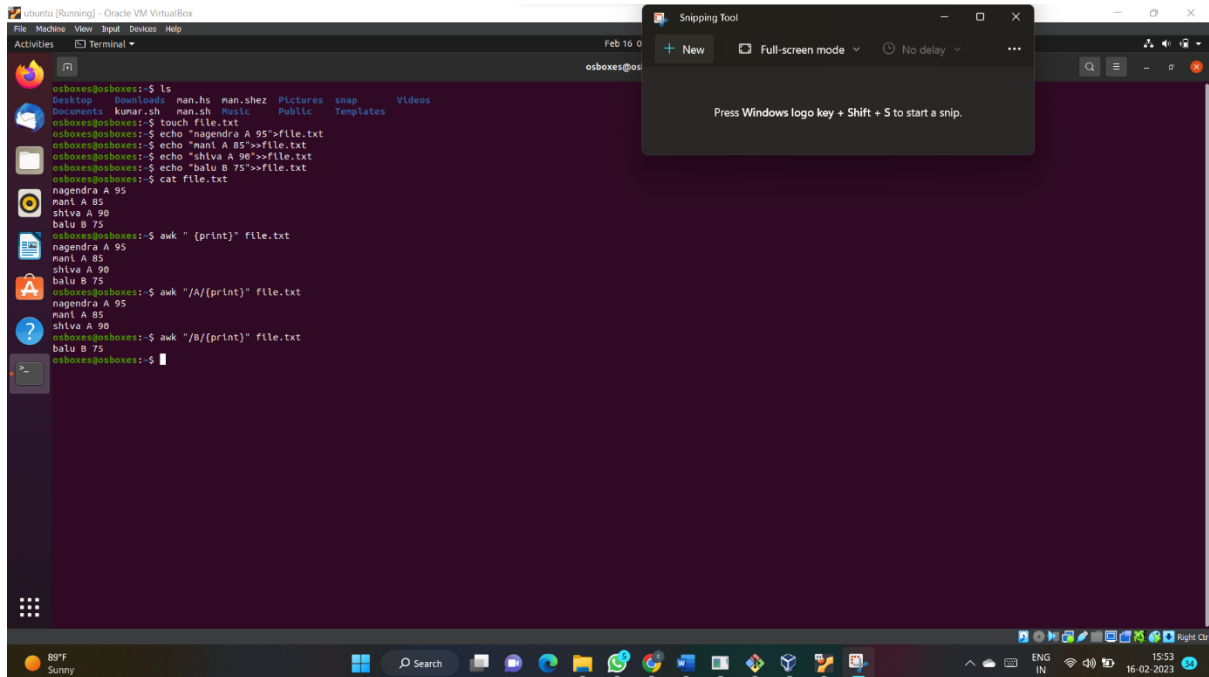
git@chanti: MINGW64 ~/Desktop/assiment-1 (master)
$ git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 641 bytes | 64.00 KiB/s, done.
From https://github.com/naigendra180/assiment
+ branch      master      -> FETCH_HEAD
31b3e51..a6fc9a0 master    -> origin/master
Fast-forward
  pull file | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 pull file

git@chanti: MINGW64 ~/Desktop/assiment-1 (master)
$ ls
new
'pull file'

git@chanti: MINGW64 ~/Desktop/assiment-1 (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 awk command is a linux tool and programming language that allows users to process and manipulate data.



The screenshot shows a terminal window with the following commands and output:

```
osboxes@osboxes:~$ ls
Desktop  Downloads  man.hs  man.sh  man.shez  Pictures  snap  Videos
Documents  kunar.sh  man.sh  Music  Public  Templates

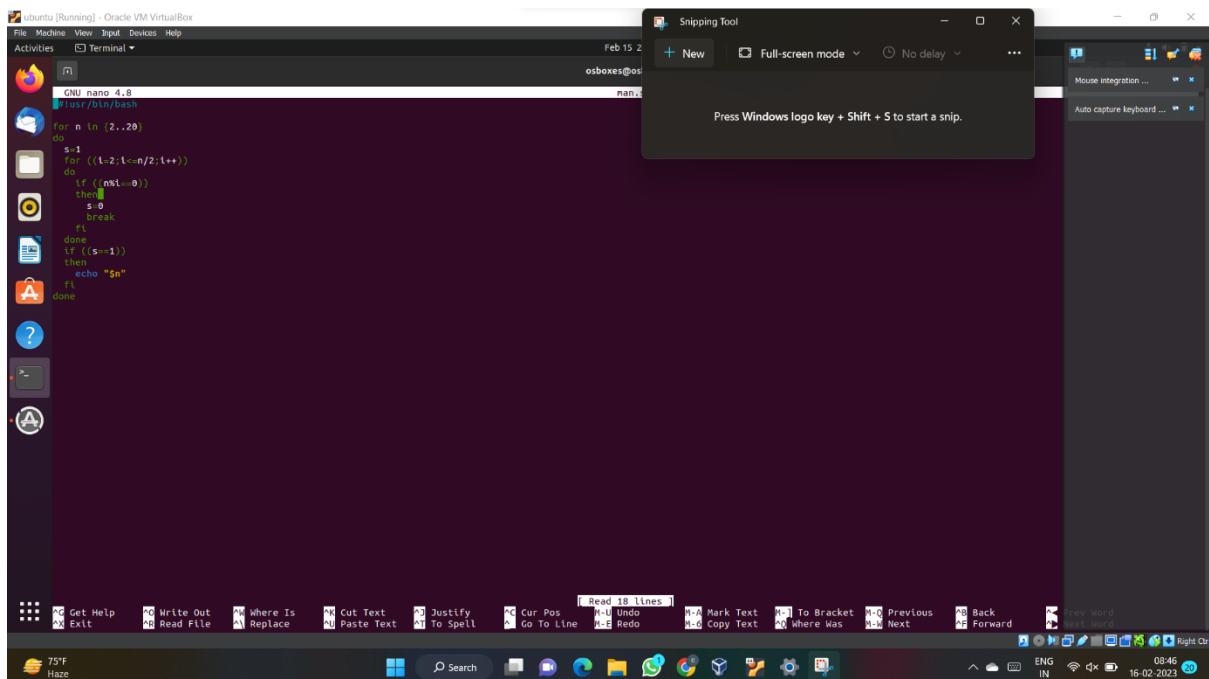
osboxes@osboxes:~$ touch file.txt
osboxes@osboxes:~$ echo "nagendra A 95">>file.txt
osboxes@osboxes:~$ echo "manj A 85">>file.txt
osboxes@osboxes:~$ echo "shiva A 90">>file.txt
osboxes@osboxes:~$ echo "balu B 75">>file.txt
osboxes@osboxes:~$ cat file.txt
nagendra A 95
manj A 85
shiva A 90
balu B 75

osboxes@osboxes:~$ awk '{print}' file.txt
nagendra A 95
manj A 85
shiva A 90
balu B 75

osboxes@osboxes:~$ awk '/A/{print}' file.txt
nagendra A 95
manj A 85
shiva A 90

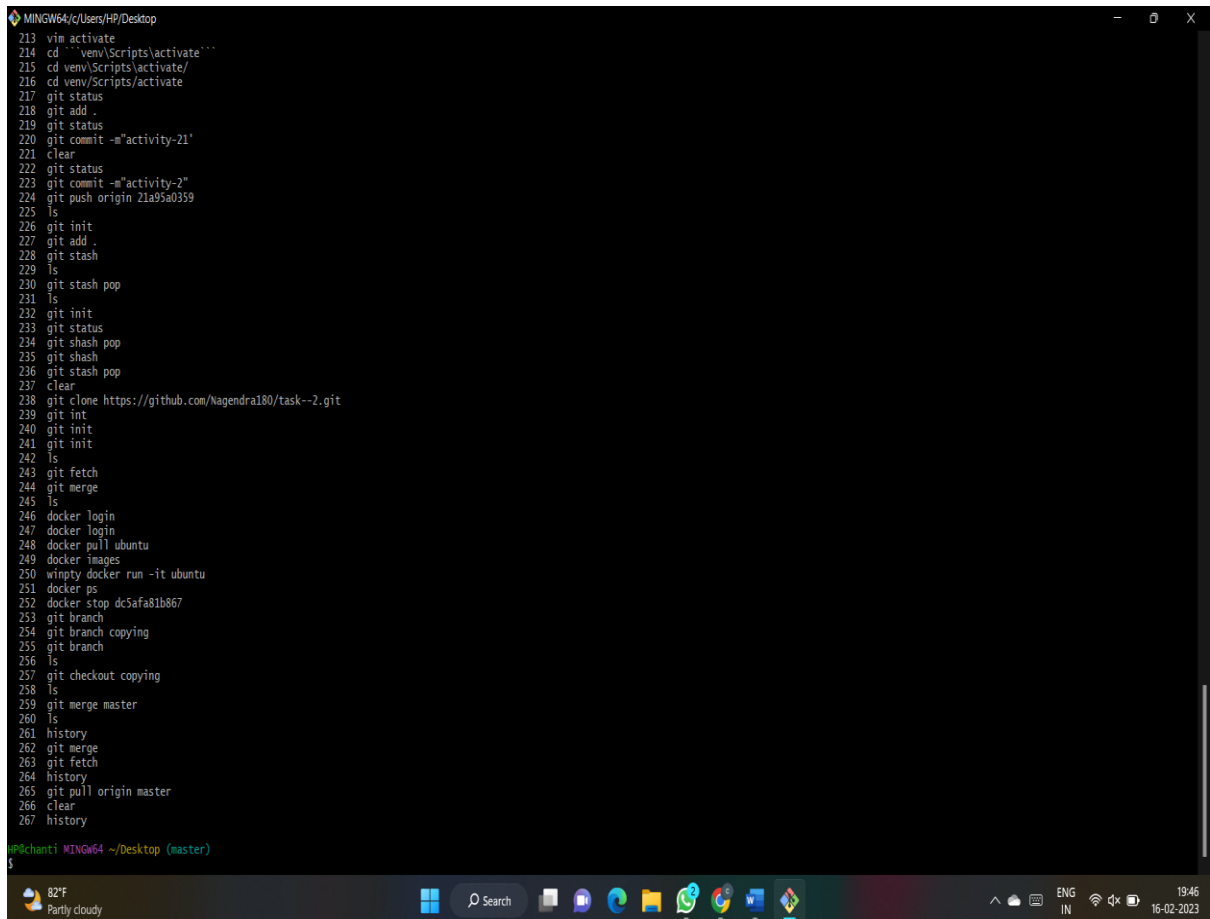
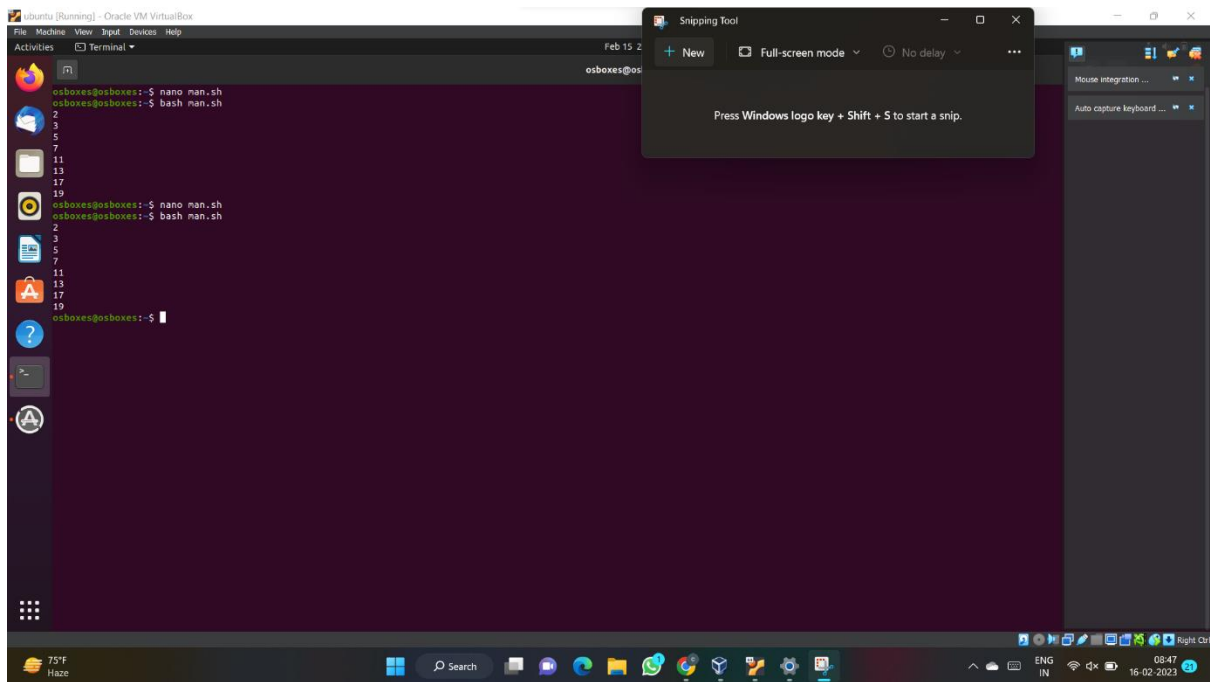
osboxes@osboxes:~$ awk '/B/{print}' file.txt
balu B 75
```

Program to print prime number in range 1-20



The screenshot shows a terminal window with the following bash script:

```
#!/usr/bin/bash
for n in {1..20}
do
    s=1
    for ((l=2;l<=n/2;l++))
    do
        if ((n%l==0))
        then
            s=0
            break
        fi
    done
    if ((s==1))
    then
        echo "$n"
    fi
done
```



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.

Frist check we are login are not in docker desktop by using command “docker login”

If not login with your credentials

>>Pull the image ubuntu images. **”docker pull ubuntu”**

>>to run the container in interactive mode .**”winpty docker run -it ubuntu”**

```
root@dcsafa81b867:/
winchanti MINGW64 ~/Desktop (master)
$ docker login
Error: Cannot perform an interactive login from a non TTY device

winchanti MINGW64 ~/Desktop (master)
$ docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

winchanti MINGW64 ~/Desktop (master)
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:9408d4e41880896a37280fbc2384015e90e3f84906b750c1a53539b585fbb7f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

winchanti MINGW64 ~/Desktop (master)
$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
pythonn             latest      92e276cb2067  25 hours ago  156MB
nigendra180/pythonn latest      92e276cb2067  25 hours ago  156MB
chanti              latest      aa12b7aa0ee6  2 days ago   919MB
nigendra180/chanti latest      aa12b7aa0ee6  2 days ago   919MB
ubuntu              latest      92e276cb2067  25 hours ago  156MB

winchanti MINGW64 ~/Desktop (master)
$ winpty docker run -it ubuntu
root@dcsafa81b867:/#
root@dcsafa81b867:/#
root@dcsafa81b867:/#
```

>> to stop the the image **“docker ps”** and copy the image id. And stop the images by commad **“docker stop id”**

```
MINGW64 ~/Desktop
winchanti MINGW64 ~/Desktop (master)
$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS
dcsafa81b867  ubuntu   "/bin/bash"  52 seconds ago  up 50 seconds

winchanti MINGW64 ~/Desktop (master)
$ AC

winchanti MINGW64 ~/Desktop (master)
$ docker stop dcsafa81b867
dcsafa81b867

winchanti MINGW64 ~/Desktop (master)
$
```

```
MINGW64/c/Users/HP/Desktop
nagchamt! MINGW64 ~/Desktop (master)
$ docker login
Error: Cannot perform an interactive login from a non TTY device

nagchamt! MINGW64 ~/Desktop (master)
$ docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

nagchamt! MINGW64 ~/Desktop (master)
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:9a0b0d4e188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

nagchamt! MINGW64 ~/Desktop (master)
$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
python              latest      92e276cb2067  25 hours ago  156MB
nagendra180/python  latest      92e276cb2067  25 hours ago  156MB
chanti              latest      aal2b7aa0ee6  2 days ago    919MB
nagendra180/chanti  latest      aal2b7aa0ee6  2 days ago    919MB
ubuntu              latest      58db3edaf2be  2 weeks ago   77.8MB

nagchamt! MINGW64 ~/Desktop (master)
$ winpty docker run -it ubuntu
root@dc5afa81b867:/#
root@dc5afa81b867:/#
root@dc5afa81b867:/#
nagchamt! MINGW64 ~/Desktop (master)
$ |
```