Assignment 1: (Complex)

1. Commit a repository with a single file change.

2. Create 2 new directories in local PC and clone the repository on both directories.

3. Make changes in one of the directories and commit and merge the changes.

4. Parallelly make a change in the same file in another directory and when the changes are pushed, there should be merge conflict noticed which should be resolved and pushed changes again.

5. Merge the changes

Note:

Share the repository name along with the output/screen shot of the history of commands executed.

Capture the commit history after each commit.

Solution:

Step 1: Commit a repository with a single file change

Step 2: Create 2 new directories in local PC and clone the repository on both directories.

```
Gowtham@DESKTOP-TP2JR11 MINGW64 ~/my-repo (master)
$ cd ..

Gowtham@DESKTOP-TP2JR11 MINGW64 ~
$ mkdir clone1 clone2

Gowtham@DESKTOP-TP2JR11 MINGW64 ~
$ git clone ./my-repo clone1
Cloning into 'clone1'...
done.

Gowtham@DESKTOP-TP2JR11 MINGW64 ~
$ git clone ./my-repo clone2
Cloning into 'clone2'...
done.
```

```
Gowtham@DESKTOP-TP2JR11 MINGW64 ~
$ cd clone1

Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone1 (master)
$ echo "Change from clone1" >> file.txt

Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone1 (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
 time Git touches it

Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone1 (master)
$ git commit -m "Change from clone1"
[master d3125cc] Change from clone1
 1 file changed, 1 insertion(+)
```

Step 3: Make changes in one of the directories and commit and merge the changes

```
Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone2 (master)
$ git pull ../clone1 master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 263 bytes | 11.00 KiB/s, done.
From ../clone1
 * branch            master     -> FETCH_HEAD
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.

Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone2 (master|MERGING)
$ nano file.txt
```

Step 4: Parallelly make a change in the same file in another directory and when the changes are pushed, there should be merge conflict noticed which should be resolved and pushed changes again.

Step 5: Merge the changes

```
Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone2 (master)
$ git push ../my-repo master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 799 bytes | 399.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: error: refusing to update checked out branch: refs/heads/master
remote: error: By default, updating the current branch in a non-bare repository
remote: is denied, because it will make the index and work tree inconsistent
remote: with what you pushed, and will require 'git reset --hard' to match
remote: the work tree to HEAD.
remote:
remote: You can set the 'receive.denyCurrentBranch' configuration variable
remote: to 'ignore' or 'warn' in the remote repository to allow pushing into
remote: its current branch; however, this is not recommended unless you
remote: arranged to update its work tree to match what you pushed in some
remote: other way.
remote:
remote: To squelch this message and still keep the default behaviour, set
remote: 'receive.denyCurrentBranch' configuration variable to 'refuse'.
```

History of commits

```
MINGW64:/c/Users/gtwis/clone2

Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone2 (master)
$ git log
commit 19eda02230df1057c38a508a31afd49ed5b62b26 (HEAD -> master)
Merge: 262bdbd d3125cc
Author: Soundariya <ksoundariya19@gmail.com>
Date:   Fri Jun 7 09:59:55 2024 +0530

    Resolved merge comflict

commit 262bdbd079978d4710c70feb0ec1843c9bb034e2
Author: Soundariya <ksoundariya19@gmail.com>
Date:   Fri Jun 7 09:58:07 2024 +0530

    Conflicting change from clone2

commit d3125cc2a55286c2397ac18ae98a09d04e21a490
Author: Soundariya <ksoundariya19@gmail.com>
Date:   Fri Jun 7 09:57:10 2024 +0530

    Change from clone1

commit 3c0019b82bbd83d42793404de818dd312e623df6 (origin/master, origin/HEAD)
Author: Soundariya <ksoundariya19@gmail.com>
Date:   Fri Jun 7 09:55:23 2024 +0530

    Initial commit with a single line

Gowtham@DESKTOP-TP2JR11 MINGW64 ~/clone2 (master)
```

Assignment 2: (Simple)

Create a story.txt file using below lines, save and quit the file

===

The quick brown fox jumps over the lazy dog.

The quick brown fox is fast and clever.

Lazy dogs often lie in the sun all day.

The dog chased the fox through the field.

A quick fox can outsmart a lazy dog.

Dogs are loyal and friendly animals.

Foxes are cunning and quick in nature.

The field was green and full of life.

The fox found food in the field.

Dogs and foxes rarely become friends.

===
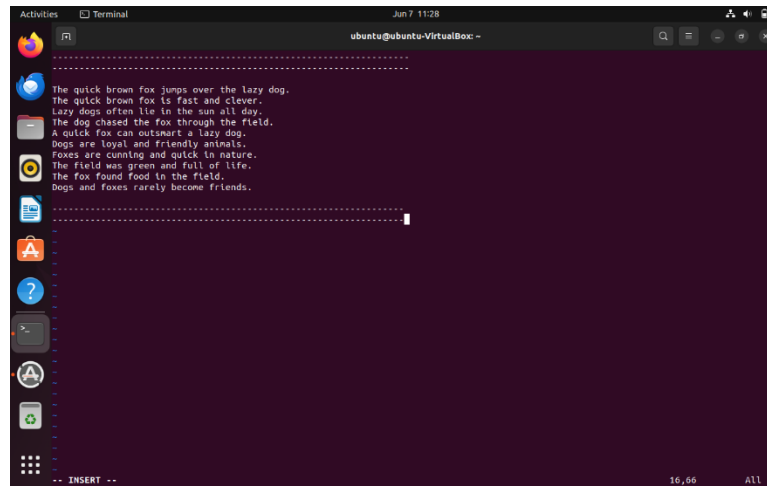
Open story.txt file and do the following,

1. Replace fox with wolf, save and quit the file. Open and see if the changes gets reflected.

2. Search for string "field" in the file and move to next and prev occurence of the string.

3. Copy the entire line "A quick fox can outsmart a lazy dog" and paste at the end of file and save it.

4. Replace all the vowels with letter "x" in the file.

Note: Use only commands to do above operations and share the commands, screenshots of the file while and after the execution of the command.
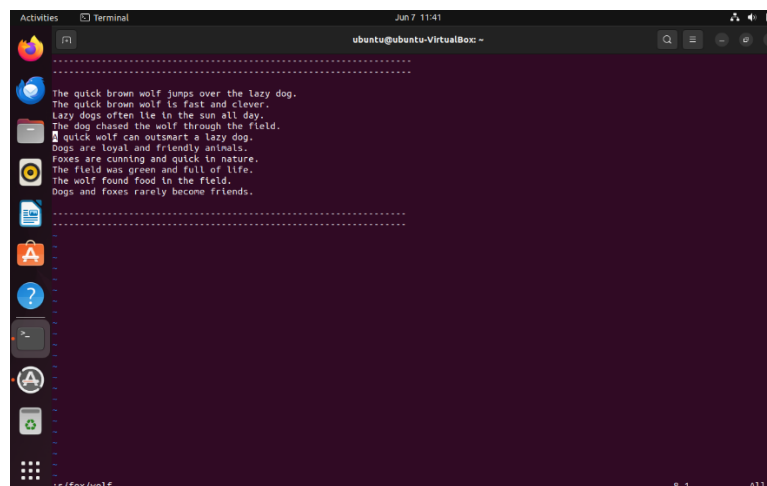
Solution:

story.txt

Step 1: Replace fox with wolf, save and quit the file. Open and see if the changes gets reflected
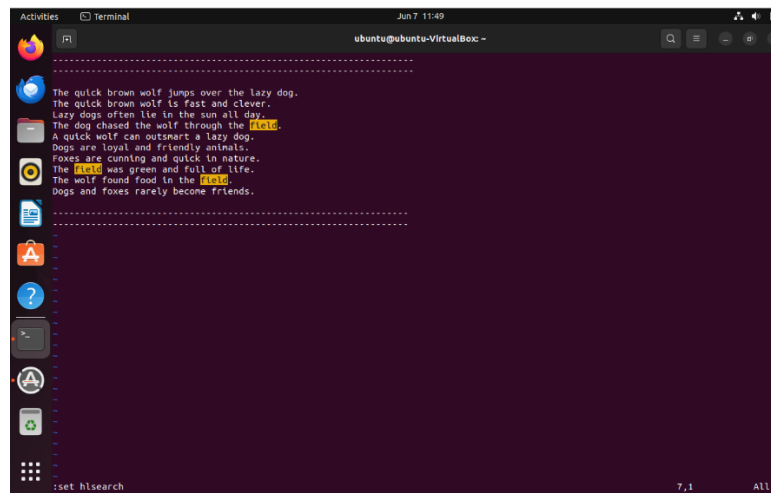
:s/fox/wolf

Step 2: Search for string "field" in the file and move to next and prev occurence of the string

:s/field

:set hlsearch

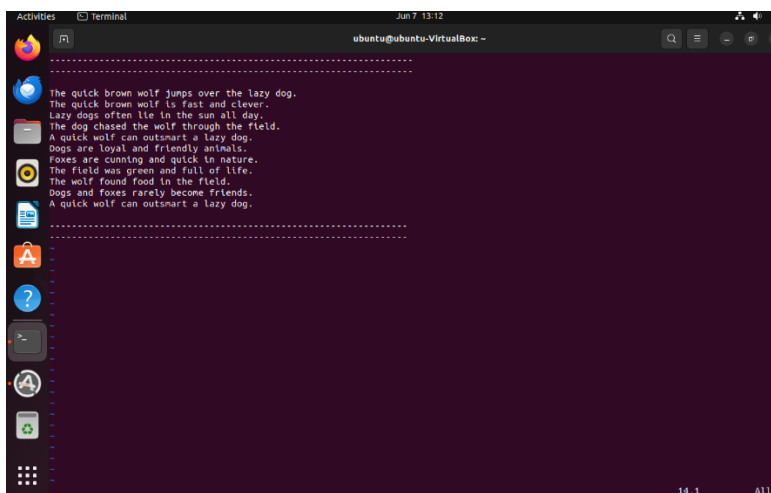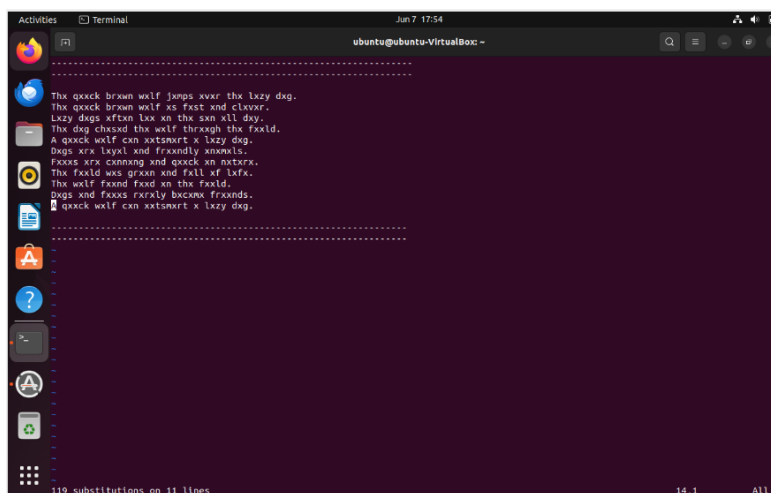n-next occurrences in forward direction

N-next occurrences in reverse direction

Step 3: Copy the entire line "A quick fox can outsmart a lazy dog" and paste at the end of file and save it.

Copy-yy

Paste-p

Step 4: Replace all the vowels with letter "x" in the file

:%s/a\|A\|e\|E\|i\|I\|o\|O\|u\|U/x/g