**Git**

1. Suppose you had a file, called first.md, and you made a copy of this file, named it second.md and made some changes to it. Next, suppose you ran diff -u first.md second.md.

Here is the content of the original first.md

A

B

C

D

E

F

Here is the output of the diff command:

Text

Description automatically generated with low confidenceWhat is the content of second.md?

Second.md should be:

A

B

$

C

#

%

E

F

1. (True or False) If you accidentally add a file to the staging area, you can remove it using git reset. For example, if you accidentally add thrid.md, but don’t want it to be committed yet, run git reset thrid.md and the file will be removed from the staging area, **but it will still be in your working directory**.

Ture

1. (True or False) The commands git reset and git revert can only be used to undo commits in the git repository.

Flase

git reset is to undo changes in the staging area.

git revert is to create a new commit that undo the changes from previous commit.

1. (True or False) The commands git checkout can be used to roll back to a certain commit hash (check the documentation if you are unsure).

Ture, git checkout <commit-hash>

1. (True or False) We cannot commit changes in the working directory directly to the repo without adding it to the staging index first (read the documentation if you are unsure).

Ture, we need git add first then git commit.

1. (True or False) git log -p and git log will give you the same output.

Flase, git log –p will show detail diffs between each commit.

1. (True or False) git log --oneline and git log --stat will give you the same output.

Flase, git log –-oneline shows commit logs in a single line,

git log –stat shows commit logs along with statistics.

1. (True or False) It is recommended that in most cases we should use git revert rather than git reset to undo commits because git revert is safer.

Ture, git revert without change commit history, git reset will modify commit history.