

Solution of assignment 1 (git intro)

1.1: Your first git repository (3 points)

Example solution:

```
$ git clone <repository_url> <destination_folder>
$ cd <destination_folder>
$ echo 'Hello world' > Readme.txt
$ git add Readme.txt
$ git commit -m 'My first commit!'
$ git push
```

To correct, look for a file named “Readme.txt” in the repository containing “Hello world”.

1.2: Tagging (2 points)

Example solution: (to find commit hash, use e.g. `git log`)

```
$ git tag week1_submission COMMITHASH
$ git push origin week1_submission
```

To correct, enter the repository and use “git tag -l” to see if the tag “week1_submission” exists. 2

1.2: Getting back old versions of files (optional)

Example solution:

```
$ echo "Hei! :)" > greeting.txt
$ git add greeting.txt
$ git commit -m "Made friendly greeting."
$ echo "Hei! >:(" > greeting.txt
$ git add greeting.txt
$ git commit -m "Made greeting less friendly."
$ git log --oneline # note hash of friendly commit
$ git checkout COMMITHASH greeting.txt
$ git add greeting.txt # not strictly speaking needed
$ git commit -m "Made greeting friendly again."
```

Don't bother correcting this, because it's optional.

1.4: Resolve git conflicts (optional)

Example solution:

```
$ cd repo_1
$ echo "no conflict" > gitconflict.txt
$ git add gitconflict.txt
```

```

$ git commit -m "Prepared to cause a conflict."
$ git push
$ cd ..
$ git clone <repository_url> <reponame>_copy
$ cd <reponame>_copy
$ echo "A change" > gitconflict.txt
$ git add gitconflict.txt
$ git commit -m "Made a change."
$ git push
$ cd ../<reponame>
$ echo "Another change" > gitconflict.txt
$ git add gitconflict.txt
$ git commit -m "Made another change."
$ git push                # this fails
$ git pull                # automatic merge will fail
$ emacs gitconflict.txt # resolve conflict and save
$ git add gitconflict.txt
$ git commit -m "Resolved merge conflict."
$ git push

```

Don't bother correcting this, because it's optional.