```
$ echo "hello world" > test_dir_1/test_file_2.txt
$ mkdir test_dir_2
$ echo "hello world" > test_dir_2/test_file_3.txt
```

And then run your program like this:

```
$ /path/to/your_git.sh write-tree
4b825dc642cb6eb9a060e54bf8d69288fbee4904
```

You're expected to write the entire working directory as a tree object and print the 40-char SHA to stdout.

The tester will verify that the output of your program matches the SHA hash of the tree object that the official `git` implementation would write.

Notes

- Remember to ignore the `.git` directory when creating entries in the tree object.
- Your implementation of `git write-tree` will need to handle nested directories. A recursive implementation will help here, since you'll need to create tree objects for each subdirectory to be able to create the parent directory's tree object.
- The implementation of `git write-tree` here differs slightly from the official `git` implementation. The official `git` implementation uses the staging area to determine what to write to the tree object. We'll just assume that all files in the working directory are staged.

View Code Examples | View Screencasts | Test Cases

Collapse ↑

Tests failed. Show logs