The `git commit-tree` command creates a commit object. Example usage:

```
# Create a new directory and cd into it
$ mkdir test_dir && cd test_dir

# Initialize a new git repository
$ git init
Initialized empty Git repository in /path/to/test_dir/.git/

# Create a tree, get its SHA
$ echo "hello world" > test.txt
$ git add test.txt
$ git write-tree
4b825dc642cb6eb9a060e54bf8d69288fbee4904

# Create the initial commit
$ git commit-tree 4b825dc642cb6eb9a060e54bf8d69288fbee4904 -m "Initial commi
3b18e512dba79e4c8300dd08aeb37f8e728b8dad

# Write some changes, get another tree SHA
$ echo "hello world 2" > test.txt
$ git add test.txt
$ git write-tree
5b825dc642cb6eb9a060e54bf8d69288fbee4904

# Create a new commit with the new tree SHA
$ git commit-tree 5b825dc642cb6eb9a060e54bf8d69288fbee4904 -p 3b18e512dba79e
```

The output of `git commit-tree` is the 40-char SHA hash of the commit object that was written to `.git/objects`.

Tests

Your program will be invoked like this:

```
$ ./your_git.sh commit-tree <tree_sha> -p <commit_sha> -m <message>
```

Tests failed. Show logs