

## Python Passwords

The Facebook Security team decided to “accept several forms of a user’s password to help overcome the most common reason that authentic logins are rejected. In addition to the original password, we also accept the password if a user inadvertently has the caps lock enabled, if their mobile device automatically capitalizes the first character of the password, or if an extra character is added to the beginning of the password.”

Write a Python program called “pass.py” that accepts exactly two positional arguments from the command-line: a password and an alternative version. If not provided two arguments, print a “usage” statement:

```
$ ./pass.py
Usage: pass.py PASSWORD ALT
```

If:

1. The password and the alternate are exactly the same
2. The alternate is the same as the password with the first letter capitalized
3. The password and the alternate are the same if compared as both upper-case
4. The password is the same as the alternate if you add a single character (anything printable!) to beginning and/or end

print “ok”; otherwise, print “nah”.

```
$ ./pass.py foo foo
ok
$ ./pass.py foo Foo
ok
$ ./pass.py foo F00
ok
$ ./pass.py foo 'foo'
ok
$ ./pass.py foo 'foo '
ok
$ ./pass.py foo 'foo '
ok
$ ./pass.py foo bar
nah
```

## Test Suite

A passing test suite looks like this:

```
$ pytest -v test.py
```

```

===== test session starts =====
platform darwin -- Python 3.6.8, pytest-4.2.0, py-1.7.0, pluggy-0.8.1 -- /anaconda3/bin/python
cachedir: .pytest_cache
rootdir: /Users/kyclark/work/worked_examples/12-regex-passwords, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 8 items

test.py::test_usage PASSED [ 12%]
test.py::test_reject PASSED [ 25%]
test.py::test_accept_01 PASSED [ 37%]
test.py::test_accept_02 PASSED [ 50%]
test.py::test_accept_03 PASSED [ 62%]
test.py::test_accept_04 PASSED [ 75%]
test.py::test_accept_05 PASSED [ 87%]
test.py::test_accept_06 PASSED [100%]

===== 8 passed in 0.43 seconds =====

```