

Table of Contents

[tl;dr](#)

[Specification](#)

[Password List](#)

[Walkthrough](#)

[Usage](#)

[Testing](#)

[Correctness](#)

[Style](#)

[How to Submit](#)

Crack

tl;dr

Implement a program that cracks passwords, per the below.

```
$ python crack.py 51v3Nh6ZWGHOQ
ROFL
```

Specification

Design and implement a program, `crack`, that cracks passwords, exactly as you did in [Problem Set 2 \(https://lab.cs50.io/cs50/labs/2019/x/crack/\)](https://lab.cs50.io/cs50/labs/2019/x/crack/), except that your program this time should be written (a) in Python and (b) in CS50 IDE..

- Implement your program in a file called `crack.py` in your `~/workspace/pset6/crack` directory (if it doesn't already exist, create it now!).

- Your program should accept a single command-line argument: a hashed password.
- If your program is executed without any command-line arguments or with more than one command-line argument, your program should print an error (of your choice) and exit immediately.
- Otherwise, your program must proceed to crack the given password, ideally as quickly as possible, ultimately printing the password in the clear followed by `\n`, nothing more, nothing less.
- Assume that each password has been hashed with C's DES-based (not MD5-based) `crypt` function.
- Assume that each password is no longer than five (5) characters. Gasp!
- Assume that each password is composed entirely of alphabetical characters (uppercase and/or lowercase).

Password List

```
brian:51.xJagtPnb6s
bjbrown:50GApilQSG3E2
emc:502sDZxA/ybHs
greg:50C6B0oz0HWzo
jana:50WUNAFdX/yjA
lloyd:50n0AAUD.pL8g
malan:50CcfIk1QrPr6
natmelo:50JIIyhDORqMU
rob:51v3Nh6ZWGHOQ
veronica:61v1CDwwP95bY
walker:508ny6Rw0aRio
zamyla:50cI2vYkF0YU2
```

Walkthrough

Note: The walkthrough video incorrectly states that passwords are no longer than four (4) characters. Instead, per the specification, make sure to handle passwords that are up to five (5) characters.

crack (Python)



Usage

Your program should behave per the examples below. Assume that the underlined text is what some user has typed.

```
$ python crack.py
```

```
Usage: python crack.py hash
```

```
$ python crack.py 51v3Nh6ZWGHOQ
```

```
ROFL
```

Testing

Correctness

No `check50` for this one! But odds are, if you can crack all twelve passwords above, you're in good shape!

Style

```
style50 crack.py
```

How to Submit

Execute the below, logging in with your GitHub username and password when prompted. For security, you'll see asterisks (*) instead of the actual characters in your password.

```
submit50 cs50/problems/2019/x/sentimental/crack
```

You can then go to <https://cs50.me/cs50x> (<https://cs50.me/cs50x>) to view your current scores!