

← Python Day00

AboutTaskReviews

Private Git project

ssh://git@repos-ssh.21-school.ru:2289/students/Python_day00.ID_283902



Go to GitLab

Submit the project

Finish project

Private Git project

Day 00 - Python Piscine

Task

Day 00 - Python Piscine

Gumshoe's Gambit

Help the world's famous detective to defend London

Contents

- Chapter I
 - General rules
- Chapter II
 - Rules of the day
- Chapter III
 - Intro
- Chapter IV
 - Exercise 00: Blockchain
- Chapter V
 - Exercise 01: Decypher
- Chapter VI
 - Exercise 02: Track and Capture

- Your scripts should not quit unexpectedly (giving an error on a valid input). If this happens, your project will be considered non functional and will receive a 0 during the evaluation.
- We encourage you to create test programs for your project even though this work won't have to be submitted and won't be graded. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded.

- You should only turn in `*.py` files
- This day's programs in EX00 and EX02 should receive text to standard input and NOT read it from files. EX01 should just receive an input as a command line argument.

Inspector Lestrade was furious. Not only they haven't managed to arrest any terrorists, they also couldn't retrieve anything useful from their base of operations.

Sherlock, on the opposite, was calm and relaxed. The only problem was inspector feeling even more annoyed because of that.

– So...absolutely nothing? - Sherlock clarified with a smirk, - No maps or evil plans, not a single lost passport or credit card?

– Several people almost died, Sherlock! Our constables managed to escape basically on the last second before the whole place blew up. The only thing we managed to do was pulling some files from their server, but they won't help.

– Why do you think that?

– Because they've run some scrambler on that data before escaping, and now it's just some jibberish. Here, if you insist, feel free to take a look!

Sherlock carefully took the laptop from the inspector's hands making multiple notes on the way about Lestrade's methods of handling state property. Specifically, it seemed that using the laptop directly after eating street food was not a really good idea, not mentioning food habits themselves.

On the screen there was one of a several files that consisted of lots of lines which looked similar to this:

```
00000254b208c0f43409d8dc00439896
000000434dd5469464f5cafd8ffe3609
00000f31eaabadebf948f28d1
e7a1ee0b7de74786a2c0180bcd632da
0000085a34260d1c84e89865c210ceb4
073f7873a75c457cbb3307d729501cb5
b7c93ff4cc1c4e0486a8fc66605
fe564b26f25e47c393d07e494021479e
a5dff06057d14566b45caef813511738
0000071f49cffeaa4184be3d507086v
```

– You are correct, this is totally a nonsense. I would even say, nonce-n-sense.

– What the bloody hell do you mean?

– You see, Lestrade, your guess is as good as mine, but I would say these look like distributed blockchain hashes.

– The what now? And how do you know that?

– This is the way of storing the data securely in a distributed fashion. Also, the file is literally called `data_hashes.txt`. Apparently, our criminals are using blockchain to store documents.

Sherlock put his fingers together in front of his face. That was a common gesture for when he was going deeper into his mind palace to extract some knowledge and draw some conclusions. After several moments of silence he said just one thing:

– I think we should filter it.

So, right now we don't know much about this blockchain's implementation, all we have is a file with some lines line shown above. But you may have already noticed a pattern - some lines are starting with several zeroes. So, let's write a Python script which will be able to receive a text file to its standard input, and then print out only those lines that start with exactly 5 zeroes.

Keep in mind that the data has been corrupted, so you have to be very careful. That means, only lines that fit into certain criteria are considered valid:

- Correct lines are 32 characters long
- They start with exactly 5 zeroes, so e.g. line starting with 6 zeroes is NOT considered correct

So, for the example above your script should print:

```
00000254b208c0f43409d8dc00439896
0000085a34260d1c84e89865c210ceb4
0000071f49cffeaa4184be3d507086v
```

Your code should accept the number of lines as an argument, like this:

```
~$ cat data_hashes_10lines.txt | python blocks.py 10
```

This way the program when stop when it processed 10 lines.

This time inspector was a lot more calm, even a bit cheerful. In the crime lab they've figured out that the blockchain implementation used by criminals was really poorly designed, so even in these circumstances they've managed to pull some documents out of the distributed network.

But then one incoming SMS changed everything.

– Sherlock! They want to blow up several buildings in London! Today!

Inspector jumped up from the armchair and started nervously walking back and forth.

– Give me more.

– One of the documents had a list of locations and directions. At first we thought that it is a list of their meeting points, but then one operative found a hidden smartphone on one of those locations. It had instructions for the engineers for setting up the device, but no exact locations. Only a whole bunch of very weird emails.

– Can I see?

– Sure.

This WAS weird. Emails consisted of some strange texts, like "The only way everyone reaches Brenda rapidly is delivering groceries explicitly" or "Britain is Great because everyone necessitates".

After about two minutes of silently looking at those emails, Sherlock grabbed his laptop again and started furiously typing. In next minute, he turned it around and showed the list to Lestrade.

– I think these are the locations where bombs will be planted. The algorithm is actually pretty simple.

Could you find out how Sherlock figured it out and write a Python script that can be used to decrypt any messages like that? (If you want to solve this mystery yourself - don't peek at the checklist right away). It should be launchable like this:

```
~$ python decipher.py "Have you delivered eggplant pizza at restored keep?"
```

and print out the answer as a single word without spaces.

– But almost all these are major tourist attractions! Of course we do have cameras all over them, but there are dozens, maybe hundreds of people there! How do we find those we're looking for?

– I think I know the person who is behind all this.

– Ehh... You do? How?

– Doesn't matter for now. It's just a guess. But I also happen to know, that all goons who work for him and perform dirty work also have a pretty distinct tattoo on the side of the neck. Can your cameras capture that?

– Yes, Sherlock, but we don't have the proper recognition software in place...

– I think we still can do it. Just use a simple ASCII filter on the image and we will solve it from there.

– Which filter?

– Just tell your technicians. They will know what to do.

So, as an input your code is given a 2d "image" as text, five characters over three lines, like this:

```
*d&t*
**h**
*!*&*
```

You may notice that there is a pattern of stars here, with a letter M. All your code has to do is to print 'True' if this M-pattern exists in a given input image or 'False' otherwise. Other characters (outside the M pattern) should be different, so these examples should print out 'False':

```
*****
*****
*****
```

```
*s*f*
**f**
*a***
```

If a given pattern is not 3x5, then 'Error' word should be printed instead.

When you do that, Lestrade will upode that code to police servers and all terrorists will be located by the cameras in no time.

And Sherlock will get ready for some another challenge organized by a mysterious man hiding behind a letter M. Sometimes it seems like these puzzles were specifically engineered for him to solve...