

TimeLock

Write a Python program that implements our TimeLock algorithm. This is a group programming assignment (i.e., **each group will submit one program**). Please make sure that only one member of your group submits the program!

Requirements:

- Submit your source code only;
- Read the epoch from `stdin` in the format `YYYY MM DD HH mm SS`;
- Use the system's current time (although you can manually set the time when testing and debugging); and
- Send the calculated 4-character code to `stdout`.

Algorithm:

- Calculate the time elapsed (in seconds) of the current system time since an epoch time (e.g., 1999 12 31 23 59 59). Please note that you must consider a potential difference in the daylight saving window for both the epoch time and the current time!
- Compute `MD5(MD5(time elapsed))`.
- Extract and concatenate the first two letters (`[a-f]`) of the hash from left-to-right followed by the first two single-digit integers (`[0-9]`) of the hash from right-to-left.
- Output this “code.” Note that each code is valid for 60 seconds even though the code actually changes every second! So the very first code is valid from 1999 12 31 23 59 59 through (including) 2000 01 01 00 00 58; the second code is valid from 2000 01 01 00 00 59 through 2000 01 01 00 01 58.

Example: Suppose the current system time is 2013 05 06 07 43 25 and the epoch time is 1999 12 31 23 59 59. The seconds elapsed is then 421,137,806. But this is in a 60 second interval. The beginning of the relevant 60 second interval is at 421,137,780 elapsed seconds (FYI, the end of the relevant 60 second interval is at 421,137,839 elapsed seconds). The resulting compound hash is `3ee1df13bc19a968b89629c749fee39d`; so the TimeLock “code” is `ee93`.

Although not probable (~1 in 3.4 million), it is possible that the resulting hash does not contain at least two letters or two single-digit integers (e.g., the hash contains only digits or only letters). We've added two cases to handle this situation:

1. Zero or one letter in the hash. In this case, simply grab as many letters as are in the hash and fill in with extra single-digit integers. So for the hash `123456`, the code would be `6543` and for the hash `12a345`, the code would be `a543`.

2. Zero or one single-digit integer in the hash. In this case, grab the two letters, grab as many single-digit integers as are in the hash and fill in with extra letters. So for the hash abcdef, the code would be abcd and for the hash abc1de, the code would be ab1c.

Realistically, I don't think that you need to handle these special cases since they will probably never happen. And if they do, you will most likely not be affected for more than 60 seconds (our valid code interval). **Note that your grade will be no different whether or not you implement this feature.**

Please, no GUIs. Make this a command line application without frills that I can execute at the command line. It should output the calculated 4-character code as in the various examples illustrated below:

```
epoch.txt
```

```
2017 01 01 00 00 00
```

```
current system time: 2017 03 23 18 02 06
```

```
jgourd@latech:~$ python timelock.py < epoch.txt  
fa51
```

```
current system time: 2017 04 23 18 02 30
```

```
jgourd@latech:~$ echo "1999 12 31 23 59 59" | python timelock.py  
ca45
```

```
current system time: 2010 06 13 12 55 34
```

```
jgourd@latech:~$ echo "2001 02 03 04 05 06" | python timelock.py  
dd15
```

```
current system time: 2015 05 15 14 00 00
```

```
jgourd@latech:~$ echo "2015 01 01 00 00 00" | python timelock.py  
ba26
```

```
current system time: 2015 01 01 00 00 00
```

```
jgourd@latech:~$ echo "2014 12 31 00 00 00" | python timelock.py  
dc24
```

```
current system time: 2015 01 01 00 00 30
```

```
jgourd@latech:~$ echo "2014 12 31 00 00 00" | python timelock.py  
dc24
```

```
current system time: 2015 01 01 00 01 00
```

```
jgourd@latech:~$ echo "2014 12 31 00 00 00" | python timelock.py  
ec29
```

current system time: 2015 01 01 00 01 30

```
jgourd@latech:~$ echo "2014 12 31 00 00 00" | python timelock.py  
ec29
```

current system time: 2017 04 26 15 14 30

```
jgourd@latech:~$ echo "1974 06 01 08 57 23" | python timelock.py  
de25
```