ROT13 (Rotate 13)

Write a Python program called `rot13.py` that will encrypt/decrypt input text by shifting the text by a given `-s|--shift` argument or will move each character halfway through the alphabet, e.g., "a" becomes "n," "b" becomes "o," etc. The text to rotate should be provided as a single positional argument to your program and can either be a text file, text on the command line, or `-` to indicate STDIN so that you can round-trip data through your program to ensure you are encrypting and decrypting properly.

Discussion

The way I approached the solution is to think of adding time. If it's 8 in the morning and I want to know the time in 6 hours on a 12-hour (not military/24-hour) clock, I need to think in terms of 12 when the clock rolls over from AM to PM. To do that, I need to know the remainder of dividing by 12, which is given by the modulus `%` operator:

```
>>> now = 8
>>> (now + 6) % 12
2
```

And 6 hours from 8AM is, indeed, 2PM.

Similarly if I want to know how many hours (in decimal) are a particular number of minutes, I need to mod by 60:

```
>>> minutes = 90
>>> int(minutes / 60) + (minutes % 60) / 60
1.5
>>> minutes = 204
>>> int(minutes / 60) + (minutes % 60) / 60
3.4
```

If you `import string`, you can see all the lower/uppercase letters

```
>>> import string
>>> string.ascii_lowercase
'abcdefghijklmnopqrstuvwxyz'
>>> string.ascii_uppercase
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

So I think about "rot13" like adding 13 (or some other shift interval) to the position of the letter in the list and modding by the length of the list to wrap it around. If the shift is 13 and we are at "a" and want to know what the letter 13 way is, we can use `pos` to find "a" and add 13 to that:

```
>>> lcase = list(string.ascii_lowercase)
>>> lcase.index('a')
0
>>> lcase[lcase.index('a') + 13]
'n'
```

But if we want to know the value for something after the 13th letter in our list, we are in trouble!

```
>>> lcase[lcase.index('x') + 13]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
```

`%` to the rescue!

```
>>> lcase[(lcase.index('x') + 13) % len(lcase)]
'k'
```

It's not necessary in this algorithm to shift by any particular number. 13 is special because it's halfway through the alphabet, but we could shift by just 2 or 5 characters. If we want to round-trip our text, it's necessary to shift in the opposite direction on the second half of the trip, so be sure to use the negative value there!

Expected Behavior

```
$ ./rot13.py
usage: rot13.py [-h] [-s int] str
rot13.py: error: the following arguments are required: str
$ ./rot13.py -h
usage: rot13.py [-h] [-s int] str

Argparse Python script

positional arguments:
  str                   Input text, file, or "-" for STDIN

optional arguments:
  -h, --help            show this help message and exit
  -s int, --shift int   Shift arg (default: 0)
$ ./rot13.py AbCd
NoPq
$ ./rot13.py AbCd -s 2
CdEf
$ ./rot13.py fox.txt
```

```
Gur dhvpx oebja sbk whzcf bire gur ynml qbt.
$ ./rot13.py fox.txt | ./rot13.py -
The quick brown fox jumps over the lazy dog.
$ ./rot13.py -s 3 fox.txt | ./rot13.py -s -3 -
The quick brown fox jumps over the lazy dog.
```

Test Suite

A passing test suite looks like this:

```
$ make test
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/pyth
cachedir: .pytest_cache
rootdir: /Users/kyclark/work/python/practical_python_for_data_science/ch05-python-strings-li
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 5 items

test.py::test_usage PASSED                                               [ 20%]
test.py::test_text PASSED                                                [ 40%]
test.py::test_file PASSED                                                [ 60%]
test.py::test_roundtrip_text PASSED                                      [ 80%]
test.py::test_roundtrip_file PASSED                                      [100%]

========================== 5 passed in 0.63 seconds ===========================
```