Python Gradaute - Parsing SwissProt

> "Without requirements or design, programming is the art of adding bugs to an empty text file." - Louis Srygley

Create a Python program called "swisstake.py" that processes a SwissProt-formatted file as a positional argument. It should have a *required* `-k|--keyword` argument of the keyword to match in the "keyword" field of the input record in order to determine which sequences to "take" (hence the name). It should also have an *optional* `-s|--skip` argument to "skip" records with given taxa (which could be many so `nargs='+'`), as well as an *optional* `-o|--output` argument to where to write the output in FASTA format (default "out.fa").

If the given input file is not a file, it should die with '"XXX" is not a file'.

```
$ ./swiss.py
usage: swiss.py [-h] [-s STR [STR ...]] [-k STR [STR ...]] [-o FILE] FILE
swiss.py: error: the following arguments are required: FILE
[cholla@~/work/worked_examples/07-grad-swissprot]$ ./swiss.py -h
usage: swiss.py [-h] [-s STR [STR ...]] [-k STR [STR ...]] [-o FILE] FILE

Filter Swissprot file for keywords, taxa

positional arguments:
  FILE                  Uniprot file

optional arguments:
  -h, --help            show this help message and exit
  -s STR [STR ...], --skip STR [STR ...]
                        Skip taxa (default: )
  -k STR [STR ...], --keyword STR [STR ...]
                        Take on keyword (default: )
  -o FILE, --output FILE
                        Output filename (default: out.fa)
$ ./swisstake.py swiss.txt
usage: swisstake.py [-h] [-s STR [STR ...]] -k STR [-o FILE] FILE
swisstake.py: error: the following arguments are required: -k/--keyword
$ ./swisstake.py -k proteome foo
"foo" is not a file
$ ./swisstake.py swiss.txt -k "complete proteome" -s Metazoa FUNGI viridiplantae
Processing "swiss.txt"
Done, skipped 14 and took 1. See output in "out.fa".
$ ./swisstake.py swiss.txt -k "complete proteome" -s metazoa fungi
Processing "swiss.txt"
Done, skipped 13 and took 2. See output in "out.fa".
```

Test Suite

A passing test suite looks like this:

```
$ make test
python3 -m 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/worked_examples/07-grad-swissprot, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 3 items

test.py::test_usage PASSED                                               [ 33%]
test.py::test_bad_input PASSED                                           [ 66%]
test.py::test_good_input1 PASSED                                         [100%]

========================== 3 passed in 1.75 seconds ===========================
```