

List sorter

Write a Python program called `order.py` that will sort the positional arguments and print them on separate lines preceded by their number in the list, starting at 1:

```
$ ./order.py foo bar baz
1: bar
2: baz
3: foo
```

The program should also accept a `-r` or `--reverse` flag to indicate that the order should be reversed:

```
$ ./order.py foo bar baz -r
1: foo
2: baz
3: bar
```

Note that the number should be printed in a field 3 characters wide:

```
$ ./order.py one two three four five six seven eight nine ten zero
1: eight
2: five
3: four
4: nine
5: one
6: seven
7: six
8: ten
9: three
10: two
11: zero
```

If provided no arguments, the program should express disappointment:

```
$ ./order.py
You have failed me for the last time, Commander.
```

If run with `-h` or `--help`, the program should print a usage:

```
$ ./order.py -h
usage: order.py [-h] [-r] [str [str ...]]

Order all the things

positional arguments:
  str                The things to order (default: None)

optional arguments:
  -h, --help        show this help message and exit
  -r, --reverse     Reverse the sort order (default: False)
```

The program should pass all tests:

```
$ ./order.py
You have failed me for the last time, Commander.
[cholla@~/work/teaching/extra/02_lists]$ make test
pytest -xv test.py
===== test session starts =====
...
collected 8 items

test.py::test_exists PASSED [ 12%]
test.py::test_usage PASSED [ 25%]
test.py::test_nothing PASSED [ 37%]
test.py::test_one_element PASSED [ 50%]
test.py::test_two_elements PASSED [ 62%]
test.py::test_two_elements_reversed PASSED [ 75%]
test.py::test_more PASSED [ 87%]
test.py::test_more_reverse PASSED [100%]

===== 8 passed in 0.41s =====
```