

**Objectives.** Write simple Python programs that involve the following:

1. Command-line input.
2. Standard input.
3. Formatted (standard) output.

**Problem 1.** (*Min Max*) Write a program `min_max.py` that reads in floats (as many as the user enters) from standard input and writes out the minimum and maximum values along with their ranks, ie, their positions (starting at 1) in the input.

```
$ python3 min_max.py
3.0 1.0 5.0 4.0 2.0
<ctrl-d>
min val = 1.000000, min rank = 2
max val = 5.000000, max rank = 3
```

**Problem 2.** (*Missing Number*) Write a program `missing.py` that takes an integer  $N$  as command-line argument, reads in  $N - 1$  distinct integers between 1 and  $N$  from standard input, and writes the missing number.

```
$ python3 missing.py 5
1 3 2 5
<ctrl-d>
4
```

**Problem 3.** (*Geometric and Harmonic Means*) Write a program `means.py` that reads in positive real numbers from standard input and writes their geometric and harmonic means. The *geometric mean* of  $n$  positive numbers  $x_1, x_2, \dots, x_n$  is  $(x_1 \times x_2 \times \dots \times x_n)^{1/n}$  and their *harmonic mean* is  $n/(1/x_1 + 1/x_2 + \dots + 1/x_n)$ . Hint: for the geometric mean, consider taking logarithms to avoid overflow; if  $y = (a \times b \times c)^{1/d}$ , then  $\ln y = (1/d)(\ln a + \ln b + \ln c) = x$  (say), and  $y = e^x$ .

```
$ python3 means.py
1.0 2.0 3.0 4.0 5.0
<ctrl-d>
geometric mean = 2.605171, harmonic mean = 2.189781
```

**Problem 4.** (*Word Count*) Write a program `word_count.py` that reads text from standard input and writes to standard output the number of words in the text. For the purpose of this problem, a word is a sequence of non-whitespace characters that is surrounded by whitespace.

```
$ python3 word_count.py
it was the best of times it was the worst of times
<ctrl-d>
number of words = 12
```

**Problem 5.** (*Closest Point*) Write a program that takes three floats  $x$ ,  $y$ , and  $z$  from the command line, reads from standard input a sequence of coordinates  $(x_i, y_i, z_i)$ , and writes the coordinates of the point closest to  $(x, y, z)$ . Recall that the square of the distance between  $(x, y, z)$  and  $(x_i, y_i, z_i)$  is  $(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2$ . For efficiency, do not use either `math.sqrt()` or the `**` operator.

```
$ python3 closest.py 1.0 5.0 2.0
1.0 3.0 9.0 5.0 3.0 2.5 9.0 6.0 2.0 2.0 6.0 3.0 5.0 6.0 5.0
<ctrl-d>
closest point = (2.000000, 6.000000, 3.000000)
```

## Files to Submit

1. `min_max.py`

2. `missing.py`
3. `means.py`
4. `word_count.py`
5. `closest.py`

**Before you submit:**

- Make sure your programs meet the input and output specifications by running the following command on the terminal:

```
$ python3 run_tests.py -v [<problems>]
```

where the optional argument `<problems>` lists the problems (`Problem1`, `Problem2`, etc.) you want to test, separated by spaces; all the problems are tested if no argument is given.

- Make sure your programs meet the style requirements by running the following command on the terminal:

```
$ pycodestyle <program>
```

where `<program>` is the `.py` file whose style you want to check.