Objectives. Write simple Python programs that involve the following:

1. Conditional (if) statements.
2. Loop (while and for) statements.

Problem 1. (Three Equal Numbers) Write a program equality.py that takes three integers as command-line arguments and writes “equal” if all three are equal, and “not equal” otherwise.

```
$ python3 equality.py 5 5 5
equal
$ python3 equality.py 5 1 5
not equal
```

Problem 2. (Five in a Row) Write a program five_per_row.py that writes the integers 101 to 200 with five numbers per line. Hint: use the % operator.

```
$ python3 five_per_row.py
101 102 103 104 105
106 107 108 109 110
...
196 197 198 199 200
```

Problem 3. (Newton’s Method) Using Program 1.3.6 (sqrt.py) from the IPP text as a model, develop a program root.py that takes a float $c$ and an integer $k$ as command-line arguments and writes $\sqrt[k]{c}$ (kth root of $c$) to 5 decimal places of accuracy, i.e., use $\epsilon = 0.00001$. Hint: at each iteration, replace the estimate $t$ by $t - f(t)/f'(t)$, where $f(t) = t^k - c$ and $f'(t) = k t^{k-1}$; and use the condition $|1 - c/t^k| > \epsilon$ to continue the loop.

```
$ python3 root.py 3 2
1.73205081001
$ python3 root.py 64 3
4.00000050864
```

Problem 4. (Euclid’s Algorithm) Write a program gcd.py that takes two integers $x$ and $y$ as command-line arguments and writes their greatest common divisor (gcd) computed using Euclid’s Algorithm: if $y$ divides $x$, the gcd of $x$ and $y$ is $y$; otherwise, the gcd of $x$ and $y$ is the same as the gcd of $y$ and $x$ mod $y$.

```
$ python3 gcd.py 54 24
6
$ python3 gcd.py 22 45
1
```

Problem 5. (Counting Primes) Write a program prime_counter.py that takes an integer $N$ as command-line argument and writes the number of primes less than or equal to $N$. Note that if you are not careful about the upper bound of the loop that tests for the primality of a number, your program may not finish in a reasonable amount of time. Hint: a number $i$ is prime if it is not divisible by any number $j \in [2, \sqrt{i}]$.

```
$ python3 prime_counter.py 1000
168
```

Files to Submit

1. equality.py
2. five_per_row.py
3. root.py
4. gcd.py
5. prime_counter.py

**Before you submit:**

1. 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.

2. 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.