Exercises marked with a ⋆ involve writing programs. These exercises do not have accompanying solutions. You are encouraged to work them out on your own. If you need help with any of them, please reach out to the course staff.

**Exercise 1.** What is the value and type of each of the following expressions?

a. `"1" + "-" + "1"`

b. `"This bird wouldn't voom if you put " + str(4) + " million volts through it!"`

c. `"42" * 3`

d. `int("42") * 3`

e. `1 - 1 - 1 - 1`

f. `4 + 6 / 2`

g. `(4 + 6) / 2`

h. `3 / 2 + 2 * 5`

i. `3 // 2 + 2 * 5`

j. `(3 + 6) * 6 - 9 / 3 + 20`

k. `2 ** 1 ** 2`

l. `(2 ** 1) ** 2`

m. `3 + 2 % 5`

n. `((3 + 2) % 5`

o. `3 < 11 and 11 < 8 or 8 > 3`

p. `-5 ** 2 + 34 > 6 + 8 / 2 * 3`

q. `8 <= 2 or 8e2 <= 2e8`

**Exercise 2.** What is the value of `x` after the execution of the following statements:

a.

```
x = 1
x += x
x += x
x += x
```

b.

```
x = True
x = not x
x = not x
x = not x
```

c.

```
x = 2
x *= x
x *= x
```

```
x *= x
```

**Exercise 3.** Write the following numbers using scientific notation in Python:

a. 37,000,000

b. 0.000059

**Exercise 4.** Consider the following program `mystery.py`:

```
import stdio
import sys

x = int(sys.argv[1])
y = int(sys.argv[2])
z = x + y

stdio.writeln(z ** 2)
```

a. What does the program write when run with command-line inputs `-1` and `6`?

b. What does the program write when run with command-line inputs `3` and `4`?

c. What does the program write in general?

**Exercise 5.** Consider the following program `mystery.py`:

```
import stdio
import sys

a = int(sys.argv[1])
b = int(sys.argv[2])
c = int(sys.argv[3])

stdio.writeln(a ** 2 == b ** 2 + c ** 2 or b ** 2 == a ** 2 + c ** 2 \
              or c ** 2 == a ** 2 + b ** 2)
```

a. What does the program write when run with command-line inputs `1`, `2`, and `3`?

b. What does the program write when run with command-line inputs `3`, `4`, and `5`?

c. What does the program write in general?

**Exercise 6** ($\star$)**.** Write a program called `f2c.py` that receives `f` (float) as command-line input representing the temperature in Fahrenheit, and writes as standard output the Celsius equivalent `c` of the temperature, calculated as $\texttt{c} = \frac{5}{9}(\texttt{f} - 32)$.

```
$ python3 f2c.py 32
0.0
$ python3 f2c.py 212
100.0
```

**Exercise 7** ($\star$)**.** Write a program called `reciprocal_division.py` that receives two integers `x` and `y` as command-line inputs, and writes `True` as standard output if either number divides the other, and `False` otherwise.

```
$ python3 reciprocal_division.py 3 4
False
$ python3 reciprocal_division.py 6 3
True
```

**Exercise 8** ($\star$)**.** Write a program called `distance.py` that receives two floats `x` and `y` as command-line inputs, and writes as standard output the Euclidean distance of the point $(\texttt{x}, \texttt{y})$ to the origin $(0, 0)$, computed as $\sqrt{\texttt{x}^2 + \texttt{y}^2}$.

```
$ python3 distance.py 3 4
5.0
$ python3 distance.py 6 8
10.0
```

**Exercise 9** ($\star$)**.** Write a program called `sum_of_sines.py` that receives an angle `t` (float) in degrees as command-line input, and writes as standard output the value of $\sin(2\texttt{t}) + \sin(3\texttt{t})$.

```
$ python3 sum_of_sines.py 30
1.8660254037844386
$ python3 sum_of_sines.py 60
0.8660254037844388
```

**Exercise 10** ($\star$)**.** Write a program called `spring.py` that receives `m` (int) and `d` (int) as command-line inputs, and writes `True` as standard ouput if day `d` of month `m` is between 3/20 (inclusive) and 6/20 (inclusive), and `False` otherwise.

```
$ python3 spring.py 3 19
False
$ python3 spring.py 4 15
True
```

**Exercise 11** ($\star$)**.** Write a program called `displacement.py` that receives `x0` (float), `v0` (float), and `t` (float) as command-line inputs, and writes as standard output the value of $\texttt{x0} + \texttt{v0t} - \texttt{gt}^2/2$, where `g` is the constant 9.80665. This value is the displacement in meters after `t` seconds when an object is thrown straight up from initial position `x0` at velocity `v0` meters per second.

```
$ python3 displacement.py 1 50 5
128.416875
$ python3 displacement.py 0 10 1
5.096675
```

**Exercise 12** ($\star$)**.** Write a program called `compound_interest.py` that receives `p` (float), `r` (float), and `t` (float) as command-line inputs, and writes as standard output the amount of money you would have after `t` years if you invested `p` dollars at an annual interest rate `r` compounded continuously. The desired value is computed as $\texttt{pe}^{\texttt{rt}}$.

```
$ python3 compound_interest.py 1000 0.04 1
1040.8107741923882
$ python3 compound_interest.py 15000 0.0375 2
16168.262263269473
```

**Exercise 13** ($\star$)**.** Write a program called `die.py` that receives `n` (int) as command-line input, simulates the roll of an `n`-sided die, and writes the number rolled as standard output.

```
$ python3 die.py 6
3
$ python3 die.py 20
11
```

**Exercise 14** ($\star$)**.** Write a program called `random_gaussian.py` that writes as standard output a random number `r` drawn from the Gaussian distribution. One way to do so is to use the Box-Muller formula $\mathtt{r} = \sin(2\pi\mathtt{v})(-2\ln\mathtt{u})^{1/2}$, where `u` and `v` are real numbers between 0 and 1 generated using the `stdrandom.uniformFloat()` function.

```
$ python3 random_gaussian.py
1.2052313442777243
$ python3 random_gaussian.py
2.301423472572782
```

**Exercise 15** ($\star$)**.** Write a program called `order_check.py` that receives three floats `x`, `y`, and `z` as command-line inputs, and writes `True` as standard output if the values are in strictly ascending ($\mathtt{x} < \mathtt{y} < \mathtt{z}$) or descending ($\mathtt{x} > \mathtt{y} > \mathtt{z}$) order, and `False` otherwise.

```
$ python3 order_check.py 3 2 1
True
$ python3 order_check.py 1 3 2
False
```

## Solutions

**Solution 1.**

a. `"1-1" (str)`

b. `"This bird wouldn't voom if you put 4 million volts through it!" (str)`

c. `"424242" (str)`

d. `126 (int)`

e. `-2 (int)`

f. `7 (int)`

g. `5 (int)`

h. `11.5 (float)`

i. `11 (int)`

j. `89.0 (float)`

k. `2 (int)`

l. `4 (int)`

m. `5 (int)`

n. `0 (int)`

o. `True (bool)`

p. `False (bool)`

q. `True (bool)`

**Solution 2.**

a. `8`

b. `False`

c. `16`

**Solution 3.**

a. `3.7e7`

b. `5.9e-5`

**Solution 4.**

a. `25`

b. `49`

c. $(x + y)^2$

**Solution 5.**

a. `False`

b. `True`

c. `True` if the square of any of `a`, `b`, or `c` is equal to the sum of squares of the other two, and `False` otherwise.

**Solution 6.** Discuss with course staff.

**Solution 7.** Discuss with course staff.

**Solution 8.** Discuss with course staff.

**Solution 9.** Discuss with course staff.

**Solution 10.** Discuss with course staff.

**Solution 11.** Discuss with course staff.

**Solution 12.** Discuss with course staff.

**Solution 13.** Discuss with course staff.

**Solution 14.** Discuss with course staff.

**Solution 15.** Discuss with course staff.