Exercise 1. Consider the following program:

× mystery.py

```
import stdio
import sys

stdio.write("Hi ")
stdio.write(sys.argv[4])
stdio.write(", ")
stdio.write(sys.argv[2])
stdio.write(", and ")
stdio.write(sys.argv[1])
stdio.writeln("!")
```

What does the program write to standard output when it is run as follows?

×

```
$ python3 mystery.py Alice "Bob Carol" Dan Eve
```

Exercise 2. Write a program called `phone.py` that accepts an area code (str), exchange code (str), and subscriber number (str) as command-line arguments, and writes to standard output the corresponding phone number in `+1 (XXX) XXX-XXXX` format.

×

```
$ python3 phone.py 728 560 3845
+1 (728) 560-3845
$ _
```

Exercise 3. What is the value and type of each of the following expressions?

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

b. `"This parrot would not voom if you put " + str(4) + " million volts through it!"`

c. `"42" * 3`

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

e. `float("3.14") * 3`

f. `1 - 1 - 1 - 1`

g. `3 / 2 + 2 * 5`

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

i. `3.14 + int(math.pi) ** 2 % 5`

j. `(3.14 + int(math.pi) ** 2) % 5`

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

l. `5 + int(stdrandom.uniformFloat(0, 1) * 5)`

Exercise 4. 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 arguments 1, 2, and 3?

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

c. What does the program write in general?

Exercise 5. Implement a program called `far2cen.py` that accepts f (float) as command-line argument representing the temperature in Fahrenheit, and writes to standard output the Celsius equivalent c of the temperature, calculated as $c = \frac{5}{9}(f - 32)$. How would you run the program on the terminal to convert 42 °F to °C?

Exercise 6. Implement a program called `die.py` that accepts n (int) as command-line argument, simulates the roll of an n-sided die, and writes the number rolled to standard output.

Exercise 7. Consider the following code fragment:

```
if m >= 1 and m <= 5:
    stdio.write("Spring ")
elif m >= 6 and m <= 8:
    stdio.write("Summer ")
else:
    stdio.write("Fall ")
stdio.writeln(y)
```

What does the program write when `m` and `y` take on the following values?

a. `m = 10` and `y = 2020`

b. `m = 5` and `y = 2021`

c. `m = 6` and `y = 2022`

Exercise 8. What does the following code fragment write?

```
i = 9
while i >= 0:
    stdio.writeln(str(i) + " " + str(2 ** i))
    i -= 2
```

Exercise 9. What are the arithmetic progressions returned by the following calles to `range()`?

a. `range(-5)`

b. `range(5)`

c. `range(3, 10)`

d. `range(3, 10, 2)`

e. `range(5, -5, -1)`

Exercise 10. What does the following code fragment write?

```
for i in range(3, 40, 4):
    if i % 5 == 0:
        stdio.writeln(i)
```

Exercise 11. What does the following code fragment write?

```
i = 1
for c in "hello":
    stdio.writeln(c * i)
    i += 1
```

Exercise 12. What does the following code fragment write?

```
for i in range(5):
    for j in range(6):
        if j == 5:
            stdio.writeln(i + j)
        else:
            stdio.write(str(i + j) + "-")
```

Exercise 13. Implement a program called `genaralizedharmonic.py` that accepts n (int) and r (int) as command-line arguments and writes the value of the generalized harmonic number $H(n, r)$ to standard output, computed using the formula

$$H(n, r) = \frac{1}{1^r} + \frac{1}{2^r} + \frac{1}{3^r} + \cdots + \frac{1}{n^r}.$$

Exercise 14. Implement a program called `matrix.py` that accepts n (int) and k (int) as command-line arguments and writes an $n \times n$ matrix in which the elements below the main diagonal are all zeros and the rest of the elements have the value k. The elements of the matrix must be separated by a single space and each row must end with a newline character at the end.

```
$ python matrix.py 5 2
2 2 2 2 2
0 2 2 2 2
0 0 2 2 2
0 0 0 2 2
0 0 0 0 2
$ _
```

Exercise 15. Consider the program `gambler.py`.

a. How many variables does the program define?

b. Write down the names of the variables and the scope of each variable.

Exercise 16. Consider the following code fragment:

```
a = [0]
for i in range(1, 6):
    a += [a[i - 1] + i]
```

a. What is the value of `a[5]`?

b. What is the value of `sum(a)`?

Exercise 17. What does the following code fragment write?

```
a = ["it", "was", "the", "best", "of", "times", "it", "was", "the", "worst",
     "of", "times"]
x = 0
y = 0
for v in a:
    x += 1
    y += len(v)
stdio.writeln(str(x) + " " + str(y))
```

Exercise 18. What does the following code fragment write?

```
a = [1, 2, 3, 4, 5]
b = a
b[2] = 0
stdio.writeln(sum(a))
```

Exercise 19. Suppose `a = ["mercury", "venus", "earth", "mars", "jupiter", "saturn", "uranus", "neptune"]`. What are the values of the following expressions?

a. `len(a)`

b. `a[2]`

c. `a[3:]`

d. `a[:3]`

e. `a[-2]`

f. `a[-2:]`

g. `a[:-2]`

h. `a[:]`

Exercise 20. What does the following code fragment write?

```
a = [[1, 2, 3], [2, 3, 4], [3, 4, 5]]
x = 0
for i in range(len(a)):
    for j in range(len(a[0])):
        x += a[i][j]
stdio.writeln(x)
```

Exercise 21. What does the following code fragment write?

```
a = stdarray.create1D(4, None)
for i in range(len(a)):
    a[i] = stdarray.create1D(i + 1, 2)
stdio.writeln(sum(a[3]))
```

Exercise 22. Consider the following program `mystery.py`:

× mystery.py

```
import stdarray
import stdio
import sys

n = int(sys.argv[1])
a = stdarray.create2D(n, n, "-")
for i in range(n):
    for j in range(n):
        if i == j or i + j == n - 1:
            a[i][j] = "*"
for i in range(n):
    for j in range(n):
        if j == n - 1:
            stdio.writeln(a[i][j])
        else:
            stdio.write(str(a[i][j]) + " ")
```

a. What does the program write in general?

b. What does the program write when run with the command-line argument $n = 5$?

Exercise 23. Write a program called `die_rolls.py` that accepts n (int) and *trials* (int) as command-line arguments, rolls a fair n-sided die *trials* times, and reports the number of times each of the n values was rolled. For example

×

```
$ python3 die_rolls.py 6 100
1 -> 19 times
2 -> 16 times
3 -> 12 times
4 -> 19 times
5 -> 15 times
6 -> 19 times
```

Exercise 24. What do the following code fragments write?

a.

```
x = (["a", "b", "c"], [1, 2, 3, 4, 5])
stdio.writeln(len(x) + len(x[0]) + len(x[1]))
```

b.

```
x = set("panama")
y = set("canal")
stdio.writeln(x | y)
stdio.writeln(x & y)
stdio.writeln(x - y)
stdio.writeln(y - x)
stdio.writeln(x ^ y)
```

c.

```
x = {"a": 1, "b": 2, "c": 3}
y = "a" * x["a"] + "b" * x["b"] + "c" * x["c"]
stdio.writeln(y)
```

Exercise 25. What do the following code fragments write?

a.

```
for x, y in enumerate(range(1, 10, 2)):
    stdio.writeln(str(x) + ":" + str(y * y))
```

b.

```
w = 0
for x, y, z in zip([1, 2, 3], [4, 5, 6], [7, 8, 9]):
    w += x * y * z
stdio.writeln(w)
```

c.

```
x = ["it", "was", "the", "best", "of", "times", "it", "was", "the", "worst",
     "of", "times"]
for v in reversed(sorted(x)):
    stdio.writeln(v)
```

Exercise 26. What does the following code fragment write to standard output?

```
r = 5
c = 2 * math.pi * r
a = math.pi * r ** 2
stdio.writef("radius = %.2f, circumference = %.2f, area = %.2f\n", r, c, a)
```

Exercise 27. Write a program called `randomints.py` that accepts n (int), a (int), and b (int) as command-line arguments, and writes to standard output n random integers from the interval $[a, b]$ in sorted order. For example

```
$ python3 randomints.py 5 100 1000
238
379
597
748
978
```

Exercise 28. Write a program called `stats.py` that reads a sequence of floats from standard input, and writes to standard output their mean, variance, and standard deviation, each up to 3 decimal places. For example

```
$ python3 stats.py
1 2 3 4 5
<ctrl-d>
mean = 3.000, var = 2.000, std = 1.414
```

The mean μ, variance Var, and standard deviation σ of the numbers $x_1, x_2, \ldots, x_n$ are computed as

$$\mu = \frac{x_1 + x_2 + \cdots + x_n}{n}, Var = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + \ldots (x_n - \mu)^2}{n}, \text{ and } \sigma = \sqrt{Var}.$$

Exercise 29. Consider the programs `randomints.py` and `stats.py` from the previous two problems.

a. What is the command for generating 100 random integers from the interval $[500, 1000]$?

b. What is the command for generating 100 random integers from the interval $[500, 1000]$ and saving the output in a file called `ints.txt`?

c. What is the command to compute stats for the numbers in `ints.txt`?

d. What is the command to perform the last two tasks in one shot?

Solutions

Solution 1.

```
Hi Eve, Bob Carol, and Alice!
```

Solution 2.

× phone.py

```
import stdio
import sys

stdio.write("+1 (")
stdio.write(sys.argv[1])
stdio.write(") ")
stdio.write(sys.argv[2])
stdio.write("-")
stdio.writeln(sys.argv[3])
```

Solution 3.

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

b. `"This parrot would not voom if you put 4 million volts through it!"` (`str`)

c. `"424242"` (`str`)

d. `126` (`int`)

e. `9.42` (`float`)

f. `-2` (`int`)

g. `11.5` (`float`)

h. `11` (`int`)

i. `7.14` (`float`)

j. `2.14` (`float`)

k. `True` (`bool`)

l. A random number from the interval $[5, 10)$ (`int`)

Solution 4.

a. `False`

b. `True`

c. Accepts three command-line arguments a, b, and c as integers and writes `True` if the square of any one of them is equal to the sum of squares of the other two, and `False` otherwise.

Solution 5.

× far2cen.py

```
import stdio
import sys

f = float(sys.argv[1])
c = (f - 32) * 5 / 9
stdio.writeln(f)
```

×

```
$ python3 far2cen.py 42
```

Solution 6.

× die.py

```
import stdio
import stdrandom
import sys

n = int(sys.argv[1])
result = stdrandom.uniformInt(1, n + 1)
stdio.writeln(result)
```

Solution 7.

a. `Fall 2020`

b. `Spring 2021`

c. `Summer 2022`

Solution 8.

```
9 512
7 128
5 32
3 8
1 2
$ _
```

Solution 9.

a. `[]`

b. `[0, 1, 2, 3, 4]`

c. `[3, 4, 5, 6, 7, 8, 9]`

d. `[3, 5, 7, 9]`

e. `[5, 4, 3, 2, 1, 0, -1, -2, -3, -4]`

Solution 10.

```
15
35
```

Solution 11.

```
h
ee
lll
llll
ooooo
```

Solution 12.

```
0-1-2-3-4-5
1-2-3-4-5-6
2-3-4-5-6-7
3-4-5-6-7-8
4-5-6-7-8-9
```

Solution 13.

× generalizedharmonic.py

```
import stdio
import sys

n = int(sys.argv[1])
r = int(sys.argv[2])
total = 0
for i in range(1, n + 1):
    total += 1 / i ** r
stdio.writeln(total)
```

Solution 14.

× matrix.py

```
import stdio
import sys

n = int(sys.argv[1])
k = int(sys.argv[2])
for i in range(n):
    for j in range(n):
        e = 0 if i > j else k
        if j == n - 1:
            stdio.writeln(e)
        else:
            stdio.write(str(e) + " ")
```

Solution 15.

a. There are seven variables defined in `gambler.py`.

b. Here are their names and scopes:

Variable	Scope
`stake`	lines 9 — 25
`goal`	lines 10 — 25
`trials`	lines 11 — 25
`bets`	lines 12 — 25
`wins`	lines 13 — 25
`t`	lines 14 — 23
`cash`	lines 15 — 23

Solution 16.

a. `15`

b. `35`

Solution 17.

```
12 39
```

Solution 18.

```
12
```

Solution 19.

a. `8`

b. `"earth"`

c. `["mars", "jupiter", "saturn", "uranus", "neptune"]`

d. `["mercury", "venus", "earth"]`

e. `"uranus"`

f. `["uranus", "neptune"]`

g. `["mercury", "venus", "earth", "mars", "jupiter", "saturn"]`

h. `["mercury", "venus", "earth", "mars", "jupiter", "saturn", "uranus", "neptune"]`

Solution 20.

```
27
```

Solution 21.

```
8
```

Solution 22.

a. The program writes an $n \times n$ matrix in which the diagonal elements are stars and the off-diagonal elements are dashes.

b.

```
* - - - *
- * - * -
- - * - -
- * - * -
* - - - *
```

Solution 23.

× die_rolls.py

```
import stdarray
import stdio
import stdrandom
import sys

n = int(sys.argv[1])
trials = int(sys.argv[2])
rolls = stdarray.create1D(n + 1, 0)
for i in range(trials):
    v = stdrandom.uniformInt(1, n + 1)
    rolls[v] += 1
for i in range(1, n + 1):
    stdio.writeln(str(i) + " -> " + str(rolls[i]) + " times")
```

Solution 24.

a.

```
10
```

b.

```
abbccc
```

c.

```
{"a", "p", "m", "c", "l", "n"}
{"a", "n"}
{"p", "m"}
{"c", "l"}
{"p", "m", "c", "l"}
```

Solution 25.

a.

```
0:1
1:9
2:25
3:49
4:81
```

b.

```
270
```

c.

```
worst
was
was
times
times
the
the
of
of
it
it
best
```

Solution 26.

```
radius = 5.00, circumference = 31.42, area = 78.54
```

Solution 27.

× randomints.py

```
import stdio
import stdrandom
import sys

n = int(sys.argv[1])
a = int(sys.argv[2])
b = int(sys.argv[3])
ints = []
for i in range(n):
    r = stdrandom.uniformInt(a, b + 1)
    ints += [r]
for v in sorted(ints):
    stdio.writeln(v)
```

Solution 28.

× stats.py

```
import stdio

ints = stdio.readAllInts()
```

```
mean = sum(ints) / len(ints)
var = 0.0
for v in ints:
    var += (v - mean) ** 2
var /= len(ints)
std = var ** 0.5
stdio.writef("mean = %.3f, var = %.3f, std = %.3f\n", mean, var, std)
```

Solution 29.

a.

```
$ python3 randomints.py 100 500 1000
```

b.

```
$ python3 randomints.py 100 500 1000 > ints.txt
```

c.

```
$ python3 stats.py < ints.txt
```

d.

```
$ python3 randomints.py 100 500 1000 | python3 stats.py
```