CHAPTER 2

Elementary computations

2.1 EXERCISE SOLUTIONS

Section 2.2

2.2.1
```
838 * 5280 * 12
```

2.2.2
```
0.05 * 2 ** 24
0.05 * 1.5 ** 24
```

2.2.3
```
838.0 / (32.6 * 2)
```

2.2.4
```
2e5 / 4.54e9
```

2.2.5
```
4.54e9 / 60 / 60 / 24 / 365   # 143.96245560629123 years
```

2.2.6
```
4.54e9 / 2.8e9
```

2.2.7
```
2 ** 40 / (8 * 2 ** 20)
```

2.2.8 (a) 43

(b) 9

(c) 22

(d) 22.5

(e) 1

(f) 22.5

Section 2.3

2.3.1
```
dnaCell = 3.4e-10 * 6e9
dnaBody = dnaCell * 50e12
distSun = 1.49598e8 * 1e3
roundTrips = dnaBody / (distSun * 2)
```

2.3.2
```
radius = 10
area = 3.141592 * radius ** 2
```

2.3.3 `area` remains the same because it was not assigned a new value.

2.3.4 The value of `x` is overwritten by the value of `y` in the first statement. So the second statement does nothing useful. A correct implementation is:

```
temp = x
x = y
y = temp
```

2.3.5 x is 6, y is 24.0

2.3.6 x is 3

2.3.7 x is 12.0 and y is 72.0

2.3.8
```
ones = x % 10
tens = (x % 100) // 10
hundreds = (x % 1000) // 100
```

Section 2.4

2.4.1 The result has the same type as the argument.

2.4.2
```
>>> radius = 10
>>> area = math.pi * radius ** 2
```

2.4.3
```
>>> math.sqrt(18 * 31)
```

2.4.4
```
>>> P = 10000
>>> r = 0.01
>>> n = 12
>>> t = 10
>>> A1 = P * (1 + r / n) ** (n * t)
>>> A2 = P * math.exp(r * t)
>>> A2 - A1
0.460222633122612
```

2.4.5
```
>>> x = 3.1415926
>>> x = int(x * 100) / 100
```

2.4.6
```
>>> x = x - int(x)
```

2.4.7
```
a = 3
b = 4
c = -5

x1 = (-b + math.sqrt(b ** 2 - 4 * a * c)) / (2 * a)
x2 = (-b - math.sqrt(b ** 2 - 4 * a * c)) / (2 * a)
```

2.4.8
```
distance = math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
```

2.4.9
```
volume = a * b * c * math.sqrt(1 + 2 * math.cos(x) * math.cos(y) * math.cos(z)
                               - math.cos(x)**2 - math.cos(y)**2 - math.cos(z)**2)
```

2.4.10
```
volume = a * b * c * math.sqrt(1 + 2 * math.cos(math.radians(x))
                                   * math.cos(math.radians(y))
                                   * math.cos(math.radians(z))
                                 - math.cos(math.radians(x))**2
                                 - math.cos(math.radians(y))**2
                                 - math.cos(math.radians(z))**2)
```

2.4.11
```
P = float(input('Principal: '))
r = float(input('Interest rate: '))
t = int(input('Number of years: '))
n = int(input('Number of times compounded per year: '))
A1 = P * (1 + r / n) ** (n * t)
A2 = P * math.exp(r * t)
print('You would have \$' + str(A2 - A1) + ' more.')
```

2.4.12
```
a = float(input('Value of a: '))
b = float(input('Value of b: '))
c = float(input('Value of c: '))
```

```
x1 = (-b + math.sqrt(b ** 2 - 4 * a * c)) / (2 * a)
x2 = (-b - math.sqrt(b ** 2 - 4 * a * c)) / (2 * a)
print('The two solutions are', x1, 'and', x2)
```

Section 2.5

2.5.1

```
  0 0 1 0 0 1
+ 0 0 1 1 0 1
-------------
  0 1 0 1 1 0
```

2.5.2

```
  0 0 0 1 0 1 0
+ 0 1 0 1 1 0 1
---------------
  0 1 1 0 1 1 1
```

2.5.3

```
  1 0 0 1
+ 1 1 0 1
---------
  0 1 1 0
```

This answer is incorrect since we had to discard the leftmost `1` from the answer.

2.5.4

```
  0 0 1 0 1 0
+ 1 0 1 1 0 1
-------------
  1 1 0 1 1 1
```

This answer is correct since we did not have to discard a leftmost `1` from the answer.

2.5.5 You can tell that the answer is incorrect if it is smaller than the largest operand.

2.5.6

```
  0 1 0 1
+ 1 1 0 1
---------
  0 0 1 0
```

This answer is correct: $5 + -3 = 2$.

2.5.7 The largest positive number is `0111`, which is 7 in decimal. The smallest negative number is `1000`, which is -8 in decimal.

2.5.8

```
  1 0 0 1
+ 1 1 0 1
---------
  0 1 1 0
```

This answer is incorrect: $-7 + -3 \neq 3$. This occurred because the desired answer, -10, does not fit in four bits.

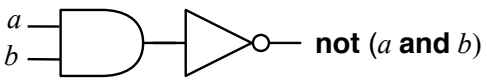
2.5.9

```
  0 0 1 0 1 0
+ 1 0 1 1 0 1
-------------
  1 1 0 1 1 1
```

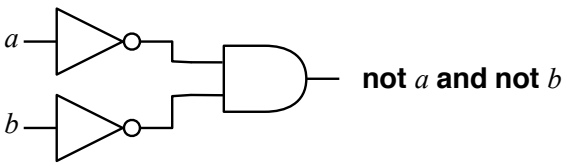
This answer is correct: $10 + -19 = -9$.

2.5.10 You can tell that the answer is incorrect if it has an incorrect sign. This can only happen when we add two positive integers or two negative integers.

2.5.14



2.5.15



2.5.16