David Liben-Nowell

Department of Computer Science

Carleton College

Discrete Mathematics for Computer Science

Solution Manual

This version: October 19, 2017

Section 2.2: Booleans, Numbers, and Arithmetic

2.1 112 and 201

2.2 111 and 201

2.3 18 and 39

2.4 17 and 38

2.5 Yes: there's no fractional part in either x or y, so when they're summed then there's still no fractional part.

2.6 Yes: if we can write $x = a/b$ and $y = c/d$ with $b \neq 0$ and $d \neq 0$, then we can write $x + y$ as $(ad + cb)/bd$ (and $bd \neq 0$ because neither $b = 0$ nor $d = 0$).

2.7 No: π and $1 - \pi$ are both irrational numbers, but $\pi + (1 - \pi) = 1$, which is a rational number.

2.8 6, because $\lfloor 2.5 \rfloor = 2$ and $\lceil 3.75 \rceil = 4$.

2.9 3, because $\lfloor 3.14159 \rfloor = 3$ and $\lceil 0.87853 \rceil = 1$.

2.10 $3^4 = 3 \cdot 3 \cdot 3 \cdot 3 = 81$, because $\lfloor 3.14159 \rfloor = 3$ and $\lceil 3.14159 \rceil = 4$.

2.11 They differ on negative numbers. For any real number x, we have $\lfloor x \rfloor \leq x$—even if it's negative. That's not true for truncation. For example, $\lfloor -3.14159 \rfloor = -4$ but $\texttt{trunc}(-3.14159) = -3.14159 = -3$.

2.12 $\lfloor x + 0.5 \rfloor$

2.13 $0.1 \cdot \lfloor 10x + 0.5 \rfloor$

2.14 $10^{-k} \cdot \lfloor 10^k x + 0.5 \rfloor$

2.15 $10^{-k} \cdot \lfloor 10^k x \rfloor$

2.16 If x is an integer, then $\lfloor x \rfloor + \lceil x \rceil = 2x$; if x isn't an integer, then $\lfloor x \rfloor + \lceil x \rceil = 2 + 3$ for any x in $[2, 3]$. Thus the expression is $x - x = 0$ for $x = 2$ or 3 (yielding values 0 and 0), and $x - 2.5$ for noninteger x. So the largest possible value for this expression is $0.4999999 \cdots$, when $x = 2.9999999 \cdots$.

2.17 By the same logic, the smallest value is $-0.5 + \varepsilon$, which occurs for $x = 2 + \varepsilon$, for arbitrarily small values of ε. For example, the value is -0.4999999 when $x = 2.00000001$.

2.18 $\lfloor x \rfloor$

2.19 $\lceil x \rceil$

2.20 $\lceil x \rceil$

2.21 $\lfloor x \rfloor$

2.22 No: for example, $|\lfloor -3.5 \rfloor| = |-4| = 4$, but $\lfloor |-3.5| \rfloor = \lfloor 3.5 \rfloor = 3$.

2.23 Yes: we have that $x - \lfloor x \rfloor = (x + 1) - \lfloor x + 1 \rfloor$ because both x and $x + 1$ have the same fractional part; rearranging this equality yields the claimed fact.

2.24 No: if $x = y = 0.5$, then $\lfloor x \rfloor + \lfloor y \rfloor = \lfloor 0.5 \rfloor + \lfloor 0.5 \rfloor = 0 + 0 = 0$, but $\lfloor x + y \rfloor = \lfloor 0.5 + 0.5 \rfloor = \lfloor 1 \rfloor = 1$.

2.25 $\lceil x \rceil = \lfloor x \rfloor$ when x is an integer (and otherwise $\lceil x \rceil = \lfloor x \rfloor + 1$). Thus $1 + \lfloor x \rfloor - \lceil x \rceil$ is 1 when x is an integer, and it's 0 when x is not an integer.

2.26 Observe that $\lfloor \frac{n+1}{2} \rfloor = \lceil \frac{n}{2} \rceil$: it's true when n is even, and it's true when n is odd. Thus there are $\lceil \frac{n}{2} \rceil - 1$ elements less than the specified entry, and $n - (\lceil \frac{n}{2} \rceil - 1 + 1) = n - \lceil \frac{n}{2} \rceil = \lfloor \frac{n}{2} \rfloor$ elements larger.

2.27 3^{10} is bigger: $3^{10} > 3^9 = (3^3)^3 = 27^3 > 10^3$.

2.28 $2^{16} = 65{,}536$

2.29 $1/2^{16} = 1/65536 = 0.00001525878$

2.30 65,536

2.31 $-4 \cdot 65{,}536 = -262{,}144$

2.32 4, because $4^4 = 256$.

2.33 $\sqrt[4]{8} \approx 1.6818$, because $1.6818^4 \approx 8.0001$.

2.34 $\sqrt[4]{512} \approx 4.7568$, because $4.7568^4 \approx 511.9877$.

2.35 Undefined (because we don't include imaginary numbers in this book): there's no real number x for which $x^4 = -9$.

2.36 3, because $2^3 = 8$.

2.37 -3, because $2^{-3} = 1/2^3 = 1/8$.

2.38 $1/3$, because $8^{1/3} = 2$ (which is true because $2^3 = 8$).

2.39 $-1/3$, because $(1/8)^{-1/3} = 8^{1/3} = 2$.

2.40 $\log_{10} 17$ is larger than 1 (because $10^1 < 17$), but $\log_{17} 10$ is smaller than 1 (because $17^1 > 10$). So $\log_{10} 17$ is larger.

2.41 By definition, $\log_b 1$ is the real number y such that $b^y = 1$. By (2.1.1), for any real number b we have $b^0 = 1$, so $\log_b 1 = 0$.

2.42 By definition, $\log_b b$ is the real number y such that $b^y = b$. By (2.1.2), for any real number b we have $b^1 = b$, so $\log_b b = 1$.

2.43 Let $q := \log_b x$. By definition of logarithms, we have $b^q = x$. Thus, raising both sides to the yth power, we have $(b^q)^y = x^y$. We can rewrite $(b^q)^y = b^{qy}$ using (2.1.4), so $x^y = b^{qy}$. Again using the definition of logarithms, therefore $\log_b x^y = qy$. By the definition of q, this value is $y \cdot \log_b x$.

2.44 Let $q := \log_b x$ and $r := \log_b y$. By definition, then, $b^q = x$ and $b^r = y$. Using (2.1.3), we have $x \cdot y = b^q \cdot b^r = b^{q+r}$. Because $xy = b^{q+r}$, we have by definition of logs that $\log_b xy = q + r$.

2.45 Let $q := \log_c b$ and $r := \log_b x$, so that $c^q = b$ and $b^r = x$. Then $x = (c^q)^r = c^{qr}$ by the definition of logs and (2.1.4), and therefore $\log_c x = qr = (\log_c b) \cdot (\log_b x)$; rearranging yields the claim.

2.46 We show that $b^{[\log_b x]} = x$ by showing that $\log_b$ of the two sides are equal:

$$\begin{aligned} \log_b \left[b^{[\log_b x]} \right] &= [\log_b x] \cdot \log_b b && \text{by (2.2.5)} \\ &= [\log_b x] \cdot 1 && \text{by (2.2.2)} \\ &= \log_b x. \end{aligned}$$

2.47 We show that $n^{[\log_b a]} = a^{[\log_b n]}$ by showing that $\log_b$ of the two sides are equal:

$$\begin{aligned} \log_b \left[n^{[\log_b a]} \right] &= [\log_b a] \cdot \log_b n && \text{by (2.2.5)} \\ &= [\log_b n] \cdot \log_b a && x \cdot y = y \cdot x \text{ for any } x, y \\ &= \log_b \left[a^{[\log_b n]} \right]. && \text{by (2.2.5), applied ``backward''} \end{aligned}$$

2.48 The property $\log_b \frac{x}{y} = \log_b x - \log_b y$ follows directly from the rule for the log of a product: $\frac{x}{y} = x \cdot y^{-1}$, so

$$\begin{aligned} \log_b \tfrac{x}{y} &= \log_b (x \cdot y^{-1}) \\ &= \log_b x + \log_b y^{-1} && \text{by (2.2.3)} \\ &= \log_b x + (-1) \cdot \log_b y. && \text{by (2.2.5)} \end{aligned}$$

2.49 $\overline{\lceil n \rceil} := 2^{\lceil \log_2 n \rceil}$

2.50 The number of columns needed to write down n in standard decimal notation is

$$\begin{cases} \lceil \log_{10}(n+1) \rceil & \text{if } n > 0 \\ 1 & \text{if } n = 0 \\ \lceil \log_{10}(|n|+1) \rceil & \text{if } n < 0. \end{cases}$$

2.51 0 (because $202 = 101 \cdot 2$)

2.52 1 (because $202 = 1 + (67 \cdot 3)$)

2.53 2 (because $202 = 2 + (20 \cdot 10)$)

2.54 8 (because $-202 = 8 + (-21 \cdot 10)$)
2.55 17 (because $17 = 17 + (0 \cdot 42)$)
2.56 8 (because $42 = 8 + (2 \cdot 17)$)
2.57 0 (because $17 = 0 + (1 \cdot 17)$)
2.58 9 (because $-42 = 9 + (-3 \cdot 17)$)
2.59 0 (because $-42 = 0 + (-1 \cdot 42)$)
2.60 If $k > 0$, then $n \bmod k := n - k \cdot \lfloor \frac{n}{k} \rfloor$. If $k < 0$, then $n \bmod k := -(-n \bmod -k) = -n + k \cdot \lfloor \frac{n}{k} \rfloor$.
2.61 30
2.62 31
2.63 10
2.64 68
2.65 53
2.66 Here's a solution in Python:

```
def isPrime(n):
   for d in range(2,n):
      if n % d == 0:
          return False
   return True

for n in range(2,202):
   if isPrime(n):
      print n
```

This program generates the following list of prime numbers: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199.

2.67 Here's a solution in Python. The four numbers that this program finds are 6, 28, 496, and 8128.

```
def perfect(n):
   factorSum = 0
   for d in range(1,n):
      if n % d == 0:
         factorSum += d
   return factorSum == n

found = 0
n = 1
while found < 4:
   if perfect(n):
      found += 1
      print n
   n += 1
```

2.68 Here's a solution in Python:

```
def threeFactors(n):
   factorCount = 0
   for d in range(1,n+1):
      if n % d == 0:
         factorCount += 1
   return factorCount == 3

for n in range(1001):
   if threeFactors(n):
      print n
```

The numbers produced as output are the following: 4 9 25 49 121 169 289 361 529 841 961. Note that these output values are $2^2, 3^2, 5^2, 7^2, 11^2, 13^2, \ldots, 31^2$—the squares of all sufficiently small prime numbers. For a prime number p, the three factors of p^2 are 1, p, and p^2 (and p^2 has no other factor).

2.69 $6 + 6 + 6 + 6 + 6 + 6 = 36$
2.70 $1 + 4 + 9 + 16 + 25 + 36 = 91$
2.71 $4 + 16 + 64 + 256 + 1024 + 4096 = 5460$
2.72 $1(2) + 2(4) + 3(8) + 4(16) + 5(32) + 6(64) = 642$
2.73 $(1+2) + (2+4) + (3+8) + (4+16) + (5+32) + (6+64) = 147$
2.74 $6 \cdot 6 \cdot 6 \cdot 6 \cdot 6 \cdot 6 = 46{,}656$
2.75 $1 \cdot 4 \cdot 9 \cdot 16 \cdot 25 \cdot 36 = 518{,}400$
2.76 $2^2 \cdot 2^4 \cdot 2^6 \cdot 2^8 \cdot 2^{10} \cdot 2^{12} = 2^{42} = 4{,}398{,}046{,}511{,}104$
2.77 $1(2) \cdot 2(4) \cdot 3(8) \cdot 4(16) \cdot 5(32) \cdot 6(64) = 1{,}509{,}949{,}440$
2.78 $(1+2) \cdot (2+4) \cdot (3+8) \cdot (4+16) \cdot (5+32) \cdot (6+64) = 10{,}256{,}400$
2.79 $21 + 42 + 63 + 84 + 105 + 126 = 441$
2.80 $21 + 40 + 54 + 60 + 55 + 36 = 266$
2.81 $1 + 6 + 18 + 40 + 75 + 126 = 266$
2.82 $8 + 14 + 18 + 20 + 20 + 18 + 14 + 8 = 120$
2.83 $36 + 35 + 33 + 30 + 26 + 21 + 15 + 8 = 204$
2.84 $44 + 49 + 51 + 50 + 46 + 39 + 29 + 16 = 324$
2.85 $(1^1 + 2^1 + 3^1 + 4^1) + (1^2 + 2^2 + 3^2 + 4^2) + (1^3 + 2^3 + 3^3 + 4^3) + (1^4 + 2^4 + 3^4 + 4^4) = (1 + 2 + 3 + 4) + (1 + 4 + 9 + 16) + (1 + 8 + 27 + 64) + (1 + 16 + 81 + 256) = 494$

Section 2.3: Sets: Unordered Collections

2.86 Yes, $6 \in \{0,1,2,3,4,5,6,7,8,9,\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d},\mathtt{e},\mathtt{f}\}$.

2.87 No, $\mathtt{h} \notin \{0,1,2,3,4,5,6,7,8,9,\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d},\mathtt{e},\mathtt{f}\}$.

2.88 No, the only elements of H are single characters, so $\mathtt{a70e} \notin H$.

2.89 $|H| = 16$

2.90 $0 = 0+0 = 0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0$, and $1 = 0+1 = 1+0 = 1 \cdot 1$, and $2 = 1+1$. So 0, 1, and 2 are elements of S, but 3 is not.

2.91 $|S| = 3$: although there are eight different descriptions of elements, only three distinct values result and so there are exactly three elements in the set.

2.92 4 is in H but not in T, because $4 \bmod 2 = 0$ but $4 \bmod 3 \neq 0$. Other elements in H but not T are 2, 3, 5, 8, 9, $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$, $\mathtt{d}$, $\mathtt{e}$, and $\mathtt{f}$.

2.93 13 is an element of T, because $13 \bmod 2 = 1 = 13 \bmod 3$. Other elements in T but not H are 12, 18, and 19.

2.94 The elements of T not in S are: 6, 7, 12, 13, 18, and 19.

2.95 The only element of S that is not in T is 2.

2.96 $|T| = 8$

2.97 $\{0,1,2,3,4\}$

2.98 $\{\}$

2.99 $\{4,5,9\}$

2.100 $\{0,1,3,4,5,7,8,9\}$

2.101 $\{1,3,7,8\}$

2.102 $\{0\}$

2.103 $\{1,2,3,6,7,8\}$

2.104 $\{1,2,3,4,5,7,8,9\}$

2.105 $\{4,5\}$

2.106 $C - {\sim}C$ is just C itself: $\{0,3,6,9\}$.

2.107 $C - {\sim}A$ is $\{3,9\}$, so ${\sim}(C - {\sim}A)$ is $\{0,1,2,4,5,6,7,8\}$.

2.108 Yes, it's possible: if $E = Y$, then we have $E - Y = Y - E = \varnothing$. (But if $E \neq Y$, then there's some element x in one set but not the other—and if $x \in E$ but $x \notin Y$, then $x \in E - Y$ but $x \notin Y - E$, so $E - Y \neq Y - E$. A similar situation occurs if there's a $y \in Y$ but $y \notin E$.)

2.109 $D \cup E$ *may be* a subset of D: for $D_1 = E_1 = \{1\}$, we have $D_1 \cup E_1 = \{1\} \subseteq D$; for $D_2 = \varnothing$ and $E_2 = \{1\}$, we have $D_2 \cup E_2 = \{1\} \not\subseteq D_2$.

2.110 $D \cap E$ *must be* a subset of D: every element of $D \cap E$ is by definition an element of both D and E, which means that any $x \in D \cap E$ must by definition satisfy $x \in D$.

2.111 $D - E$ *must be* a subset of D: every element of $D - E$ is by definition an element of D (but not of E), which means that any $x \in D - E$ must by definition satisfy $x \in D$.

2.112 $E - D$ contains no elements in D, so no element $x \in E - D$ satisfies $x \in D$—but if $E - D$ is empty, then $E - D$ is a subset of every set! For $D_1 = E_1 = \{1\}$, we have $E_1 - D_1 = \{\} \subseteq D$; for $D_2 = \varnothing$ and $E_2 = \{1\}$, we have $E_2 - D_2 = \{1\} \not\subseteq D_2$. Thus $E - D$ *may be* a subset of D.

2.113 ${\sim}D$ contains no elements in D, so no element $x \in {\sim}D$ satisfies $x \in D$—but if ${\sim}D$ is empty, then ${\sim}D$ is a subset of every set, including D! For $D_1 = U$ (where U is the universe), we have ${\sim}D_1 = \{\} \subseteq D_1$; for $D_2 = \varnothing$, we have ${\sim}D_2 = U \not\subseteq D_2$. Thus ${\sim}D$ *may be* a subset of D.

2.114 Not disjoint: $1 \in F$ and $1 \in G$.

2.115 Not disjoint: $3 \in {\sim}F$ and $3 \in G$.

2.116 Disjoint: $F \cap G = \{1\}$ and $1 \notin H$.

2.117 Disjoint: by definition, no element is in both a set and the complement of that set.

2.118 $S \cup T$ is smallest when one set is a subset of the other. Specifically, if $S = \{1,2,\ldots,n\}$ and $T = \{1,2,\ldots,m\}$, then $S \cup T = \{1,2,\ldots,\max(n,m)\}$ and has cardinality $\max(n,m)$.

2.119 $S \cap T$ is smallest when the two sets are disjoint. Specifically, if $S = \{1,2,\ldots,n\}$ and $T = \{-1,-2,\ldots,-m\}$, then $S \cap T = \varnothing$ and has cardinality 0.

2.120 $S - T$ is smallest when as many elements of S as possible are also in T. Specifically, if $S = \{1,2,\ldots,n\}$ and $T = \{1,2,\ldots,m\}$, then $S - T = \{m+1,m+2,\ldots,n\}$ if $n \geq m+1$, and otherwise has cardinality 0.

2.121 $S \cup T$ is largest when the two sets are disjoint. Specifically, if $S = \{1,2,\ldots,n\}$ and $T = \{n+1,n+2,\ldots,n+m\}$, then $S \cup T = \{1,2,\ldots,n+m\}$ and has cardinality $n+m$.

2.122 $S \cap T$ is largest when one set is a subset of the other. Specifically, if $S = \{1,2,\ldots,n\}$ and $T = \{1,2,\ldots,m\}$, then $S \cap T = \{1,2,\ldots,\min(n,m)\}$ and has cardinality $\min(n,m)$.

2.123 $S - T$ is largest when no element T is also in S, so that $S - T = S$. Specifically, if $S = \{1,2,\ldots,n\}$ and $T = \{-1,-2,\ldots,-m\}$, then $S - T = \{1,2,\ldots,n\}$ and has cardinality n.

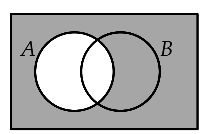
2.124 $|A \cap B| = 2$, $|A \cap C| = 1$, and $|B \cap C| = 1$.

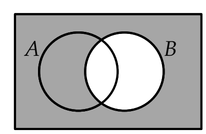
2.125 We have $|A \cap B| = 2$, $|A \cap C| = 1$, and $|B \cap C| = 1$; and furthermore $|A \cup B| = 5$, $|A \cup C| = 3$, and $|B \cup C| = 4$. Thus the Jaccard similarity of A and B is $2/5 = 0.4$; the Jaccard similarity of A and C is $1/3 = 0.33$; and the Jaccard similarity of B and C is $1/4 = 0.25$.

2.126 No. If $A = \{1\}$ and $B = \{1,2,3\}$ and $C = \{2,3,4\}$, then A's closest set is B but B's closest set is C, not A.

2.127 Still no! If $A = \{1\}$ and $B = \{1,2,3\}$ and $C = \{2,3,4\}$, then A's closest set is B (1/3 for B versus 0/4 for C) but B's closest set is C (2/4), not A (1/3).

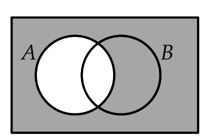
2.128 True. Looking at Venn diagrams for ${\sim}A$ and ${\sim}B$, we see that

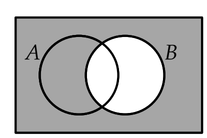


unioned with


includes every element not in $A \cap B$. Thus ${\sim}A \cup {\sim}B$ contains precisely those elements not in $A \cap B$, and $A \cap B = {\sim}({\sim}A \cup {\sim}B)$.

2.129 True. Looking at Venn diagrams for ${\sim}A$ and ${\sim}B$, we see that

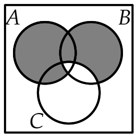


intersected with


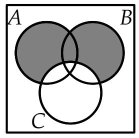
includes every element not in $A \cup B$. Thus ${\sim}A \cap {\sim}B$ contains precisely those elements not in $A \cup B$, and $A \cup B = {\sim}({\sim}A \cap {\sim}B)$.

2.130 No, they're not the same. The left-hand side includes elements of $A \cap {\sim}B \cap C$; the right-hand side doesn't:

$(A - B) \cup (B - C)$

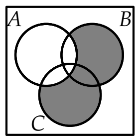


$(A \cup B) - C$



For example, suppose $A = C = \{1\}$ and $B = \varnothing$. Then $(A - B) \cup (B - C) = (\{1\} - \varnothing) \cup (\varnothing - \{1\}) = \{1\} - \varnothing = \{1\}$, but $(A \cup B) - C = (\{1\} \cup \varnothing) - \{1\} = \{1\} - \{1\} = \varnothing$.

2.131 Yes, they are the same. The left-hand side contains any element that's in B but not A and also in C but not A—in other words, an element in B and C but not A. That's just the definition of $(B \cap C) - A$:



2.132 There are five ways of partitioning $\{1, 2, 3\}$:

- All separate: $\{\{1\}, \{2\}, \{3\}\}$.
- 1 alone; 2 and 3 together: $\{\{1\}, \{2, 3\}\}$.
- 2 alone; 1 and 3 together: $\{\{2\}, \{1, 3\}\}$.
- 3 alone; 1 and 2 together: $\{\{3\}, \{1, 2\}\}$.
- All together: $\{\{1, 2, 3\}\}$.

2.133 A partition that does it is

$$\{\{\text{Alice}, \text{Bob}, \text{David}\}, \{\text{Charlie}\}, \{\text{Eve}, \text{Frank}\}\}.$$

The ABD set has intracluster distances $\{0.0, 1.7, 0.8, 1.1\}$; the EF set has intracluster distances $\{0.0, 1.9\}$; and the only intracluster distance in the C set is 0.0. The largest of these distances is less than 2.0.

2.134 Just put each person in his or her own subset. The only way $x \in S_i$ and $y \in S_i$ is when $x = y$, so the intracluster distance is 0.0.

2.135 Again, just put all people in their own subset. The intercluster distance is the largest entry in the table, namely 7.8.

2.136 No, it's not a partition of S, because the sets are not disjoint. For example, $6 \in W$ and $6 \in H$.

2.137 $\{\{\}, \{1\}, \{a\}, \{1, a\}\}$

2.138 $\{\{\}, \{1\}\}$

2.139 $\{\{\}\}$

2.140 $\mathscr{P}(\mathscr{P}(1)) = \mathscr{P}(\{\{\}, \{1\}\})$. This set is the power set of a 2-element set, and so it contains four elements: the empty set, the two singleton sets, and the set containing both elements:

$$\left\{ \begin{array}{l} \{\}, \\ \{\{\}\}, \\ \{\{1\}\}, \\ \{\{\}, \{1\}\} \end{array} \right\}.$$

Section 2.4: Sequences: Ordered Collections

2.141 $\{\langle 1,1\rangle, \langle 1,4\rangle, \langle 1,16\rangle, \langle 2,1\rangle, \langle 2,4\rangle, \langle 2,16\rangle, \langle 3,1\rangle, \langle 3,4\rangle, \langle 3,16\rangle\}$

2.142 $\{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 4,1\rangle, \langle 4,2\rangle, \langle 4,3\rangle, \langle 16,1\rangle, \langle 16,2\rangle, \langle 16,3\rangle\}$

2.143 $\{\langle 1,1,1\rangle\}$

2.144 $\{\langle 1,2,1\rangle, \langle 1,2,4\rangle, \langle 1,2,16\rangle, \langle 1,3,1\rangle, \langle 1,3,4\rangle, \langle 1,3,16\rangle, \langle 2,2,1\rangle, \langle 2,2,4\rangle, \langle 2,2,16\rangle, \langle 2,3,1\rangle, \langle 2,3,4\rangle, \langle 2,3,16\rangle\}$

2.145 $A = \{1,2\}$ and $B = \{1\}$.

2.146 $T = \{2,4\}$. Because $\langle ?,2\rangle, \langle ?,4\rangle \in ? \times T$, we know that $2 \in T$ and $4 \in T$. And because $S \times \{2,4\}$ contains 16 elements already, there can't be any other elements in T.

2.147 $T = \varnothing$. If there were any element $t \in T$, then $\langle 1,t\rangle \in S \times T$—but we're told that $S \times T$ is empty, so there can't be any $t \in T$.

2.148 For $\langle x,x\rangle$ to be in $S \times T$, we need $x \in S \cap T$; for $\langle x,x\rangle$ *not* to be in $S \times T$, we need $x \notin S \cap T$. Thus $3 \in T$, and $1,2,4,5,\ldots,8 \notin T$. But T can contain any other element not in S: for example, if $T = \{3,9\}$ then $S \times T = \{\langle 1,3\rangle, \ldots, \langle 8,3\rangle, \langle 1,9\rangle, \ldots, \langle 8,9\rangle\}$ and $T \times S = \{\langle 3,1\rangle, \ldots, \langle 3,8\rangle, \langle 9,1\rangle, \ldots, \langle 9,8\rangle\}$. The only element that appears in both of these sets is $\langle 3,3\rangle$.

2.149 Every pair in $S \times T$ must also be in $T \times S$, and vice versa. This situation can happen in two ways. First, we could have $T = S = \{1,2,\ldots,8\}$, so that $S \times T = T \times S = S \times S$. Second, we could have $T = \varnothing$, because $S \times \varnothing = \varnothing \times S = \varnothing$.

2.150 $\{\mathsf{a},\mathsf{h}\} \times \{1\}$

2.151 $\{\mathsf{c},\mathsf{f}\} \times \{1,8\}$

2.152 $\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e},\mathsf{f},\mathsf{g},\mathsf{h}\} \times \{2,7\}$

2.153 $\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e},\mathsf{f},\mathsf{g},\mathsf{h}\} \times \{3,4,5,6\}$

2.154 There are 27 elements in this set: $\{\langle 0,0,0\rangle, \langle 0,0,1\rangle, \langle 0,0,2\rangle, \langle 0,1,0\rangle, \langle 0,1,1\rangle, \langle 0,1,2\rangle, \langle 0,2,0\rangle, \langle 0,2,1\rangle, \langle 0,2,2\rangle, \langle 1,0,0\rangle, \langle 1,0,1\rangle, \langle 1,0,2\rangle, \langle 1,1,0\rangle, \langle 1,1,1\rangle, \langle 1,1,2\rangle, \langle 1,2,0\rangle, \langle 1,2,1\rangle, \langle 1,2,2\rangle, \langle 2,0,0\rangle, \langle 2,0,1\rangle, \langle 2,0,2\rangle, \langle 2,1,0\rangle, \langle 2,1,1\rangle, \langle 2,1,2\rangle, \langle 2,2,0\rangle, \langle 2,2,1\rangle, \langle 2,2,2\rangle\}$

2.155 Omitting the angle brackets and commas, the set is: $\{\mathtt{ACCE}, \mathtt{ACDE}, \mathtt{ADCE}, \mathtt{ADDE}, \mathtt{BCCE}, \mathtt{BCDE}, \mathtt{BDCE}, \mathtt{BDDE}\}$.

2.156 Omitting the angle brackets and commas, the set is: $\{0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111\}$.

2.157 Σ^8

2.158 $(\Sigma - \{\mathtt{A},\mathtt{E},\mathtt{I},\mathtt{O},\mathtt{U}\})^5$

2.159 Here's one reasonably compact way of representing the answer: in a sequence of 6 symbols, we'll allow the ith symbol to be any element of Σ, while the others must not be vowels. The desired set is:

$$\bigcup_{i=0}^{5} \left[(\Sigma - \{\mathtt{A},\mathtt{E},\mathtt{I},\mathtt{O},\mathtt{U}\})^i \times \Sigma \times (\Sigma - \{\mathtt{A},\mathtt{E},\mathtt{I},\mathtt{O},\mathtt{U}\})^{5-i}\right].$$

(There are many other ways to express the answer too.)

2.160 Here's one of many ways to express this set:

$$\bigcup_{v \in \{\mathtt{A},\mathtt{E},\mathtt{I},\mathtt{O},\mathtt{U}\}} \left[(\Sigma - \{\mathtt{A},\mathtt{E},\mathtt{I},\mathtt{O},\mathtt{U}\}) \cup \{v\}\right]^6.$$

2.161 $\sqrt{1^2+3^2} = \sqrt{1+9} = \sqrt{10}$

2.162 $\sqrt{2^2+(-2)^2} = \sqrt{4+4} = \sqrt{8}$

2.163 $\sqrt{4^2+0^2} = \sqrt{16+0} = \sqrt{16} = 4$

2.164 $\langle 1+2, 3+(-2)\rangle = \langle 3,1\rangle$

2.165 $\langle 3\cdot(-3), 3\cdot(-1)\rangle = \langle -9,-3\rangle$

2.166 $\langle 2\cdot 1 + 4 - 3\cdot 2, 2\cdot 3 + 0 - 3\cdot(-2)\rangle = \langle 0,12\rangle$

2.167 $\|a\| + \|c\| = \sqrt{10} + 4 \approx 7.1623$, while $\|a+c\| = \|\langle 5,3\rangle\| = \sqrt{5^2+3^2} = \sqrt{34} \approx 5.8310$.

2.168 $\|a\| + \|b\| = \sqrt{10} + \sqrt{8} \approx 5.9907$, while $\|a+b\| = \|\langle 3,-1\rangle\| = \sqrt{3^2+(-1)^2} = \sqrt{10} \approx 3.1623$.

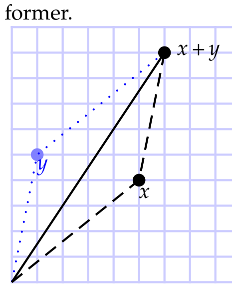
2.169 $3\|d\| = 3\sqrt{(-3)^2+(-1)^2} = 3\sqrt{10} \approx 9.4868$, and we get the same value for $\|3d\| = \|\langle -9,-3\rangle\| = \sqrt{(-9)^2+(-3)^2} = \sqrt{90} \approx 9.4868$.

2.170 Here's a derivation:

$$\begin{aligned}
\|ax\| &= \sqrt{\sum_i [(ax)_i]^2} && \textit{definition of length} \\
&= \sqrt{\sum_i a^2 x_i^2} && \textit{definition of vector times scalar} \\
&= \sqrt{a^2 \sum_i x_i^2} && a^2 \textit{ is the same for every } i \\
&= \sqrt{a^2}\sqrt{\sum_i x_i^2} && (2.1.5) \\
&= a\sqrt{\sum_i x_i^2} && \textit{definition of square root} \\
&= a\|x\|. && \textit{definition of length}
\end{aligned}$$

2.171 We have $\|x\| + \|y\| = \|x+y\|$ whenever x and y "point in exactly the same direction"—that is, when we can write $x = ay$ for some scalar $a \geq 0$. In this case, we have $\|x\| + \|y\| = \|x\| + \|ax\| = (1+a)\|x\|$, and we also have $\|x+y\| = \|x+ax\| = \|(1+a)x\| = (1+a)\|x\|$. When $x \neq ay$, then the two sides aren't equal.

Visually, the sum of the lengths of the dashed lines in this picture show $\|x\| + \|y\|$ (and the dotted lines show $\|y\| + \|x\|$), while the solid line has length $\|x+y\|$. Because the latter "cuts the corner" whenever x and y don't point in exactly the same direction, it's smaller than the former.



2.172 $1\cdot 2 + 3\cdot -2 = 2 - 6 = -4$

2.173 $1\cdot -3 + 3\cdot -1 = -3 - 3 = -6$

2.174 $4\cdot 4 + 0\cdot 0 = 16 + 0 = 16$

2.175 The Manhattan distance is $|1-2| + |3 - -2| = 1 + 5 = 6$; the Euclidean distance is $\sqrt{(1-2)^2 + (3 - -2)^2} = \sqrt{1^2+5^2} = \sqrt{26}$.

2.176 The Manhattan distance is $4 + 4 = 8$; the Euclidean distance is $\sqrt{4^2+4^2} = \sqrt{32} = 4\sqrt{2}$.

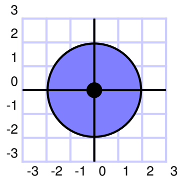
2.177 The Manhattan distance is $2 + 2 = 4$; the Euclidean distance is $\sqrt{2^2+2^2} = \sqrt{8} = 2\sqrt{2}$.

2.178 The largest possible Euclidean distance is 1—for example, if $x = \langle 0,0\rangle$ and $y = \langle 1,0\rangle$.

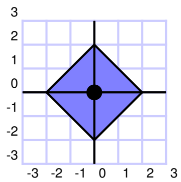
2.179 The smallest possible Euclidean distance is $1/\sqrt{2}$: if $x = \langle 0,0\rangle$ and $y = \langle 0.5,0.5\rangle$, then the Manhattan distance is indeed $1 = 0.5 + 0.5$, while the Euclidean distance is $\sqrt{0.5^2+0.5^2} = \sqrt{0.5}$.

2.180 The smallest possible Euclidean distance is $1/\sqrt{n}$—for example, if $x = \langle 0, 0, \ldots, 0 \rangle$ and $y = \langle \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \rangle$, then the Manhattan distance is indeed $1 = \frac{1}{n} + \frac{1}{n} + \cdots + \frac{1}{n}$, while the Euclidean distance is $\sqrt{n \cdot (\frac{1}{n})^2} = \sqrt{1/n}$.

2.181



2.182



2.183 Under Manhattan distance, the point $\langle 16, 40 \rangle$ has distance $12 + 2 = 14$ from g and has distance $8 + 7 = 15$ from s, so it's closer to g.

Under Euclidean distance, the point $\langle 16, 40 \rangle$ has distance $\sqrt{12^2 + 2^2} = \sqrt{146} \approx 12.08305 \cdots$ from g and distance $\sqrt{8^2 + 7^2} = \sqrt{113} \approx 10.6301 \cdots$ from s, so it's closer to s.

2.184 The Manhattan distance from the point $\langle 8 + \delta, y \rangle$ to s is $\delta + |33 - y|$. The Manhattan distance from this point to g is $4 + \delta + |42 - y|$. Because δ appears in both formulas, the point is closer to g exactly when $4 + |42 - y| < |33 - y|$.

If $y < 33$, then these distances are $4 + 42 - y$ and $33 - y$; the distance to g is always larger.

If $33 \leq y < 42$, then these distances are $4 + 42 - y$ and $y - 33$; the former is smaller when $4 + 42 - y < y - 33$, which, solving for y, occurs when $2y > 79$. The distance to g is smaller when $y > 39.5$.

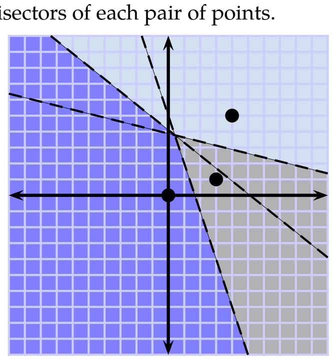
If $y > 42$, then these distances are $4 + y - 42$ and $y - 33$; the distance to g is always smaller.

Thus s is closer if $y < 39.5$ and g is closer if $y > 39.5$ (and they're equidistant if $y = 39.5$).

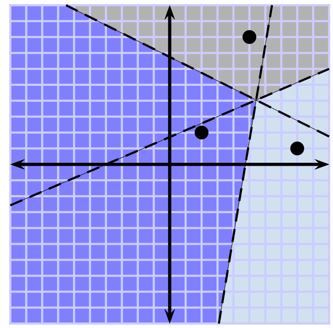
2.185 A point $\langle 4 - \delta, y \rangle$ is 4 units closer to g than to s in the horizontal direction. Thus, unless the point is 4 or more units closer to g than to s vertically, g is closer. To be 4 units closer vertically means $|42 - y| < 4 + |33 - y|$, which happens exactly when $y < 35.5$.

2.186 $3|x - 8| + 1.5|y - 33|$

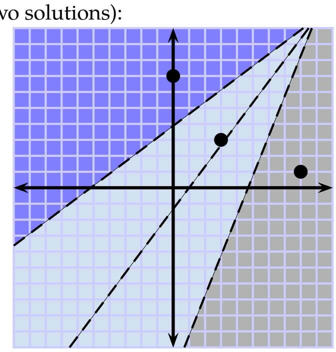
2.187 An image of the Voronoi diagram is shown below. The three different colors represent the three regions; the dashed lines represent the bisectors of each pair of points.



2.188 As before, the three different colors represent the three regions; the dashed lines represent the bisectors of each pair of points:



2.189 Here is an image of the diagram (with notation as in the previous two solutions):



2.190 For a pair of points p and q, we will need to be able to find the line that bisects them. The midpoint of p and q is on that line—namely the point $\langle (p_1 + q_1)/2, (p_2 + q_2)/2 \rangle$. The slope of the bisecting line is the negative reciprocal of the slope of the line joining p and q—that is, the slope of the bisecting line is $-(p_1 + q_1)/(p_2 + q_2)$.

Given the above, the algorithm is fairly straightforward: given three points p, q, r, we find the bisector for each pair of points. The portion of the plane closest to p is that portion that's both on p's side of the p–q bisector and on p's side of the p–r bisector. The other regions are analogous.

Coding this in a programming language requires some additional detail in representing lines, but the idea is just as described here. Here is a solution in Python:

```
def midpoint(a,b):
    '''Finds the point halfway between a and b.'''
    return [(a[0] + b[0]) / 2.0,
            (a[1] + b[1]) / 2.0]

def bisectorSlope(a,b):
    '''Finds the slope of the bisector
       between a and b.'''
    return -(a[0] - b[0]) / (a[1] - b[1])

def bisectorRegion(a,b):
    '''Returns a representation of the half plane on a's
       side of bisector between a and b.'''
    # Case I:  the bisector of a,b is a vertical line.
    if a[1] == b[1]:
        if a[0] < midpoint(a,b)[0]: # a is to the left
            return "x < %.2f" % midpoint(a,b)[0]
        else:
            return "x > %.2f" % midpoint(a,b)[0]

    # Case II:  the bisector of a,b is a horizontal line.
    slope = bisectorSlope(a,b)
    if slope == 0:
        if a[1] < midpoint(a,b)[1]: # a is below
            return "y < %.2f" % midpoint(a,b)[1]
        else:
            return "y > %.2f" % midpoint(a,b)[1]

    # Case III:  the bisector of a,b has nonzero,
    #            noninfinite slope.
    yIntercept = - slope * midpoint(a,b)[0]
                 + midpoint(a,b)[1]
    if a[1] < b[1]: # a is below the bisector
        return "y < %.2f*x + %.2f" % (slope, yIntercept)
    else:
        return "y > %.2f*x + %.2f" % (slope, yIntercept)

def voronoiRegion(a,b,c):
    '''Returns a representation of the portion of the
       plane closer to a than to either b or c.'''
    return bisectorRegion(a,b)
          + " and " + bisectorRegion(a,c)
```

2.191 6 by 3

2.192 6

2.193 $\langle 4,1\rangle$, $\langle 5,1\rangle$, and $\langle 7,3\rangle$

2.194

$$3M = \begin{bmatrix} 9 & 27 & 6 \\ 0 & 27 & 24 \\ 18 & 6 & 0 \\ 21 & 15 & 15 \\ 21 & 6 & 12 \\ 3 & 18 & 21 \end{bmatrix}$$

2.195

$$\begin{bmatrix} 0 & 8 & 0 \\ 9 & 6 & 0 \\ 2 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 7 & 2 & 7 \\ 3 & 5 & 6 \\ 1 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 7 & 10 & 7 \\ 12 & 11 & 6 \\ 3 & 5 & 8 \end{bmatrix}$$

2.196 Undefined; the dimensions of B and F don't match.

2.197

$$\begin{bmatrix} 3 & 1 \\ 0 & 8 \end{bmatrix} + \begin{bmatrix} 8 & 4 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 11 & 5 \\ 3 & 10 \end{bmatrix}$$

2.198

$$\begin{bmatrix} 0 & 8 & 0 \\ 9 & 6 & 0 \\ 2 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 0 & 8 & 0 \\ 9 & 6 & 0 \\ 2 & 3 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 16 & 0 \\ 18 & 12 & 0 \\ 4 & 6 & 6 \end{bmatrix}$$

2.199

$$-2 \cdot \begin{bmatrix} 3 & 1 \\ 0 & 8 \end{bmatrix} = \begin{bmatrix} -6 & -2 \\ 0 & -16 \end{bmatrix}$$

2.200

$$0.5 \cdot \begin{bmatrix} 1 & 2 & 9 \\ 5 & 4 & 0 \end{bmatrix} = \begin{bmatrix} 0.5 & 1 & 4.5 \\ 2.5 & 2 & 0 \end{bmatrix}$$

2.201

$$\begin{bmatrix} 0 & 8 & 0 \\ 9 & 6 & 0 \\ 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} 5 & 8 \\ 7 & 5 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 56 & 40 \\ 87 & 102 \\ 40 & 37 \end{bmatrix}$$

2.202

$$\begin{bmatrix} 0 & 8 & 0 \\ 9 & 6 & 0 \\ 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} 7 & 2 & 7 \\ 3 & 5 & 6 \\ 1 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 24 & 40 & 48 \\ 81 & 48 & 99 \\ 26 & 25 & 47 \end{bmatrix}$$

2.203 Undefined: A has three columns but F has only two rows, so the dimensions don't match up.

2.204 Undefined: B has two columns but C has three rows, so the dimensions don't match up.

2.205

$$\begin{bmatrix} 3 & 1 \\ 0 & 8 \end{bmatrix} \begin{bmatrix} 8 & 4 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 27 & 14 \\ 24 & 16 \end{bmatrix}$$

2.206

$$\begin{bmatrix} 8 & 4 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & 8 \end{bmatrix} = \begin{bmatrix} 24 & 40 \\ 9 & 19 \end{bmatrix}$$

2.207

$$0.25 \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} + 0.75 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} .25 & 0 & 0 \\ .25 & .75 & 0 \\ 1 & 1 & .75 \end{bmatrix}$$

2.208

$$0.5 \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} + 0.5 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} .5 & 0 & 0 \\ .5 & .5 & 0 \\ 1 & 1 & .5 \end{bmatrix}$$

2.209 Here's one example (there are many more):

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

2.210 Here's a solution in Python:

```
def average(imageA, imageB, lambda):
    width = imageA.getWidth()
    height = imageA.getHeight()
    if lambda < 0 or lambda > 1
         or imageB.getWidth() != width or
         or imageB.getHeight() != height:
       return "Error!"
    newImage = Image(width, height)
    for x in range(height):
       for y in range(width):
          avg = lambda * imageA.getPixel(x,y)
                  + (1 - lambda) * imageB.getPixel(x,y)
          newImage.setPixel(x, y, avg)
    return newImage
```

2.211 Here's a derivation:

$$\begin{aligned}
(AI)_{i,j} &= \sum_{k=1}^{n} A_{i,k} I_{k,j} && \textit{definition of matrix multiplication} \\
&= \sum_{k=1}^{n} A_{i,k} \cdot \begin{cases} 0 & \text{if } k \neq j \\ 1 & \text{if } k = j \end{cases} && \textit{definition of I} \\
&= A_{i,j} \cdot 1 && \textit{simplfying} \\
&= A_{i,j}.
\end{aligned}$$

Thus for any indices i and j we have that $(AI)_{i,j} = A_{i,j}$.

2.212

$$\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 7 & 9 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 23 & 30 \\ 10 & 13 \end{bmatrix}$$

2.213 $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$

2.214 $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}^2 = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$

2.215 $\begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}^2 \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 34 & 21 \\ 21 & 13 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \cdot \begin{bmatrix} 55 & 34 \\ 34 & 21 \end{bmatrix}$

2.216 We're given that $y + w = 0$ and $2y + w = 1$. From the former, we know $y = -w$; plugging this value for y into the latter yields $-2w + w = 1$, so we have $w = -1$ and $y = 1$.

From $x + z = 1$ and $2x + z = 0$, similarly, we know $z = 1 - x$ and thus that $2x + 1 - x = 0$. Therefore $x = -1$ and $z = 2$. The final inverse is therefore

$$\begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}.$$

And, indeed,

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

as desired.

2.217 $\begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$

2.218 $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

2.219 $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

2.220 Suppose that the matrix

$$\begin{bmatrix} x & y \\ z & w \end{bmatrix}$$

were the inverse of

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then we'd need $x + z = 1$ (to make the $\langle 1, 1 \rangle$st entry of the product correct) and $x + z = 0$ (to make the $\langle 1, 2 \rangle$th entry of the product correct). But it's not possible for $x + z$ to simultaneously equal 0 and 1!

2.221 $[0, 0, 0, 0, 0, 0, 0]$

2.222 $[0, 1, 1, 0, 0, 1, 1]$

2.223 $[1, 0, 0, 1, 1, 0, 0]$

Section 2.5: Functions

2.224 $f(3) = (3^2 + 3) \bmod 8 = 12 \bmod 8 = 4$

2.225 $f(7) = (7^2 + 3) \bmod 8 = 52 \bmod 8 = 4$

2.226 $f(x) = 3$ for $x \in \{0, 4\}$.

2.227

x	$f(x)$
0	3
1	4
2	7
3	4
4	3
5	4
6	7
7	4

2.228

$$\textbf{quantize}(n) := 52 \cdot \left\lfloor \frac{n}{52} \right\rfloor + 28$$

2.229 The domain is $\{0, 1, \ldots, 255\} \times \{1, \ldots, 256\}$. The range is $\{0, 1, \ldots, 255\}$: all output colors are possible, because $\textbf{quantize}(n, 256) = n$.

2.230 The step size is $\lceil \frac{256}{k} \rceil$, so we have

$$\textbf{quantize}(n,k) := \left\lceil \frac{256}{k} \right\rceil \cdot \left\lfloor \frac{n}{\lceil \frac{256}{k} \rceil} \right\rfloor + \frac{\lceil \frac{256}{k} \rceil}{2}.$$

Note that this expression will do something a little strange for large k: for example, when $k = 200$, we have $\lceil \frac{256}{k} \rceil = 2$—so the first 127 quanta correspond to colors $\{1, 3, 5, \ldots, 255\}$, and the remaining 73 quanta are never used (because they're bigger than 256).

2.231 A function $\textbf{quantize}(n) : \{0, 1, \ldots, 255\} \to \{a_1, a_2, \ldots, a_k\}$ can be c-to-1 only if $k \mid 256$. (For example, there's no way to quantize 256 colors into three precisely equal pieces, because 256 is not evenly divisible by 3.) So k must be a power of two; specifically, we have to have $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$.

2.232 Here's a solution in Python:

```
output = image.copy()
stepSize = int(math.ceil(256.0 / k))
for x in range(image.getWidth()):
   for y in range(image.getHeight()):
      color = stepSize / 2
               + stepSize * int(image.getPixel(x,y) / stepSize)
      output.setPixel(x, y, color)
```

2.233 The domain is $\mathbb{R}$; the range is $\mathbb{R}^{\geq 0}$.

2.234 The domain is $\mathbb{R}$; the range is $\mathbb{Z}$.

2.235 The domain is $\mathbb{R}$; the range is $\mathbb{R}^{>0}$.

2.236 The domain is $\mathbb{R}^{>0}$; the range is $\mathbb{R}$.

2.237 The domain is $\mathbb{Z}$; the range is $\{0, 1\}$.

2.238 The domain is $\mathbb{Z}^{\geq 2}$; the range is $\{0, 1, 2\}$.

2.239 The domain is $\mathbb{Z} \times \mathbb{Z}^{\geq 2}$; the range is $\mathbb{Z}^{\geq 0}$.

2.240 The domain is $\mathbb{Z}$; the range is $\{\text{True}, \text{False}\}$.

2.241 The domain is $\bigcup_{n \in \mathbb{Z}^{\geq 1}} \mathbb{R}^n$; the range $\mathbb{R}^{\geq 0}$.

2.242 The domain is $\mathbb{R}$; the range is the set of all unit vectors in $\mathbb{R}^2$—that is, the set $\{x \in \mathbb{R}^2 : \|x\| = 1\}$.

2.243 The function *add* can be written as $add(\langle h, m \rangle, x) := \langle [(h + \lfloor \frac{m+x}{60} \rfloor - 1) \bmod 12] + 1, m + x \bmod 60 \rangle$

2.244 $(f \circ f)(x) = f(f(x)) = x \bmod 10$

2.245 $(h \circ h)(x) = h(h(x)) = 4x$

2.246 $(f \circ g)(x) = f(g(x)) = x + 3 \bmod 10$

2.247 $(g \circ h)(x) = g(h(x)) = 2x + 3$

2.248 $(h \circ g)(x) = h(g(x)) = 2(x + 3) = 2x + 6$

2.249 $(f \circ h)(x) = f(h(x)) = 2x \bmod 10$

2.250 $(f \circ g \circ h)(x) = f(g(h(x))) = 2x + 3 \bmod 10$

2.251 $(g \circ h)(x) = g(h(x)) = 2 \cdot h(x)$, so we need $h(x) = \frac{3}{2}x + \frac{1}{2}$.

2.252 $(h \circ g)(x) = h(g(x)) = h(2x)$, so we need $h(2x) = 3x + 1$ for every x. That's true when $h(x) = \frac{3}{2}x + 1$.

2.253 Yes, $f(x) = x$ is onto; every output value is hit.

2.254

$$\begin{aligned} f(0) &= 0^2 \bmod 4 = 0 \bmod 4 = 0 \\ f(1) &= 1^2 \bmod 4 = 1 \bmod 4 = 1 \\ f(2) &= 2^2 \bmod 4 = 4 \bmod 4 = 0 \\ f(3) &= 3^2 \bmod 4 = 9 \bmod 4 = 1 \end{aligned}$$

There's no x such that $f(x) = 2$ or $f(x) = 3$, so the function is not onto.

2.255

$$\begin{aligned} f(0) &= 0^2 - 0 \bmod 4 = 0 - 0 \bmod 4 = 0 \\ f(1) &= 1^2 - 1 \bmod 4 = 1 - 1 \bmod 4 = 0 \\ f(2) &= 2^2 - 2 \bmod 4 = 4 - 2 \bmod 4 = 2 \\ f(3) &= 3^2 - 3 \bmod 4 = 9 - 3 \bmod 4 = 2 \end{aligned}$$

There's no x such that $f(x) = 1$ or $f(x) = 3$, so the function is not onto.

2.256 Yes, this function, which we could write as $f(x) = 3 - x$, is onto; every output value is hit.

2.257 There's no x such that $f(x) = 0$ or $f(x) = 3$, so the function is not onto.

2.258

$$\begin{aligned} f(0) &= 0^2 \bmod 8 = 0 \bmod 8 = 0 \\ f(1) &= 1^2 \bmod 8 = 1 \bmod 8 = 1 \\ f(2) &= 2^2 \bmod 8 = 4 \bmod 8 = 4 \\ f(3) &= 3^2 \bmod 8 = 9 \bmod 8 = 1 \end{aligned}$$

This function is not one-to-one, because $f(1) = f(3)$.

2.259

$$\begin{aligned} f(0) &= 0^3 \bmod 8 = 0 \bmod 8 = 0 \\ f(1) &= 1^3 \bmod 8 = 1 \bmod 8 = 1 \\ f(2) &= 2^3 \bmod 8 = 8 \bmod 8 = 0 \\ f(3) &= 3^3 \bmod 8 = 27 \bmod 8 = 7 \end{aligned}$$

This function is not one-to-one, because $f(0) = f(2)$.

2.260

$$\begin{aligned} f(0) &= 0^3 - 0 \bmod 8 = 0 \bmod 8 = 0 \\ f(1) &= 1^3 - 1 \bmod 8 = 0 \bmod 8 = 0 \\ f(2) &= 2^3 - 2 \bmod 8 = 6 \bmod 8 = 6 \\ f(3) &= 3^3 - 3 \bmod 8 = 24 \bmod 8 = 0 \end{aligned}$$

This function is not one-to-one, because $f(0) = f(1) = f(3)$.

2.261

$$\begin{aligned} f(0) &= 0^3 + 0 \bmod 8 = 0 \bmod 8 = 0 \\ f(1) &= 1^3 + 2 \bmod 8 = 3 \bmod 8 = 3 \\ f(2) &= 2^3 + 4 \bmod 8 = 12 \bmod 8 = 4 \\ f(3) &= 3^3 + 6 \bmod 8 = 33 \bmod 8 = 1 \end{aligned}$$

This function is one-to-one, as no output is hit more than once.

2.262 This function is not one-to-one, because $f(1) = f(3)$.

2.263 Every element *except* $A[1]$ has a parent. An element i is a parent only if it has a left child, which occurs when $2i \leq n$. Thus $\textbf{parent} : \{2, 3, \ldots, n\} \to \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$. The domain is $\{2, 3, \ldots, n\}$ and the range is $\{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$. And **parent** is *not* one-to-one: for example, $\textbf{parent}(2) = \textbf{parent}(3) = 1$.

2.264 An element i has a left child when $2i \leq n$ and a right child when $2i + 1 \leq n$. Every element *except* $A[1]$ is the child of some other element; even-numbered elements are left children and odd-numbered elements are right children.

Thus **left** : $\{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\} \to \{k \geq 2 : 2 \mid k \text{ and } k \leq n\}$.

And **right** : $\{1, 2, \ldots, \lfloor \frac{n-1}{2} \rfloor\} \to \{k \geq 2 : 2 \not\mid k \text{ and } k \leq n\}$.

Both **left** and **right** are one-to-one: if $i \neq i'$, then $2i \neq 2i'$ and $2i + 1 \neq 2i' + 1$. Thus there are no two elements i and i' that have **left**(i) = **right**(i') or **right**(i) = **left**(i').

2.265 (**parent** $\circ$ **left**)(i) = **parent**(**left**(i)) = $\lfloor \frac{2i}{2} \rfloor = \lfloor i \rfloor = i$.

In other words, **parent** $\circ$ **left** is the identity function: for any i, we have **parent** $\circ$ **left**$(i) = i$. In English, this composition asks for the parent of the left child of a given index—but the parent of i's left child simply *is* i!

2.266 (**parent** $\circ$ **right**)(i) = **parent**(**right**(i)) = $\lfloor \frac{2i+1}{2} \rfloor = \lfloor i + \frac{1}{2} \rfloor = i$. In other words, **parent** $\circ$ **right** is the identity function: for any i, we have **parent** $\circ$ **right**$(i) = i$. In English, this composition asks for the parent of the right child of a given index—but the parent of i's right child simply *is* i!

2.267 (**left** $\circ$ **parent**)(i) = **left**(**parent**(i)) = $2 \lfloor \frac{i}{2} \rfloor$ for any $i \geq 2$. In other words, this function is $f(i) = i$ if $2 \mid i$ and $f(i) = i - 1$ if $2 \not\mid i$ (and $f(1)$ is undefined).

In English, this composition asks for the left child of a given index's parent. An index that's a left child *is* the left child of its parent; for an index that's a right child, the left child of its parent is its left-hand sibling.

2.268 (**right** $\circ$ **parent**)(i) = **right**(**parent**(i)) = $2 \lfloor \frac{i}{2} \rfloor + 1$ for any $i \geq 2$. In other words, this function is $f(i) = i + 1$ if $2 \mid i$ and $f(i) = i$ if $2 \not\mid i$ (and $f(1)$ is undefined).

In English, this composition asks for the right child of a given index's parent. An index that's a right child *is* the right child of its parent; for an index that's a left child, the right child of its parent is its right-hand sibling.

2.269 $f^{-1}(y) = \frac{y-1}{3}$

2.270 $g^{-1}(y) = \sqrt[3]{y}$

2.271 $h^{-1}(y) = \log_3 y$

2.272 This function isn't one-to-one (it can't be: there are 24 different inputs and only 12 different outputs—for example, $f(7) = f(19)$), so it doesn't have an inverse.

2.273 The degree is 3.

2.274 The degree is 3.

2.275 The degree is 2; the polynomial is $p(x) = 4x^4 + x^2 - 4x^4 = x^2$.

2.276 The largest possible degree is still 7; the smallest is 0 (if p and q are identical—for example, $p(x) = x^7$ and $q(x) = x^7$).

2.277 The largest possible degree is 14, and that's the smallest, too! If $p(x) = \sum_{i=0}^{7} a_i x^i$ and $q(x) = \sum_{i=0}^{7} b_i x^i$, where $a_7 \neq 0$ and $b_7 \neq 0$, then there's a term $a_7 b_7 x^{14}$ in the product.

2.278 The largest possible degree is 49, and that's the smallest, too! If $p(x) = \sum_{i=0}^{7} a_i x^i$ and $q(x) = \sum_{i=0}^{7} b_i x^i$, where $a_7 \neq 0$ and $b_7 \neq 0$, then

$$\begin{aligned}
p(q(x)) &= p(\sum_{i=0}^{7} b_i x^i) \\
&= \sum_{j=0}^{7} a_j \left(\sum_{i=0}^{7} b_i x^i \right)^j \\
&= a_7 \left(\sum_{i=0}^{7} b_i x^i \right)^7 + \text{lower-degree terms} \\
&= a_7 \left(b_7^7 x^7 + \text{lower-degree terms} \right)^7 + \text{lower-degree terms} \\
&= a_7 b_7^7 (x^7)^7 + \text{lower-degree terms}.
\end{aligned}$$

2.279 $p(x) = 1 + x^2$

2.280 $p(x) = x^2$ (which has a single root at $x = 0$).

2.281 $p(x) = x^2 - 1$ (which has roots at $x = -1$ and $x = 1$).

2.282 Here's a simple algorithm (though there are much faster solutions!):

findMedian(L):

Input: A list L with $n \geq 1$ elements $L[1], \ldots, L[n]$.

Output: An index $i \in \{1, 2, \ldots, n\}$ such that $L[i]$ is the maximum value in L

```
while n ≥ 1
    maxIndex := findMaxIndex(L)
    minIndex := findMinIndex(L)
    delete L[minIndex] and L[maxIndex]
return L[1]
```

Section 3.2: An Introduction to Propositional Logic

3.1 False: $2^2 + 3^2 = 4 + 9 = 13 \neq 16 = 4^2$.

3.2 False: the binary number 11010010 has the value $2 + 16 + 64 + 128 = 210$, not 202.

3.3 True: in repeated iterations, the value of x is 202, then 101, then 50, then 25, then 12, then 6, and then 3. When x is 3, then we do one last iteration and set x to 1, and the loop terminates.

3.4 $r \Leftrightarrow u \wedge v$

3.5 $p \Leftrightarrow (u \wedge v) \vee (w \wedge z)$

3.6 $q \Leftrightarrow (u \wedge v) \vee (u \wedge z) \vee (v \wedge w)$

3.7 $s \Leftrightarrow q \wedge \neg(u \wedge v)$

3.8 $s \Rightarrow \neg t$

3.9 $p \wedge q \Rightarrow \neg w$

3.10 She should answer "yes": $p \Rightarrow q$ is true whenever p is false, and "you're over 55 years old" is false for her. (That is, False $\Rightarrow$ True is true.)

3.11 To write the solution more compactly, we'll use notation for ands and ors that's similar to $\sum$ and $\prod$ notation: we will write $\bigwedge_{i=1}^n p_i$ to mean $p_1 \wedge p_2 \wedge \cdots \wedge p_n$, and $\bigvee_{i=1}^n p_i$ to mean $p_1 \vee p_2 \vee \cdots \vee p_n$.

To express "at least 3 of $\{p_1, \ldots, p_n\}$ are true," we'll require that p_i, p_j, and p_k are all true, where $i < j < k$, and taking the "or" over all the values of i, j, and k. Formally, the proposition is

$$\bigvee_{i=1}^n \bigvee_{j=i+1}^n \bigvee_{k=j+1}^n (p_i \wedge p_j \wedge p_k)$$

3.12 The easiest way to write "at least $n-1$ of $\{p_1, \ldots, p_n\}$ are true" is to say that, for every two variables from $\{p_1, \ldots, p_n\}$, at least one of the two is true. Using the same notation as in the previous exercise, we can write this proposition as

$$\bigwedge_{i=1}^n \bigwedge_{j=i+1}^n (p_i \vee p_j).$$

3.13 The identity of $\vee$ is False: $x \vee \text{False} \equiv \text{False} \vee x \equiv x$.

3.14 The identity of $\wedge$ is True: $x \vee \text{True} \equiv \text{True} \vee x \equiv x$.

3.15 The identity of $\Leftrightarrow$ is True: $x \Leftrightarrow \text{True} \equiv \text{True} \Leftrightarrow x \equiv x$.

3.16 The identity of $\oplus$ is True: $x \oplus \text{False} \equiv \text{False} \vee x \equiv x$.

3.17 The zero of $\vee$ is True: $x \vee \text{True} \equiv \text{True} \vee x \equiv \text{True}$.

3.18 The zero of $\wedge$ is False: $x \wedge \text{False} \equiv \text{False} \wedge x \equiv \text{False}$.

3.19 The operator $\Leftrightarrow$ doesn't have a zero. Because True $\Leftrightarrow x \equiv x$, the proposition True $\Leftrightarrow x$ is true when x = True and it's false when x = False. Similarly, because True $\Leftrightarrow x \equiv \neg x$, the proposition False $\Leftrightarrow x$ is false when x = True and it's true when x = False. So neither True nor False is a zero for $\Leftrightarrow$.

3.20 The operator $\oplus$ doesn't have a zero: there's no value z such that $z \oplus \text{True} \equiv z \oplus \text{False}$, so no z is a zero for $\oplus$.

3.21 The left identity of $\Rightarrow$ is True; the right identity is False: $p \Rightarrow \text{False} \equiv p$ and $\text{True} \Rightarrow p \equiv p$.

3.22 The right zero of $\Rightarrow$ is True: $p \Rightarrow \text{True} \equiv \text{True}$. But there is no left zero for implies: True $\Rightarrow p$ is equivalent to p, not to True; and False $\Rightarrow p$ is equivalent to True, not to False.

3.23 `x * y` is equivalent to $x \wedge y$.

3.24 `x + y` is equivalent to $x \vee y$.

3.25 `1 - x` is equivalent to $\neg x$.

3.26 `(x * (1 - y)) + ((1 - x) * y)` is equivalent to $(x \wedge \neg y) \vee (\neg x \wedge y)$, which is equivalent to $x \oplus y$.

3.27 $(\mathtt{p}_1 + \mathtt{p}_2 + \mathtt{p}_3 + \ldots + \mathtt{p}_n) \geq 3$

3.28 $(\mathtt{p}_1 + \mathtt{p}_2 + \mathtt{p}_3 + \ldots + \mathtt{p}_n) \geq \mathtt{n} - 1$

3.29 $x_3 \vee x_2$

3.30 $\neg x_0 \wedge \neg x_1$

3.31 The value of x is $a + b$, where $a := 4x_2 + 1x_0$ and $b := 8x_3 + 2x_1$. It takes a bit of work to persuade yourself of this fact, but it turns out that x is divisible by 5 if and only if both a and b are divisible by 5. Thus the proposition is:

$$\underbrace{(x_0 \Leftrightarrow x_2)}_{a \text{ is divisible by } 5} \wedge \underbrace{(x_1 \Leftrightarrow x_3)}_{b \text{ is divisible by } 5}.$$

You can verify that this proposition is correct with a truth table.

3.32 There are several ways to express that x is an exact power of two. Here's one:

$$[(x_1 \vee x_2 \vee x_3) \oplus x_0] \wedge [(x_0 \vee x_2 \vee x_3) \oplus x_1]$$
$$\wedge [(x_0 \vee x_1 \vee x_3) \oplus x_2] \wedge [(x_0 \vee x_1 \vee x_2) \oplus x_3]$$

This proposition expresses for each bit position i, either $x_i = 1$ and no other bit is 1, or $x_i \neq 1$ and some other bit is 1. Another version is to say that at least one bit is 1 but that no two bits are 1:

$$(x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3)$$
$$\wedge (\neg x_1 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_4) \wedge (\neg x_3 \vee \neg x_4).$$

3.33 $(x_0 \Leftrightarrow y_0) \wedge (x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2) \wedge (x_3 \Leftrightarrow y_3)$

3.34 Expressing $x \leq y$ is a bit more complicated than expressing equality. The following proposition does so by writing out "x and y are identical before the ith position, and furthermore $x_i < y_i$" for each possible index i:

$$\begin{aligned} &(y_3 \wedge \neg x_3) \\ \vee\ &[(y_3 \Leftrightarrow x_3) \wedge (y_2 \wedge \neg x_2)] \\ \vee\ &[(y_3 \Leftrightarrow x_3) \wedge (y_2 \Leftrightarrow x_2) \wedge (y_1 \wedge \neg x_1)] \\ \vee\ &[(y_3 \Leftrightarrow x_3) \wedge (y_2 \Leftrightarrow x_2) \wedge (y_1 \Leftrightarrow x_1) \wedge (y_0 \vee \neg x_0)]. \end{aligned}$$

(The last disjunct also permits $y_0 = x_0$ when $x_{1,2,3} = y_{1,2,3}$.)

3.35

$$\begin{aligned} y_0 &= \neg x_0 \\ y_1 &= (\neg x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1) \\ y_2 &= ((\neg x_0 \vee \neg x_1) \wedge x_2) \vee (x_0 \wedge x_1 \wedge \neg x_2) \\ y_3 &= ((\neg x_0 \vee \neg x_1 \vee \neg x_2) \wedge x_3) \vee (x_0 \wedge x_1 \wedge x_2 \wedge \neg x_3) \end{aligned}$$

3.36 Here is a solution in Python:

```
def variables(phi):
  if type(phi) == str:   # phi is an atomic proposition.
      return [phi]
  elif phi[0] == "not":
      return variables(phi[1])
  elif phi[0] in ["implies", "and", "or", "xor", "iff"]:
      result = variables(phi[1])
      for var in variables(phi[2]):
          if var not in result:
              result.append(var)
      return result
```

3.37 Continuing in Python:

```
def eval(phi, rho):
    if type(phi) == str:
        return rho[phi]
    elif phi[0] == "not":
        return not eval(phi[1], rho)
    elif phi[0] == "implies":
        return (not eval(phi[1], rho)) or eval(phi[2], rho)
    elif phi[0] == "and":
        return eval(phi[1], rho) and eval(phi[2], rho)
    elif phi[0] == "or":
        return eval(phi[1], rho) or eval(phi[2], rho)
    elif phi[0] == "xor":
        return eval(phi[1], rho) != eval(phi[2], rho)
    elif phi[0] == "iff":
        return eval(phi[1], rho) == eval(phi[2], rho)
```

3.38 Continuing in Python:

```
def allAssignments(vars):
    assignments = []
    for var in vars:
        newAssignmentsT = copy.deepcopy(assignments)
        newAssignmentsF = copy.deepcopy(assignments)
        for rho in newAssignmentsT:
            rho[var] = True
        for rho in newAssignmentsF:
            rho[var] = False
        assignments = newAssignmentsT + newAssignmentsF
    return assignments

def satisfiers(phi):
   goodAssignments = []
   for rho in allAssignments(variables(phi)):
      if eval(phi, rho):
         goodAssignments.append(rho)
   return goodAssignments
```

Section 3.3: Propositional Logic: Some Extensions

3.39 True

3.40 False

3.41 True

3.42 Either $p \Rightarrow (\neg p \Rightarrow (p \Rightarrow q))$ or $(p \Rightarrow \neg p) \Rightarrow (p \Rightarrow q)$ is a tautology.

3.43 $(p \Rightarrow (\neg p \Rightarrow p)) \Rightarrow q$

3.44 The implicit parentheses make the given proposition into $((p \Rightarrow \neg p) \Rightarrow p) \Rightarrow q$, which is logically equivalent to $p \Rightarrow q$.

3.45 Set $p = \text{F}, q = \text{F}, r = \text{F}$. Then $\text{F} \Rightarrow (\text{F} \Rightarrow \text{F}) \equiv \text{F} \Rightarrow \text{T} \equiv \text{T}$, but $(\text{F} \Rightarrow \text{F}) \Rightarrow \text{F} \equiv \text{T} \Rightarrow \text{F} \equiv \text{F}$.

3.46 Here's the truth table:

p	q	$p \Rightarrow q$	$(p \Rightarrow q) \Rightarrow q$	$q \Rightarrow q$	$p \Rightarrow (q \Rightarrow q)$
T	T	T	T	T	T
T	F	F	T	T	T
F	T	T	T	T	T
F	F	T	F	T	T

So $p \Rightarrow (q \Rightarrow q)$ is a tautology (because its conclusion $q \Rightarrow q$ is a tautology), but $(p \Rightarrow q) \Rightarrow q$ is false when both p and q are false.

3.47 The propositions $p \Rightarrow (p \Rightarrow q)$ and $(p \Rightarrow p) \Rightarrow q$ are logically equivalent; they're both true except when p is true and q is false. Thus they are both logically equivalent to $p \Rightarrow q$. Neither proposition is a tautology; both are satisfiable.

3.48 This answer is wrong: False $\oplus$ False is false, but when $p = q = \text{False}$, then $\neg(p \wedge q) \Rightarrow (\neg p \wedge \neg q)$ has the value

$$\begin{aligned} &\neg(\text{False} \wedge \text{False}) \Rightarrow (\neg\text{False} \wedge \neg\text{False}) \\ &\equiv \neg\text{False} \Rightarrow \text{True} \\ &\equiv \text{True} \Rightarrow \text{True} \\ &\equiv \text{True}. \end{aligned}$$

3.49 This answer is wrong: False $\oplus$ False is false, but when $p = q = \text{False}$, then $(p \Rightarrow \neg q) \wedge (q \Rightarrow \neg p)$ has the value

$$\begin{aligned} &(\text{False} \Rightarrow \neg\text{False}) \wedge (\text{False} \Rightarrow \neg\text{False}) \\ &\equiv (\text{False} \Rightarrow \text{True}) \wedge (\text{False} \Rightarrow \text{True}) \\ &\equiv \text{True} \wedge \text{True} \\ &\equiv \text{True}. \end{aligned}$$

3.50 Write φ to denote $(\neg p \Rightarrow q) \wedge \neg(p \wedge q)$, and let's build a truth table:

p	q	$\neg p$	$\neg p \Rightarrow q$	$\neg(p \wedge q)$	φ
T	T	F	T	F	F
T	F	F	T	T	T
F	T	T	T	T	T
F	F	T	F	T	F

This truth table matches the truth table for $p \oplus q$, so the solution is correct.

3.51 The given proposition is

$$\neg\left[(p \wedge \neg q \Rightarrow \neg p \wedge q) \wedge (\neg p \wedge q \Rightarrow p \wedge \neg q)\right].$$

Let's rewrite it as $\neg[(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)]$, where φ is $p \wedge \neg q$ and φ is $\neg p \wedge q$). Here's a truth table:

p	q	$\varphi = p \wedge \neg q$	$\psi = \neg p \wedge q$	$\varphi \Rightarrow \psi$	$\psi \Rightarrow \varphi$	$(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$
T	T	F	F	T	T	T
T	F	T	F	F	T	F
F	T	F	T	T	F	F
F	F	F	F	T	T	T

The desired truth table for $p \oplus q$ is the negation of the last column of this truth table—and the given proposition is $\neg[(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)]$, the negation of the last column. So the solution is correct.

3.52 One proposition that works: $(p \vee q) \Rightarrow (\neg p \vee \neg q)$. (There are many other choices too.)

3.53 Note that $p \vee (\neg p \wedge q)$ is equivalent to $p \vee q$, so we can write the given code as

```
if (x > 20 or y < 0)
then foo(x,y)
else bar(x,y)
```

3.54 Note that $(x - y) \cdot y \geq 0$ if and only if

$$(x - y \geq 0) \Leftrightarrow (y \geq 0).$$

Thus, writing $y \geq 0$ as p, and $x \geq y$ as q, the given expression is $p \vee q \vee (p \Leftrightarrow q)$. This proposition is a tautology! So we can rewrite the given code as

```
foo(x,y)
```

3.55 Write p to denote $12 \mid x$ and q to denote $4 \mid x$. Note that if p is true, then q must be true. By unnesting the conditions, we see that `foo` is called if $p \wedge \neg q$ (which is impossible). Thus we can rewrite the given code as

```
if (x % 12 == 0):
  then bar(x)
  else if (x == 17):
          then baz(x)
          else quz(x)
```

3.56 Let's build a truth table for $(\neg p \Rightarrow q) \wedge (q \wedge p \Rightarrow \neg p)$, which we'll denote by φ:

p	q	$\neg p \Rightarrow q$	$(q \wedge p \Rightarrow \neg p)$	φ
T	T	T	F	F
T	F	T	T	T
F	T	T	T	T
F	F	F	T	F

Thus φ is logically equivalent to $p \oplus q$.

3.57 First, observe that $p \Rightarrow p$ is always true, and $q \Rightarrow \text{True}$ is also always true. Thus we can immediately rewrite the given statement as

$$\begin{aligned} &(p \Rightarrow \neg p) \Rightarrow ((q \Rightarrow (p \Rightarrow p)) \Rightarrow p) \\ \equiv\ &(p \Rightarrow \neg p) \Rightarrow ((q \Rightarrow \text{True}) \Rightarrow p) \\ \equiv\ &(p \Rightarrow \neg p) \Rightarrow (\text{True} \Rightarrow p). \end{aligned}$$

If p is true, then $(p \Rightarrow \neg p) \Rightarrow (\text{True} \Rightarrow p)$ is $(\text{True} \Rightarrow \text{False}) \Rightarrow (\text{True} \Rightarrow \text{True})$, which is just False $\Rightarrow$ True, or True. If p is false, then $(p \Rightarrow \neg p) \Rightarrow (\text{True} \Rightarrow p)$ is $(\text{False} \Rightarrow \text{True}) \Rightarrow (\text{True} \Rightarrow \text{False})$, which is just True $\Rightarrow$ False, or False. Thus the whole expression is logically equivalent to p.

3.58 Both $p \Rightarrow p$ and $\neg p \Rightarrow \neg p$ are tautologies, so the given proposition $(p \Rightarrow p) \Rightarrow (\neg p \Rightarrow \neg p) \wedge q$ is equivalent to $\text{True} \Rightarrow \text{True} \wedge q$. That proposition is true when q is true, and false when q is false—so the whole expression is logically equivalent to q.

3.59 False; a proposition over a single variable p might be logically equivalent to T or F instead of being logically equivalent to p or $\neg p$. (For example, $p \wedge \neg p$.) However, every proposition φ over p is logically equivalent to one of $\{p, \neg p, \mathrm{T}, \mathrm{F}\}$. To prove this, observe that there are only two truth assignments for φ, where p is true and where p is false. Thus there are only four possible truth tables for φ:

p				
T	T	T	F	F
F	T	F	T	F

These columns are, respectively: T, p, $\neg p$, and F.

3.60

p	q	$p \Rightarrow q$	$\neg q$	$(p \Rightarrow q) \wedge \neg q$	$\neg p$	$(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p$
T	T	T	F	F	F	T
T	F	F	T	F	F	T
F	T	T	F	F	T	T
F	F	T	T	T	T	T

3.61

p	q	$p \vee q$	$p \Rightarrow (p \vee q)$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	T

3.62

p	q	$p \wedge q$	$(p \wedge q) \Rightarrow p$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

3.63

p	q	$\neg p$	$p \vee q$	$(p \vee q) \wedge \neg p$	$(p \vee q) \wedge \neg p \Rightarrow q$
T	T	F	T	F	T
T	F	F	T	F	T
F	T	T	T	T	T
F	F	T	F	F	T

3.64

p	q	$\neg p$	$p \Rightarrow q$	$\neg p \Rightarrow q$	$(p \Rightarrow q) \wedge (\neg p \Rightarrow q)$	$\left[(p \Rightarrow q) \wedge (\neg p \Rightarrow q)\right] \Rightarrow q$
T	T	F	T	T	T	T
T	F	F	F	T	F	T
F	T	T	T	T	T	T
F	F	T	T	F	F	T

3.65

p	q	r	$p \Rightarrow q$	$q \Rightarrow r$	$(p \Rightarrow q) \wedge (q \Rightarrow r)$	$p \Rightarrow r$	$\left[(p \Rightarrow q) \wedge (q \Rightarrow r)\right] \Rightarrow (p \Rightarrow r)$
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	T	F	T	T
T	F	F	F	T	F	F	T
F	T	T	T	T	T	T	T
F	T	F	T	F	F	T	T
F	F	T	T	T	T	T	T
F	F	F	T	T	T	T	T

3.66 Writing φ to denote the given proposition $(p \Rightarrow q) \wedge (p \Rightarrow r) \Leftrightarrow p \Rightarrow q \wedge r$, we have:

p	q	r	$q \wedge r$	$p \Rightarrow q$	$p \Rightarrow r$	$p \Rightarrow q \wedge r$	φ
T	T	T	T	T	T	T	T
T	T	F	F	T	F	F	T
T	F	T	F	F	T	F	T
T	F	F	F	F	F	F	T
F	T	T	T	T	T	T	T
F	T	F	F	T	T	T	T
F	F	T	F	T	T	T	T
F	F	F	F	T	T	T	T

3.67 Writing φ to denote the given proposition $(p \Rightarrow q) \vee (p \Rightarrow r) \Leftrightarrow p \Rightarrow q \vee r$, we have:

p	q	r	$q \vee r$	$p \Rightarrow q$	$p \Rightarrow r$	$p \Rightarrow q \vee r$	φ
T	T	T	T	T	T	T	T
T	T	F	T	T	F	T	T
T	F	T	T	F	T	T	T
T	F	F	F	F	F	F	T
F	T	T	T	T	T	T	T
F	T	F	T	T	T	T	T
F	F	T	T	T	T	T	T
F	F	F	F	T	T	T	T

3.68

p	q	r	$q \vee r$	$p \wedge (q \vee r)$	$p \wedge q$	$p \wedge r$	$(p \wedge q) \vee (p \wedge r)$	$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
T	T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T	T
T	F	T	T	T	F	T	T	T
T	F	F	F	F	F	F	F	T
F	T	T	T	F	F	F	F	T
F	T	F	T	F	F	F	F	T
F	F	T	T	F	F	F	F	T
F	F	F	F	F	F	F	F	T

3.69

p	q	r	$q \Rightarrow r$	$p \Rightarrow (q \Rightarrow r)$	$p \wedge q$	$p \wedge q \Rightarrow r$	$p \Rightarrow (q \Rightarrow r) \Leftrightarrow p \wedge q \Rightarrow r$
T	T	T	T	T	T	T	T
T	T	F	F	F	T	F	T
T	F	T	T	T	F	T	T
T	F	F	T	T	F	T	T
F	T	T	T	T	F	T	T
F	T	F	F	T	F	T	T
F	F	T	T	T	F	T	T
F	F	F	T	T	F	T	T

3.70

p	q	$p \wedge q$	$p \vee (p \wedge q)$
T	T	T	T
T	F	F	T
F	T	F	F
F	F	F	F

Because the p column and $p \vee (p \wedge q)$ column match, the proposition $p \vee (p \wedge q) \Leftrightarrow p$ is a tautology.

3.71

p	q	$p \vee q$	$p \wedge (p \vee q)$
T	T	T	T
T	F	T	T
F	T	T	F
F	F	F	F

Because the p column and $p \wedge (p \vee q)$ column match, the proposition $p \wedge (p \vee q) \Leftrightarrow p$ is a tautology.

3.72

p	q	$p \oplus q$	$p \vee q$	$p \oplus q \Rightarrow p \vee q$
T	T	F	T	T
T	F	T	T	T
F	T	T	T	T
F	F	F	F	T

3.73

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg p \vee \neg q$
T	T	T	F	F	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	T
F	F	F	T	T	T	T

3.74

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

3.75

p	q	r	$q \vee r$	$p \vee (q \vee r)$	$p \vee q$	$(p \vee q) \vee r$
T	T	T	T	T	T	T
T	T	F	T	T	T	T
T	F	T	T	T	T	T
T	F	F	F	T	T	T
F	T	T	T	T	T	T
F	T	F	T	T	T	T
F	F	T	T	T	F	T
F	F	F	F	F	F	F

3.76

p	q	r	$q \wedge r$	$p \wedge (q \wedge r)$	$p \wedge q$	$(p \wedge q) \wedge r$
T	T	T	T	T	T	T
T	T	F	F	F	T	F
T	F	T	F	F	F	F
T	F	F	F	F	F	F
F	T	T	T	F	F	F
F	T	F	F	F	F	F
F	F	T	F	F	F	F
F	F	F	F	F	F	F

3.77

p	q	r	$q \oplus r$	$p \oplus (q \oplus r)$	$p \oplus q$	$(p \oplus q) \oplus r$
T	T	T	F	T	F	T
T	T	F	T	F	F	F
T	F	T	T	F	T	F
T	F	F	F	T	T	T
F	T	T	F	F	T	F
F	T	F	T	T	T	T
F	F	T	T	T	F	T
F	F	F	F	F	F	F

3.78

p	q	r	$q \Leftrightarrow r$	$p \Leftrightarrow (q \Leftrightarrow r)$	$p \Leftrightarrow q$	$(p \Leftrightarrow q) \Leftrightarrow r$
T	T	T	T	T	T	T
T	T	F	F	F	T	F
T	F	T	F	F	F	F
T	F	F	T	T	F	T
F	T	T	T	F	F	F
F	T	F	F	T	F	T
F	F	T	F	T	T	T
F	F	F	T	F	T	F

3.79

p	q	$p \Rightarrow q$	$\neg p$	$\neg p \vee q$
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

3.80

p	q	r	$q \Rightarrow r$	$p \Rightarrow (q \Rightarrow r)$	$p \wedge q$	$p \wedge q \Rightarrow r$
T	T	T	T	T	T	T
T	T	F	F	F	T	F
T	F	T	T	T	F	T
T	F	F	T	T	F	T
F	T	T	T	T	F	T
F	T	F	F	T	F	T
F	F	T	T	T	F	T
F	F	F	T	T	F	T

3.81

p	q	$p \Leftrightarrow q$	$\neg p$	$\neg q$	$\neg p \Leftrightarrow \neg q$
T	T	T	F	F	T
T	F	F	F	T	F
F	T	F	T	F	F
F	F	T	T	T	T

3.82

p	q	$p \Rightarrow q$	$\neg(p \Rightarrow q)$	$\neg q$	$p \wedge \neg q$
T	T	T	F	F	F
T	F	F	T	T	T
F	T	T	F	F	F
F	F	T	F	T	F

3.83 If `p` stands for an expression *that causes an error when it's evaluated,* then the two blocks of code aren't equivalent. The first block causes a crash; the second doesn't.

3.84 $\neg p$

3.85 $p \wedge (q \vee r)$

3.86 False, which can be expressed as $p \wedge \neg p$.

3.87 There are several possibilities: $\neg(p \vee q)$ or $\neg(p \vee r)$ or $\neg(q \vee r)$, or False (which can be expressed as $p \wedge \neg p$).

3.88 Here are the propositions from Figure 4.31 that can be expressed with zero, one, or two $\wedge$, $\vee$, and $\neg$ gates:

0 *gates:* p and q.
1 *gate:* $\neg p$, $\neg q$, $p \wedge q$, and $p \vee q$.
2 *gates:* true ($p \vee \neg p$); false ($p \wedge \neg p$); nand ($\neg(p \wedge q)$), nor ($\neg(p \vee q)$); $p \Rightarrow q$ (as $\neg p \vee q$); $q \Rightarrow p$ (as $\neg q \vee p$); column #12 ($p \wedge \neg q$); and column #14 ($\neg p \wedge q$).

The only two propositions that can't be expressed with two gates are $p \oplus q$ and $p \Leftrightarrow q$.

3.89 Here's a solution based on the solutions to Exercises 3.37 and 3.38:

```
def allFormulae(numGates):
    if numGates == 0:
        return ["p", "q", "r"]
    else:
        subformulae = {}
        for i in range(numGates):
            subformulae[i] = allFormulae(i)
        result = []
        for phi in subformulae[numGates-1]:
            result.append(["not", phi])
        for i in range(numGates):
            for phi in subformulae[i]:
                for psi in subformulae[numGates-i-1]:
                    result.append(["and", phi, psi])
                    result.append(["or", phi, psi])
        return result
```

```
def equivalent(phi, psi):
    for rho in allAssignments(["p", "q", "r"]):
        if eval(phi, rho) != eval(psi, rho):
            return False
    return True
```

```
def uniqueFormulae(formulae):
    result = []
    for phi in formulae:
        unique = True
        for psi in result:
            if equivalent(phi, psi):
                unique = False
                print "rejecting", phi
                break
        if unique:
            result.append(phi)
            print "adding", phi
    return result

candidates = allFormulae(0) + allFormulae(1) \
             + allFormulae(2) + allFormulae(3)
unique = uniqueFormulae(candidates)
```

Using this code, I get 84 different propositions.

3.90 The proposition $p \oplus q \oplus r \oplus s \oplus t$ is true when *an odd number of the Boolean variables* $\{p, q, r, s, t\}$ are true—that is, when $p + q + r + s + t \bmod 2 = 1$.

3.91 Here is the pseudocode:

```
for y = 1 ... height:
  for x = 1 ... width:
    if P[x,y] is more white than black:
      error = "white" - P[x,y]
      P[x,y] = "white"
    else:  # P[x,y] is closer to "black"
      error = "black" - P[x,y]
      P[x,y] = "black"

    if x < width and not (y < height):
      add 7/16 · error to P[x+1,y]    (E)
    if x > 1 and y < height:
      add 1/16 · error to P[x-1,y+1]  (SW)
    if y < height:
      add 5/16 · error to P[x,y+1]    (S)
    if x < width and y < height:
      add 3/16 · error to P[x+1,y+1]  (SE)
```

3.92 $r \vee (p \wedge q)$

3.93 $(q \wedge r) \vee (\neg q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge \neg r)$

3.94 $p \vee q$

3.95 $\neg p \wedge \neg q \wedge \neg r$

3.96 $(p \vee r) \wedge (q \vee r)$

3.97 $p \wedge (\neg q \vee \neg r)$

3.98 $(p \vee \neg q) \wedge (\neg q \vee r)$

3.99 $p \wedge (q \vee r)$

3.100 We can produce a one-clause 3CNF tautology: for example, $p \vee \neg p \vee q$.

3.101 Using De Morgan's Laws, we can interpret a clause $(a \vee b \vee c)$ as $\neg(\neg a \wedge \neg b \wedge \neg c)$. That is, the truth assignment $a = b = c =$ False does not satisfy a proposition $\cdots \wedge (a \vee b \vee c) \wedge \cdots$. However, this particular clause is satisfied by *any* other truth assignment. Thus, we'll need to use a different clause to rule out each of the eight possible truth assignments. So the smallest 3CNF formula that's not satisfiable has three variables and eight clauses, each of which rules out one possible satisfying truth assignment:

$$\begin{aligned} &(p \vee q \vee r) \wedge (p \vee q \vee \neg r) \\ \wedge\ &(p \vee \neg q \vee r) \wedge (p \vee \neg q \vee \neg r) \\ \wedge\ &(\neg p \vee q \vee r) \wedge (\neg p \vee q \vee \neg r) \\ \wedge\ &(\neg p \vee \neg q \vee r) \wedge (\neg p \vee \neg q \vee \neg r). \end{aligned}$$

3.102 One way to work out this quantity is to split up the types of legal clauses based on the number of *unnegated* variables in them. We can have clauses with 0, 1, 2, or 3 unnegated clauses:

3 *unnegated variables:* the only such clause is $p \vee q \vee r$.
2 *unnegated variables:* there are three different choices of which variable fails to appear in unnegated form (p, q, r), and there are three different choices of which variable *does* appear in negated form (again, p, q, r). Thus there are $3 \cdot 3 = 9$ total such clauses.
1 *unnegated variable:* Again, there are three choices for the unnegated variable, and three choices for the omitted negated variable. Again there are $3 \cdot 3 = 9$ total such clauses.
0 *unnegated variables:* the only such clause is $\neg p \vee \neg q \vee \neg r$.

So the total is $1 + 9 + 9 + 1 = 20$ clauses.

3.103 Because a formula in 3CNF is a conjunction of clauses, the entire proposition is a tautology if and only if *every clause is a tautology.* The only tautological clauses are of the form $a \vee \neg a \vee b$, so the largest tautological 3CNF formula has six clauses:

$$\begin{aligned}&(p \vee \neg p \vee q) \wedge (p \vee \neg p \vee r)\\ \wedge\ &(q \vee \neg q \vee p) \wedge (q \vee \neg q \vee r)\\ \wedge\ &(r \vee \neg r \vee p) \wedge (r \vee \neg r \vee q).\end{aligned}$$

3.104 Suppose that φ is satisfied by the all-true truth assignment—that is, when $p = q = r =$ True then φ itself evaluates to true. It's fairly easy to see that if the all-true truth assignment is the *only* truth assignment that satisfies φ then there can be more clauses in φ than if there are many satisfying truth assignments. We cannot have the clause $\neg p \vee \neg q \vee \neg r$ in φ; otherwise, the all-true assignment doesn't satisfy φ. But *any clause that contains at least one unnegated literal can be in φ.* Those clauses are:

1 *clause with* 3 *unnegated variables:* the only such clause is $p \vee q \vee r$.
9 *clauses with* 2 *unnegated variables:* there are three choices of which variable fails to appear in unnegated form, and there are three different choices of which variable *does* appear in negated form.
9 *clauses with* 1 *unnegated variable:* Again, there are three choices for the unnegated variable, and three choices for the omitted negated variable.

So the total is $1 + 9 + 9 = 19$ clauses:

$$\begin{aligned}&(p \vee q \vee r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r)\\ \wedge\ &(\neg p \vee p \vee q) \wedge (\neg p \vee p \vee r) \wedge (\neg q \vee q \vee p) \wedge (\neg q \vee q \vee r)\\ \wedge\ &(\neg r \vee r \vee p) \wedge (\neg r \vee r \vee q) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee q \vee \neg r)\\ \wedge\ &(\neg p \vee \neg q \vee r) \wedge (p \vee \neg p \vee \neg q) \wedge (p \vee \neg p \vee \neg r) \wedge (q \vee \neg q \vee \neg p)\\ \wedge\ &(q \vee \neg q \vee \neg r) \wedge (r \vee \neg r \vee \neg p) \wedge (r \vee \neg r \vee \neg q)\end{aligned}$$

3.105 Tautological DNF formulae correspond to satisfiable CNF formulae. A Boolean formula φ is a tautology if and only if $\neg\varphi$ is not satisfiable, and for an m-clause φ in 3DNF, by an application of De Morgan's Laws we see that $\neg\varphi$ is equivalent to an m-clause 3CNF formula. Thus the smallest 3DNF formula that's a tautology has three variables and eight clauses, each of which "rules in" one satisfying truth assignment.

$$\begin{aligned}&(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r)\\ \vee\ &(\neg p \wedge q \wedge r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r).\end{aligned}$$

3.106 We can give a one-clause non-satisfiable 3DNF formula: $p \wedge \neg p \wedge q$.

Section 3.4: An Introduction to Predicate Logic

3.107 $P(x) = x$ has strong typing and x is object-oriented.

3.108 $P(x) = x$ has strong typing and x is not object-oriented.

3.109 $P(x) = x$ has either scope, or x is both imperative and has strong typing.

3.110 $P(x) = x$ is either object-oriented or scripting.

3.111 $P(x) = x$ does not have strong typing or x is not object-oriented.

3.112 $n \geq 8$

3.113 $\exists i \in \{1, 2, \ldots, n\} : \mathit{isLower}(x_i)$

3.114 $\exists i \in \{1, 2, \ldots, n\} : \neg \mathit{isLower}(x_i) \wedge \neg \mathit{isUpper}(x_i) \wedge \neg \mathit{isDigit}(x_i)$

3.115 If there are no chairs at all in Bananaland, then both "all chairs in Bananaland are green" and "no chairs in Bananaland are green" are true.

3.116 The given predicate is logically equivalent to $[\mathit{age}(x) < 18] \vee [\mathit{gpa}(x) \geq 3.0]$. Here's the relevant truth table, writing p for $[\mathit{age}(x) < 18]$ and q for $[\mathit{gpa}(x) \geq 3.0]$]:

p	q	$\neg p \wedge q$	$p \vee (\neg p \wedge q)$	$p \vee q$
T	T	F	T	T
F	T	T	T	T
T	F	F	T	T
F	F	F	F	F

3.117 The given predicate is logically equivalent to $\neg[\mathit{cs}(x)]$. Here's the relevant truth table, writing p for $\mathit{cs}(x)$ and q for $\mathit{hawaii}(x)$:

p	q	$q \wedge p$	$q \Rightarrow q \wedge p$	$\neg(q \Rightarrow q \wedge p)$	$p \Rightarrow \neg(q \Rightarrow (q \wedge p))$	$\neg p$
T	T	T	T	F	F	F
T	F	F	T	F	F	F
F	T	F	F	T	T	T
F	F	F	T	F	T	T

3.118 The given predicate is logically equivalent to $\mathit{hasMajor}(x) \wedge (\neg \mathit{junior}(x) \vee \neg \mathit{oncampus}(x))$. Here's the relevant truth table, writing p for $\mathit{hasMajor}(x)$ and q for $\mathit{junior}(x)$ and r for $\mathit{oncampus}(x)$:

p	q	r	$p \wedge \neg q \wedge r$	$p \wedge \neg q \wedge \neg r$	$p \wedge q \wedge \neg r$	$(p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge \neg r)$	$p \wedge (\neg q \vee \neg r)$
T	T	T	F	F	F	F	F
T	T	F	F	F	T	T	T
T	F	T	T	F	F	T	T
T	F	F	F	T	F	T	T
F	T	T	F	F	F	F	F
F	T	F	F	F	F	F	F
F	F	T	F	F	F	F	F
F	F	F	F	F	F	F	F

3.119 The claim is false. Let's think of this problem as a question about a proposition involving $\{\text{True}, \text{False}, \wedge, \vee, \neg, \Rightarrow\}$ and—once each—p and q. It's hopeless to express $p \Leftrightarrow q$ with this kind of proposition. Any such proposition must involve a subproposition of the form $\varphi \diamond \psi$, where φ uses p and ψ uses q. Unfortunately, φ can only be logically equivalent to one of four functions: $\text{True}, \text{False}, p, \neg p$. And ψ can only be logically equivalent to one of four functions: $\text{True}, \text{False}, q, \neg q$.

But $\varphi \diamond \psi$ therefore can only compute the following functions: $\text{True}, \text{False}, p, \neg p, q, \neg q, p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q, p \Rightarrow q, \neg p \Rightarrow q, p \Rightarrow \neg q, \neg p \Rightarrow \neg q, p \Leftarrow q, \neg p \Leftarrow q, p \Leftarrow \neg q, \neg p \Leftarrow \neg q$. Alas, though, each of the implications can be expressed as a disjunction, so the unique outputs are these:

$$\text{True}, \text{False}, p, \neg p, q, \neg q, p \wedge q, \neg p \wedge q, p \wedge \neg q,$$
$$\neg p \wedge \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \text{ and } \neg p \vee \neg q.$$

And there are only 14 entries here. That leaves two unimplementable functions. They are, intuitively enough, $\Leftrightarrow$ and $\oplus$.

3.120 "I am ambivalent about writing my program in Java."

3.121 "I was excited to learn about a study-abroad computer science program on the island of Java."

3.122 "I love to start my day with a cup o' Java."

3.123 We can prove that this statement is false by counterexample: the number $n = 2$ is prime, but $\frac{n}{2} = 1$ *is* in $\mathbb{Z}$.

3.124 $\exists x : \neg[Q(x) \Rightarrow P(x)]$, which is logically equivalent to $\exists x : Q(x) \wedge \neg P(x)$.

3.125 *There exists a negative entry in the array A.*

3.126 The "or" of the original statement might be inclusive or exclusive, though the much more natural reading is the former. Negating that version yields: *There exists a decent programming language that denotes block structure with neither parentheses nor braces.*

3.127 *No odd number is evenly divisible by any other odd number.*

3.128 This sentence is ambiguous: the type of quantification expressed by "a lake" could be universal or existential. Thus there are two reasonable readings of the original sentence: denoting by L the set of lakes, these readings are

$$\exists x \in \mathsf{MN} : \exists \ell \in L : d(x, \ell) \geq 10$$
$$\exists x \in \mathsf{MN} : \forall \ell \in L : d(x, \ell) \geq 10.$$

The second reading seems more natural, so: *For every point in Minnesota, there is some lake within ten miles of that point.*

3.129 This sentence is ambiguous; the order of quantification is unclear. There are two natural readings:

$$\forall x \in \mathcal{A} : \exists y \in I_n : x(y) \text{ takes } \geq n \log n \text{ steps}$$
$$\text{and } \exists y \in I_n : \forall x \in \mathcal{A} : x(y) \text{ takes } \geq n \log n \text{ steps}$$

where we've written $\mathcal{A}$ to denote the set of sorting algorithms and I_n to denote the set of n-element arrays. (There's also some ambiguity in the meaning of n, which may be implicitly quantified or may have a particular numerical value from context. We'll leave this ambiguity alone.)

The first reading, basically: after I see your code for sorting, it's possible for me to construct an array that make your particular sorting algorithm slow. The second reading, basically: there's a single devilish array that makes every sorting algorithm slow.

Negating the first reading: *There's a sorting algorithm that, for every n-element array, takes fewer than n* $\log$ *n steps on that array.* Negating the second reading: *For every n-element array, there's some sorting algorithm that takes fewer than n* $\log$ *n steps on that array.*

3.130 Assume the antecedent—that is, assume that there exists a particular value $x^* \in S$ such that $P(x^*) \wedge Q(x^*)$. Then