appendix

F

Answers to Questions & Exercises

Chapter 1

Section 1.1

1. One and only one of the upper two inputs must be 1, and the lowest input must be 1.
2. The 1 on the lower input is negated to 0 by the NOT gate, causing the output of the AND gate to become 0. Thus both inputs to the OR gate are 0 (remember that the upper input to the flip-flop is held at 0) so the output of the OR gate becomes 0. This means that the output of the AND gate will remain 0 after the lower input to the flip-flop returns to 0.
3. The output of the upper OR gate will become 1, causing the upper NOT gate to produce an output of 0. This will cause the lower OR gate to produce a 0, causing the lower NOT gate to produce a 1. This 1 is seen as the output of the flip-flop as well as being fed back to the upper OR gate, where it holds the output of that gate at 1, even after the flip-flop's input has returned to 0.
4. a. The entire circuit is equivalent to a single OR gate.
   b. This entire circuit is also equivalent to a single XOR gate.
5. a. 6AF2 b. E85517 c. 48
6. a. 01011111110110010111
   b. 0110000100001010
   c. 101010 1111001101
   d. 0000000100000000

Section 1.2

1. In the first case, memory cell number 6 ends up containing the value 5. In the second case, it ends up with the value 8.
2. Step 1 erases the original value in cell number 3 when the new value is written there. Consequently, Step 2 does not place the original value from cell number 3 in cell number 2. The result is that both cells end up with

the value that was originally in cell number 2. A correct procedure is the following:

Step 1. Move the contents of cell number 2 to cell number 1.

Step 2. Move the contents of cell number 3 to cell number 2.

Step 3. Move the contents of cell number 1 to cell number 3.

3. 32768 bits

Section 1.3

1. Faster retrieval of data and higher transfer rates
2. The point to remember here is that the slowness of mechanical motion compared with the speed of the internal functioning of the computer dictates that we minimize the number of times we must move the read/write heads. If we fill a complete surface before starting the next, we must move the read/write head each time we finish with a track. The number of moves therefore is approximately the same as the total number of tracks on the two surfaces. If, however, we alternate between surfaces by electronically switching between the read/write heads, we must move the read/write heads only after each cylinder has been filled.
3. In this application, information must be retrieved from mass storage in a random manner, which would be time consuming in the context of the spiral system used on CDs and DVDs. (Moreover, current technology does not allow individual portions of data to be updated on a CD or DVD.)
4. CD, DVD, and Blu-ray disks are all the same physical size and use the same spiral track layout for placing data on the platter. A player equipped with multiple lasers is able to read all three types of optical disk.
5. Flash drives do not require physical motion so they have shorter response times and do not suffer from physical wear.
6. Magnetic hard disks are faster and have higher capacities than other forms of magnetic media, such as floppy disks and magnetic tape. Speed, capacity, and rewriting ability compare favorably to optical media. Price per units of storage on magnetic disks continues to be lower than solid-state disks, although this advantage has steadily eroded in recent years.

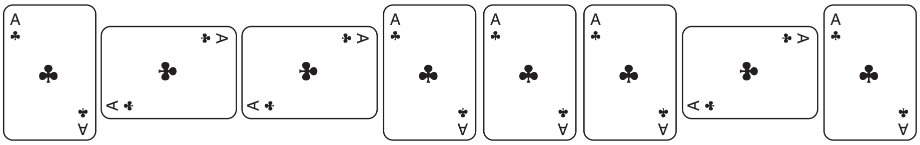
Section 1.4

1. Computer Science
2. The two patterns are the same, except that the sixth bit from the low-order end is always 0 for uppercase and 1 for lowercase.
3. a.

00100010	01010011	01110100	01101111
01110000	00100001	00100010	00100000
01000011	01101000	01100101	01110010
01111001	01101100	00100000	01110011
01101000	01101111	01110101	01110100
01100101	01110100	00101110	

b.

01000100	01101111	01100101	01110011
00100000	00110010	00100000	00101011
00100000	00110011	00100000	00111101
00100000	00110101	00111111	

4. 

5. a. 5 b. 9 c. 11 d. 6 e. 16 f. 18

6. a. 110 b. 1101 c. 1011 d. 10010 e. 11011 f. 100

7. In 24 bits, we can store three symbols using ASCII. Thus we can store values as large as 999. However, if we use the bits as binary digits, we can store values up to 16,777,215.

8. a. 15.15 b. 51.0.128 c. 10.160

9. Geometric representations are more conducive to changes in scale than images encoded as bit maps. However, geometric representations do not typically provide the same photographic quality that bit maps produce. Indeed, as discussed in Section 1.9, JPEG representations of bit maps are very popular in photography.

10. With a sample rate of 44,100 samples per second, one hour of stereo music would require 635,040,000 bytes of storage. Thus, it would just about fill a CD whose capacity is slightly more than 600MB.

Section 1.5

1. a. 42 b. 33 c. 23 d. 6 e. 31

2. a. 100000 b. 1000000 c. 1100000 d. 1111 e. 11011

3. a. $3\frac{1}{4}$ b. $5\frac{7}{8}$ c. $2\frac{1}{2}$ d. $6\frac{3}{8}$ e. $\frac{5}{8}$

4. a. 100.1 b. 10.11 c. 1.001 d. 0.0101 e. 101.101

5. a. 100111 b. 1011.110 c. 100000 d. 1000.00

Section 1.6

1. a. 3 b. 15 c. −4 d. −6 e. 0 f. −16

2. a. 00000110 b. 11111010 c. 11101111
   d. 00001101 e. 11111111 f. 00000000

3. a. 11111111 b. 10101011 c. 00000100
   d. 00000010 e. 00000000 f. 10000001

4. a. With 4 bits the largest value is 7 and the smallest is −8.
   b. With 6 bits the largest value is 31 and the smallest is −32.
   c. With 8 bits the largest value is 127 and the smallest is −128.

5. a. 0111 $(5 + 2 = 7)$ b. 0100 $(3 + 1 = 4)$ c. 1111 $(5 + (-6) = -1)$
   d. 0001 $(-2 + 3 = 1)$ e. 1000 $(-6 + (-2) = -8)$

6. a. 0111 b. 1011 (overflow) c. 0100 (overflow)
d. 0001 e. 1000 (overflow)

7.

a.	b.	c.	d.	e.
0110	0011	0100	0010	0001
+ 0001	+ 1110	+ 1010	+ 0100	+1011
0111	0001	1110	0110	1100

8. No. Overflow occurs when an attempt is made to store a number that is too large for the system being used. When adding a positive value to a negative value, the result must be between the values being added. Therefore the result will be small enough to be stored without error.

9. a. 6 because 1110 → 14 − 8
b. −1 because 0111 → 7 − 8
c. 0 because 1000 → 8 − 8
d. −6 because 0010 → 2 − 8
e. −8 because 0000 → 0 − 8
f. 1 because 1001 → 9 − 8

10. a. 1101 because 5 + 8 = 13 → 1101
b. 0011 because −5 + 8 = 3 → 0011
c. 1011 because 3 + 8 = 11 → 1011
d. 1000 because 0 + 8 = 8 → 1000
e. 1111 because 7 + 8 = 15 → 1111
f. 0000 because −8 + 8 = 0 → 0000

11. No. The largest value that can be stored in excess eight notation is 7, represented by 1111. To represent a larger value, at least excess 16 (which uses patterns of 5 bits) must be used. Similarly, 6 cannot be represented in excess four notation. (The largest value that can be represented in excess four notation is 3.)

Section 1.7

1. a. $\frac{5}{8}$ b. $3\frac{1}{4}$ c. $\frac{9}{32}$ d. $-1\frac{1}{2}$ e. $-(\frac{11}{64})$

2. a. 01101011 b. 01111010 (truncation error)
c. 01001100 d. 11101110 e. 11111000 (truncation error)

3. 01001001 ($\frac{9}{16}$) is larger than 00111101 ($\frac{13}{32}$). The following is a simple way of determining which of two patterns represents the larger value:

Case 1. If the sign bits are different, the larger is the one with 0 sign bit.

Case 2. If the sign bits are both 0, scan the remaining portions of the patterns from left to right until a bit position is found where the two patterns differ. The pattern containing the 1 in this position represents the larger value.

Case 3. If the sign bits are both 1, scan the remaining portions of the patterns from left to right until a bit position is found where the two patterns differ. The pattern containing the 0 in this position represents the larger value. The simplicity of this comparison process is one of the reasons for representing the exponent in floating-point systems with an excess notation rather than with two's complement.

4. The largest value would be $7\frac{1}{2}$, which is represented by the pattern 01111111. As for the smallest positive value, you could argue that there are two "correct" answers. First, if you stick to the coding process described in the text, which requires the most significant bit of the mantissa to be 1 (called normalized form), the answer is $\frac{1}{32}$, which is represented by the pattern 00001000. However, most machines do not impose this restriction for values close to 0. For such a machine, the correct answer is $\frac{1}{256}$ represented by 00000001.

Section 1.8

1. Python is considered an *interpreted language* because users can type program fragments interactively at a prompt, rather than having to save the script, invoke a compiler, and then execute the program.
2. a. `print('Computer Science Rocks' + '!')`
   b. `print(42)`
   c. `print(3.1416)`
3. a. `rockstar = 'programmer'`
   b. `seconds_per_hour = 60 * 60`
   or
   `seconds_per_hour = 3600`
   c. `bodyTemp = 98.6`
4. `metricBodyTemp = (bodyTemp - 32)/1.8`

Section 1.9

1. Run-length encoding, frequency-dependent encoding, relative encoding, and dictionary encoding
2. 121321112343535
3. Color cartoons consist of blocks of solid color with sharp edges. Moreover, the number of colors involved is limited.
4. No. Both GIF and JPEG are lossy compression systems, meaning that details in the image will be lost.
5. JPEG's baseline standard takes advantage of the fact that the human eye is not as sensitive to changes in color as it is to changes in brightness. Thus it reduces the number of bits used to represent color information without noticeable loss in image quality.
6. Temporal masking and frequency masking
7. When encoding information, approximations are made. In the case of numeric data, these approximations are compounded when computations are performed, which can lead to erroneous results. Approximations are not as critical in the cases of images and sound because the encoded data are normally only stored, transferred, and reproduced. If, however, images or sound were repeatedly reproduced, rerecorded, and then reencoded, these approximations could compound and ultimately lead to worthless data.

Section 1.10

1. b, c, and e
2. Yes. If an even number of errors occurs in one byte, the parity technique does not detect them.
3. In this case, errors occur in bytes a and d of Question 1. The answer to Question 2 remains the same.
4.

a.	100100010	101010011	101110100
	101101111	001110000	100100001
	100100010	000100000	001000011
	001101000	101100101	101110010
	001111001	101101100	000100000
	001110011	001101000	101101111
	001110101	101110100	101100101
	001100100	100101110	
b.	101000100	101101111	101100101
	001110011	100100000	000110010
	000100000	100101011	000100000
	100110011	000100000	000111101
	000100000	100110101	100111111

5. a. BED   b. CAB   c. HEAD
6. One solution is the following:
   A 00000
   B 11100
   C 01111
   D 10011

Chapter 2

Section 2.1

1. On some machines this is a two-step process consisting of first reading the contents from the first cell into a register and then writing it from the register into the destination cell. On most machines, this is accomplished as one activity without using an intermediate register.
2. The value to be written, the address of the cell in which to write, and the command to write
3. General-purpose registers are used to hold the data immediately applicable to the operation at hand; main memory is used to hold data that will be needed in the near future; and mass storage is used to hold data that will likely not be needed in the near future.

Section 2.2

1. The term *move* often carries the connotation of removing from one location and placing in another, thus leaving a hole behind. In most cases within a machine, this removal does not take place. Rather, the object being moved is most often copied (or cloned) into the new location.

2. A common technique, called relative addressing, is to state how far rather than where to jump. For example, an instruction might be to jump forward three instructions or jump backward two instructions. You should note, however, that such statements must be altered if additional instructions are later inserted between the origin and the destination of the jump.

3. This could be argued either way. The instruction is stated in the form of a conditional jump. However, because the condition that 0 be equal to 0 is always satisfied, the jump will always be made as if there were no condition stated at all. You will often find machines with such instructions in their repertoires because they provide an efficient design. For example, if a machine is designed to execute an instruction with a structure such as "If . . . jump to. . . " this instruction form can be used to express both conditional and unconditional jumps.

4. 156C = 0001010101101100
   166D = 0001011001101101
   5056 = 0101000001010110
   306E = 0011000001101110
   C000 = 1100000000000000

5. a. STORE the contents of register 6 in memory cell number 8A.
   b. JUMP to location DE if the contents of register A equals that of register 0.
   c. AND the contents of registers 3 and C, leaving the result in register 0.
   d. MOVE the contents of register F to register 4.

6. The instruction 15AB requires that the CPU query the memory circuitry for the contents of the memory cell at address AB. This value, when obtained from memory, is then placed in register 5. The instruction 25AB does not require such a request of memory. Rather, the value AB is placed in register 5.

7. a. 2356 b. A503 c. 80A5

Section 2.3

1. Hexadecimal 34

2. a. 0F b. C3

3. a. 00 b. 01 c. four times

4. It halts. This is an example of what is often called self-modifying code. That is, the program modifies itself. Note that the first two instructions place hexadecimal C0 at memory location F8, and the next two instructions place 00 at location F9. Thus, by the time the machine reaches the instruction at F8, the halt instruction (C000) has been placed there.

Section 2.4

1. a. 00001011 b. 10000000 c. 00101101
   d. 11101011 e. 11101111 f. 11111111
   g. 11100000 h. 01101111 i. 11010010

2. 00111100 with the AND operation

3. 00111100 with the XOR operation

4. a. The final result is 0 if the string contained an even number of 1s. Otherwise it is 1.
   b. The result is the value of the parity bit for even parity.
5. The logical XOR operation mirrors addition except for the case where both operands are 1, in which case the XOR produces a 0, whereas the sum is 10. (Thus the XOR operation can be considered an addition operation with no carry.)
6. Use AND with the mask 11011111 to change lowercase to uppercase. Use OR with 00100000 to change uppercase to lowercase.
7. a. 01001101 b. 11100001 c. 11101111
8. a. 57 b. B8 c. 6F d. 6A
9. 5
10. 00110110 in two's complement; 01011110 in floating-point. The point here is that the procedure used to add the values is different depending on the interpretation given the bit patterns.
11. One solution is as follows:

    12A7 (LOAD register 2 with the contents of memory cell A7.)
    2380 (LOAD register 3 with the value 80.)
    7023 (OR registers 2 and 3 leaving the result in register 0.)
    30A7 (STORE contents of register 0 in memory cell A7.)
    C000 (HALT.)
12. One solution is as follows:

    15E0 (LOAD register 5 with the contents of memory cell E0.)
    A502 (ROTATE the contents of register 5 to the left by 2 bits.)
    260F (LOAD register 6 with the value 0F.)
    8056 (AND registers 5 and 6, leaving the result in register 0.)
    30E1 (STORE the contents of register 0 in memory cell E1.)
    C000 (HALT.)

Section 2.5

1. a. 37B5
   b. One million times
   c. No. A typical page of text contains less than 4000 characters. Thus the ability to print five pages in a minute indicates a printing rate of no more than 20,000 characters per minute, which is much less than one million characters per second. (The point is that a computer can send characters to a printer much faster than the printer can print them; thus the printer needs a way of telling the computer to wait.)
2. The disk will make 50 revolutions in one second, meaning that 800 sectors will pass under the read/write head in a second. Because each sector contains 1024 bytes, bits will pass under the read/write head at approximately 6.5 Mbps. Thus communication between the controller and the disk drive will have to be at least this fast if the controller is going to keep up with the data being read from the disk.

3. A 300-page novel represented in Unicode consists of about 2MB or 16,000,000 bits. Thus approximately 0.3 seconds would be required to transfer the entire novel at 54 Mbps.

Section 2.6

1. The `int()` function calls used when inputting the side lengths will truncate any floating-point values entered to integer values. However, the `math.sqrt()` function returns a floating-point value, regardless of whether its parameters were integers or floats. To output an integer instead, the final line of the script can be replaced with

   ```
   print(int(hypotenuse))
   ```

   or the assignment statement for hypotenuse can be replaced with

   ```
   hypotenuse = int(math.sqrt(sideA**2 + sideB**2))
   ```

2. Replacing both `int()` calls in the original script with `float()` produces a script that works with floating-point values.
3. One example of Python code that would produce cleaner output is:

   ```
   print('Your speed is ' + str(mph) + ' mph')
   . . .
   print('Your total elapsed time is ' + str(elapsed_minutes) +
         ' minutes, ' + str(elapsed_seconds) + ' seconds')
   ```

4. One example would be:

   ```
   number = int(input('Enter a base-10 number: '))
   print('Binary representation is: ' + str(bin(number)))
   ```

5. One example would be:

   ```
   number = int(input('Enter a number to encrypt or decrypt: '))
   number = number ^ 0x55555555
   print('Result is: ' + str(number))
   ```

6. Unexpected inputs in simple scripts such as these will generally cause errors or unexpected behavior. More complex scripts would check the input for suitability before trying to convert it to an integer, or would include conditionals to check for negative side lengths, etc.

Section 2.7

1. The pipe would contain the instructions B1B0 (being executed), 5002, and perhaps even B0AA. If the value in register 1 is equal to the value in register 0, the jump to location B0 is executed, and the effort already expended on the instructions in the pipe is wasted. On the other hand, no time is wasted because the effort expended on these instructions did not require extra time.
2. If no precautions are taken, the information at memory locations F8 and F9 is fetched as an instruction before the previous part of the program has had a chance to modify these cells.

3. a. The CPU that is trying to add 1 to the cell can first read the value in the cell. Following this the other CPU reads the cell's value. (Note that at this point both CPUs have retrieved the same value.) If the first CPU now finishes its addition and writes its result back in the cell before the second finishes its subtraction and writes its result, the final value in the cell reflects only the activity of the second CPU.
   b. The CPUs might read the data from the cell as before, but this time the second CPU might write its result before the first. Thus only the activity of the first CPU is reflected in the cell's final value.

Chapter 3

Section 3.1

1. A traditional example is the line of people waiting to buy tickets to an event. In this case there might be someone who tries to "break in line," which would violate the FIFO structure.
2. Options (b), (c), and (e)
3. Embedded systems often focus on dedicated tasks, whereas PCs are general-purpose computers. Embedded systems frequently have more limited resources than PCs of comparable age, but may face strict deadlines with minimal human intervention.
4. Time-sharing refers to more than one user accessing a machine at the same time. Multitasking refers to a user performing more than one task at the same time.

Section 3.2

1. *Shell:* Communicates with the machine's environment.

   File manager: Coordinates the use of the machine's mass storage.

   Device drivers: Handle communication with the machine's peripheral devices.

   Memory manager: Coordinates the use of the machine's main memory.

   Scheduler: Coordinates the processes in the system.

   Dispatcher: Controls the assignment of processes to CPU time.
2. The line is vague, and the distinction is often in the eye of the beholder. Roughly speaking, utility software performs basic, universal tasks, whereas application software performs tasks unique to the machine's application.
3. Virtual memory is the imaginary memory space whose apparent presence is created by the process of swapping data and programs back and forth between main memory and mass storage.
4. When the machine is turned on, the CPU begins executing the bootstrap, which resides in ROM. This bootstrap directs the CPU through the process of transferring the operating system from mass storage into the volatile area of main memory. When this transfer is complete, the bootstrap directs the CPU to jump to the operating system.

Section 3.3

1. A program is a set of directions. A process is the action of following those directions.
2. The CPU completes its current machine cycle, saves the state of the current process, and sets its program counter to a predetermined value (which is the location of the interrupt handler). Thus the next instruction executed will be the first instruction within the interrupt handler.
3. They could be given higher priorities so that they would be given preference by the dispatcher. Another option would be to give the higher-priority processes longer time slices.
4. If each process consumed its entire time slice, the machine could provide a complete slice to almost 20 processes in one second. If processes did not consume their entire time slices, this value could be much higher, but then the time required to perform a context switch might become more significant (see Question 5).
5. A total of $^{5000}/_{5001}$ of the machine's time would be spent actually performing processes. However, when a process requests an I/O activity, its time slice is terminated while the controller performs the request. Thus if each process made such a request after only one microsecond of its time slice, the efficiency of the machine would drop to 1/2. That is, the machine would spend as much time performing context switches as it would executing processes.

Section 3.4

1. This system guarantees that the resource is not used by more than one process at a time; however, it dictates that the resource be allocated in a strictly alternating fashion. Once a process has used and relinquished the resource, it must wait for the other process to use the resource before the original process can access it again. This is true even if the first process needs the resource right away and the other process will not need it for some time.
2. If two cars enter opposite ends of the tunnel at the same time, they will not be aware of the other's presence. The process of entering and turning on the lights is another example of a critical region, or in this case we might call it a critical process. In this terminology, we could summarize the flaw by saying that cars at opposite ends of the tunnel could execute the critical process at the same time.
3. a. This guarantees that the nonsharable resource is not required and allocated on a partial basis; that is, a car is given the whole bridge or nothing at all.
   b. This means that the nonsharable resource can be forcibly retrieved.
   c. This makes the nonsharable resource shareable, which removes the competition.
4. A sequence of arrows that forms a closed loop in the directed graph. It is on this observation that techniques have been developed that allow some operating systems to recognize the existence of deadlock and consequently to take appropriate corrective action.

Section 3.5

1. Names and dates are considered poor candidates because they are common choices and therefore represent easy targets for password guessers. The use of complete words is also considered poor because password guessers can easily write a program to try the words found in a dictionary. Moreover, passwords containing only characters are discouraged because they are formed from a limited character set.
2. Four is the number of different bit patterns that can be formed using 2 bits. If more privilege levels were required, the designers would need at least 3 bits to represent the different levels and would therefore probably choose to use a total of 8 levels. In the same manner, the natural choice for fewer than 4 privilege levels would be 2, which is the number of patterns that can be represented with 1 bit.
3. The process could alter the operating system program so that the dispatcher gave every time slice to that process.

Chapter 4

Section 4.1

1. An open network is one whose specifications and protocols are public, allowing different vendors to produce compatible products.
2. Both connect two buses to form a larger bus network. However, a bridge forwards only those messages destined for the other side of the bridge, whereas a switch has multiple connections that each may act as a bridge.
3. A router is a device that directs messages between networks in an internet.
4. How about a mail-order business and its clients, a bank teller and the bank's customers, or a pharmacist and his or her customers?
5. There are numerous protocols involved in traffic flow, verbal telephone communication, and etiquette.
6. Cluster computing typically involves multiple, dedicated computers to provide high-availability or load-balanced distributed computing. Grid computing is more loosely coupled than cluster computing, and could involve machines that join the distributed computation when they are otherwise idle.

Section 4.2

1. Tier-1 and tier-2 ISPs provide the Internet's communication "core," whereas access ISPs provide access to that core to their customers.
2. The DNS (Domain Name System) is the Internet-wide collection of name servers that allow translation from mnemonic addresses to IP addresses (and in the other direction as well).
3. The expression 3.6.9 represents the three-byte pattern 000000110000011000001001. The bit pattern 0001010100011100 would be represented as 21.28 in dotted decimal notation.

4. There could be several answers to this. One is that both progress from the specific to the general. Internet addresses in mnemonic form begin with the name of a particular machine and progress to the name of the TLD. Postal addresses begin with the name of an individual and progress to increasingly larger regions such as city, state, and country. This order is reversed in IP addresses, which start with the bit pattern identifying the domain.
5. Name servers help translate mnemonic addresses into IP addresses. Mail servers send, receive, and store email messages. FTP servers provide file transfer service.
6. Protocols can describe the format of messages that are exchanged, the proper ordering of messages in an exchange, and the meaning of messages.
7. They relieve the initial server from the burden of sending individual messages to each client. The P2P approach shifts this burden to the clients (peers) themselves, whereas multicast shifts this burden to the Internet routers.
8. Criteria to consider may include cost, portability, the practicality of using your computer as your phone, the need to preserve any existing analog phones, emergency 911 service, and the reliability and service areas of the various providers involved.

Section 4.3

1. A URL is essentially the address of a document in the World Wide Web. A browser is a program that assists a user in accessing hypertext.
2. A markup language is a system for inserting explanatory information in a document.
3. HTML is a particular markup language. XML is a standard for producing markup languages.
4. a. <html> marks the beginning of an HTML document.
   b. <head> marks the beginning of a document's head.
   c. </p> marks the end of a paragraph.
   d. marks the end of an item that is linked to another document.
5. Client side and server side are terms used to identify whether an activity is performed at the client's computer or the server's computer.

Section 4.4

1. The link layer receives the message and hands it to the network layer. The network layer determines the direction in which the message should be forwarded and gives the message back to the link layer to be forwarded. The higher layers are not required for routing, although advanced routers may use the transport or application layers to provide additional services such as selective filtering or tiered quality of service.
2. Unlike TCP, UDP is a connectionless protocol that does not confirm that the message was received at the destination.

3. The transport layer uses transport protocol port numbers to determine which unit within the application layer should receive an incoming message.
4. Nothing really. A programmer at any host could modify the software at that host to keep such records. This is why sensitive data should be encrypted.

Section 4.5

1. Phishing is a technique for obtaining sensitive information by asking users for their passwords, credit card numbers, etc., through email while masquerading as a legitimate entity such as the user's bank or the campus IT department. Computers are not secured against phishing; users must rely on sound judgment when revealing sensitive data to others without proper verification.
2. A region's gateway is a router that merely forwards packets (parts of messages) as they pass through. Thus a firewall at the gateway cannot filter traffic by its content but merely by its address information.
3. The use of passwords protects data (and therefore information as well). The use of encryption protects information.
4. In the case of a public-key encryption system, knowing how messages are encrypted does not allow messages to be decrypted.
5. The problems are international in nature and therefore not subject to the laws of a single government. Moreover, legal remedies merely provide recourse to injured parties rather than preventing the injuries.

Chapter 5

Section 5.1

1. A process is the activity of executing an algorithm. A program is a representation of an algorithm.
2. In the introductory chapter we cited algorithms for playing music, operating washing machines, constructing models, and performing magic tricks, as well as the Euclidean algorithm. Many of the "algorithms" you meet in everyday life fail to be algorithms according to our formal definition. The example of the long-division algorithm was cited in the text. Another is the algorithm executed by a clock that continues to advance its hands and ring its chimes day after day.
3. The informal definition fails to require that the steps be ordered and unambiguous. It merely hints at the requirements that the steps be executable and lead to an end.
4. There are two points here. The first is that the instructions define a nonterminating process. In reality, however, the process will ultimately reach the state in which there are no coins in your pocket. In fact, this might be the starting state. At this point the problem is that of ambiguity. The algorithm, as represented, does not tell us what to do in this situation.

Section 5.2

1. One example is found in the composition of matter. At one level, the primitives are considered molecules, yet these particles are actually composites made up of atoms, which in turn are composed of electrons, protons, and neutrons. Today, we know that even these "primitives" are composites.
2. Once a function is correctly constructed, it can be used as a building block for larger program structures without reconsidering the function's internal composition.
3.
```
X = the larger input
Y = the smaller input
while (Y not zero):
  Remainder = remainder after dividing X by Y
  X = Y
  Y = Remainder
GCD = X
```
4. All other colors of light can be produced by combining red, blue, and green. Thus a television picture tube is designed to produce these three basic colors.

Section 5.3

1. a.
```
if (n == 1 or n == 2):
    the answer is the list containing the single value n
else:
    Divide n by 3, obtaining a quotient q and a remainder r
    if (r == 0):
      the answer is the list containing q 3s
    if (r == 1):
      the answer is the list containing (q – 1) 3s and two 2s
    if (r == 2):
      the answer is the list containing q 3s and one 2
```
   b. The result would be the list containing 667 threes.
   c. You probably experimented with small input values until you began to see a pattern.
2. a. Yes. Hint: Place the first tile in the center so that it avoids the quadrant containing the hole while covering one square from each of the other quadrants. Each quadrant then represents a smaller version of the original problem.
   b. The board with a single hole contains $2^{2n} - 1$ squares, and each tile covers exactly three squares.
   c. Parts (a) and (b) of this question provide an excellent example of how knowing a solution to one problem helps solve another. See Polya's fourth phase.
3. It says, "This is the correct answer."
4. Simply trying to assemble the pieces would be a bottom-up approach. However, by looking at the puzzle box to see what the picture is supposed to look like adds a top-down component to your approach.

Section 5.4

1. Change the test in the `while` statement to read "target value != current entry and there remain entries to be considered."

2.
```
Z = 0
X = 1
repeat:
  Z = Z + X
  X = X + 1
  until (X == 6)
```

3. This has proven to be a problem with the C language. When the `do` and `while` key words are separated by several lines, readers of a program often stumble over the proper interpretation of a `while` clause. In particular, the `while` at the end of a `do` statement is often interpreted as the beginning of a `while` statement. Thus experience would say that it is better to use different key words to represent pretest and posttest loop structures.

4.

Cheryl	Alice	Alice
Gene	Cheryl	Brenda
Alice	Gene	Cheryl
Brenda	Brenda	Gene

5. It is a waste of time to insist on placing the pivot above an identical entry in the list. For instance, make the proposed change and then try the new program on a list in which all entries are the same.

6.
```
def sort (List):
  N = 1
  while (N is less than the length of List):
    J = N + 1
    while (J is not greater than length of List):
      if (the entry in position J is less than the entry in
          position N):
          interchange the two entries
      J = J + 1
    N = N + 1
```

7. The following is an inefficient solution. Can you make it more efficient?

```
def sort (List):
  N = the length of List
  while (N > 1):
    J = the length of List
    while (J > 1):
      if (the entry in position J < the entry in
          position J - 1):
          interchange the two entries
      J = J - 1
    N = N - 1
```

Section 5.5

1. The first name considered would be Henry, the next would be Larry, and the last would be Joe.
2. 8, 17
3. 1, 2, 3, 3, 2, 1
4. The termination condition is "N is bigger than or equal to 3" (or "N is not less than 3"). This is the condition under which no additional activations are created.

Section 5.6

1. If the machine can sort 100 names in one second, it can perform $\frac{1}{4}(10{,}000 - 100)$ comparisons in one second. This means that each comparison takes approximately 0.0004 second. Consequently, sorting 1000 names [which requires an average of $\frac{1}{4}(1{,}000{,}000 - 1000)$ comparisons] requires roughly 100 seconds or $1\frac{2}{3}$ minutes.
2. The binary search belongs to $\Theta(\log_2 n)$, the sequential search belongs to $\Theta(n)$, and the insertion sort belongs to $\Theta(n^2)$.
3. The class $\Theta(\log_2 n)$ is most efficient, followed by $\Theta(n)$, $\Theta(n^2)$, and $\Theta(n^3)$.
4. No. The answer is not correct, although it might sound right. The truth is that two of the three cards are the same on both sides. Thus the probability of picking such a card is two-thirds.
5. No. If the dividend is less than the divisor, such as in $\frac{3}{7}$, the answer given is 1, although it should be 0.
6. No. If the value of X is zero and the value of Y is nonzero, the answer given will not be correct.
7. Each time the test for termination is conducted, the statement "Sum $= 1 + 2 + \ \ldots\ + K$ and K less than or equal to N" is true. Combining this with the termination condition "K greater than or equal to N" produces the desired conclusion "Sum $= 1 + 2 + \ \ldots\ + N$." Because K is initialized at zero and incremented by one each time through the loop, its value must ultimately reach that of N.
8. Unfortunately, no. Problems beyond the control of hardware and software design, such as mechanical malfunctions and electrical problems, can affect computations.

Chapter 6

Section 6.1

1. A program in a third-generation language is machine independent in the sense that its steps are not stated in terms of the machine's attributes such as registers and memory cell addresses. On the other hand, it is machine dependent in the sense that arithmetic overflow and truncation errors will still occur.

2. The major distinction is that an assembler translates each instruction in the source program into a single machine instruction, whereas a compiler often produces many machine-language instructions to obtain the equivalent of a single source program instruction.
3. The declarative paradigm is based on developing a description of the problem to be solved. The functional paradigm forces the programmer to describe the problem's solution in terms of solutions to smaller problems. The object-oriented paradigm places emphasis on describing the components in the problem's environment.
4. The third-generation languages allow the program to be expressed more in terms of the problem's environment and less in terms of computer gibberish than do the earlier-generation languages.

Section 6.2

1. Using a descriptive constant can improve the accessibility of the program.
2. A declarative statement describes terminology; an imperative statement describes steps in an algorithm.
3. Integer, float, character, and Boolean
4. The `if-else` and `while` loop structures are very common.
5. All components of an array have the same type.

Section 6.3

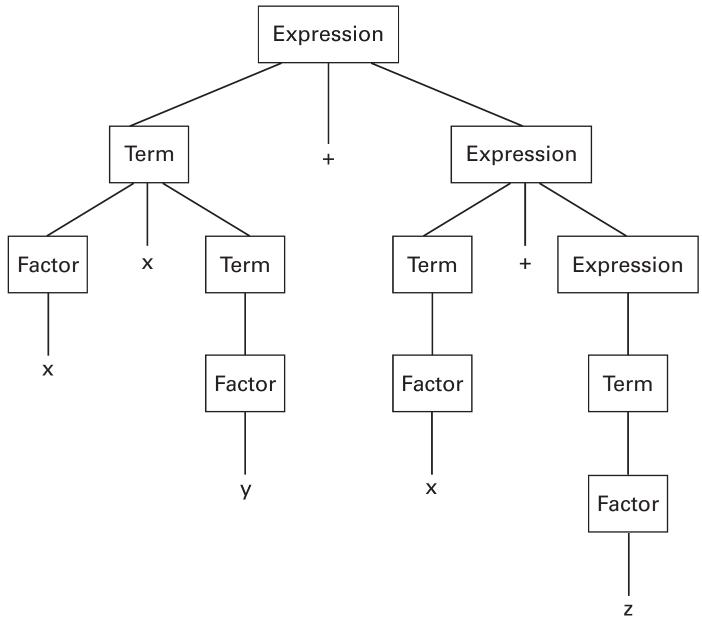
1. The scope of a variable is the range of the program in which that variable is accessible.
2. A fruitful function is a function that returns a value associated with the function's name.
3. Because that is what they are. I/O operations are actually calls to routines within the machine's operating system.
4. A formal parameter is an identifier within a function. It serves as a place holder for the value, the actual parameter, that is passed to the function when the function is called.
5. A function that passes parameters call-by-reference can potentially make changes to the parameters that will be visible to the caller; call-by-value parameters are copies, and changes made to them within the function will not be visible from outside.

Section 6.4

1. Lexical analysis: the process of identifying tokens.
   Parsing: the process of recognizing the grammatical structure of the program.
   Code generation: the process of producing the instructions in the object program.
2. A symbol table is the record of information the parser has obtained from the program's declarative statements.

3. In the syntax diagrams, terms that appear in ovals are terminals. Terms that require further description are in rectangles, and are called "nonterminals."

4.



5. The strings that conform to the structure Chacha consist of one or more of the following substrings:
forward backward cha cha cha
backward forward cha cha cha
swing right cha cha cha
swing left cha cha cha

6. `Python keywords appear in this color. Module names, functions and classes appear in this color, as do first time variable assignments. String constants and comments appear in this color. All other Python code appears in this color.`

Section 6.5

1. A class is the description of an object.
2. One would probably be `MeteorClass` from which the various meteors would be constructed. Within the class `LaserClass` one might find an instance variable named `AimDirection` indicating the direction in which the laser is aimed. This variable would probably be used by the `fire`, `turnRight`, and `turnLeft` methods.
3. The `Employee` class might contain features relating to an employee's name, address, years in service, etc. The `FullTimeEmployee` class might contain features relating to retirement benefits. The `PartTimeEmployee` class might contain features relating to hours worked per week, hourly wage, etc.

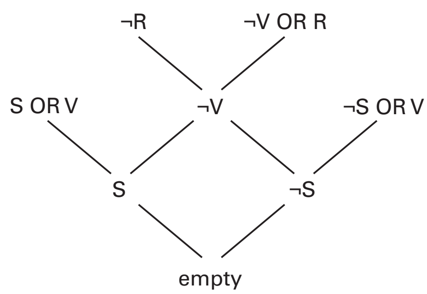
4. A constructor is a special method in a class that is executed when an instance of the class is created.
5. Some items in a class are designated as private to keep other program units from gaining direct access to those items. If an item is private, then the repercussions of modifying that item should be restricted to the interior of the class.

Section 6.6

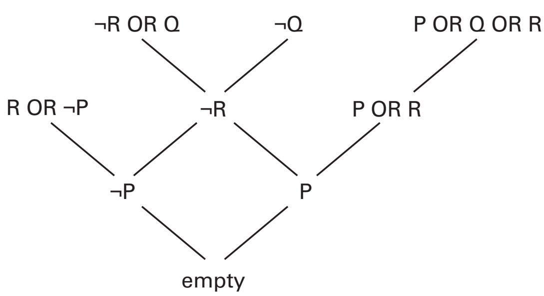
1. The list would include techniques for initiating the execution of concurrent processes and techniques for implementing interprocess communication.
2. One is to place the burden on the processes, another is to place the burden on the data. The latter has the advantage of concentrating the task at a single point in the program.
3. These include weather forecasting, air traffic control, simulation of complex systems (from nuclear reactions to pedestrian traffic), computer networking, and database maintenance.

Section 6.7

1. R, T, and V. For instance, we can show that R is a consequence by adding its negation to the collection and showing that resolution can lead to the empty statement, as shown here:



2. No. The collection is inconsistent, because resolution can lead to the empty statement, as shown here:



3.
```
mother(X, Y) :- parent(X, Y), female(X).
father(X, Y) :- parent(X, Y), male(X).
```

4. Prolog will conclude that `carol` is her own sibling. To solve this problem, the rule needs to include the fact that `X` cannot be equal to `Y`, which in Prolog is written `X \ = Y`. Thus an improved version of the rule would be

```
sibling (X, Y) :-X \= Y, parent(Z, X), parent(Z, Y).
```

which says that `X` is `Y`'s sibling if `X` and `Y` are not equal and have a common parent. The following version would insist that `X` and `Y` are siblings only if they have both parents in common:

```
sibling (X, Y) :- X \= Y, Z \= W
parent (Z, X), parent (Z, Y),
parent (W, X), parent (W, Y).
```

Chapter 7

Section 7.1

1. A long sequence of assignment statements is not as complex in the context of program design as a few nested `if` statements.
2. How about the number of errors found after a fixed period of use? One problem here is that this value cannot be measured in advance.
3. The point here is to think about how software properties can be measured. One approach for estimating the number of errors in a piece of software is to intentionally place some errors in the software when it is designed. Then, after the software has supposedly been debugged, check to see how many of the original errors are still present. For example, if you intentionally place seven errors in the software and find that five have been removed after debugging, then you might conjecture that only $5/7$ of the total errors in the software have been removed.
4. Possible answers include the discovery of metrics, the development of prefabricated components, the development of CASE tools, the move toward standards. Another, which is covered later in Section 7.5, is the development of modeling and notational systems such as UML.

Section 7.2

1. Small efforts made during development can pay enormous dividends during maintenance.
2. The requirements analysis phase concentrates on what the proposed system must accomplish. The design phase concentrates on how the system accomplishes its goals. The implementation phase concentrates on the actual construction of the system. The testing phase concentrates on making sure that the system does what it is intended to do.
3. A software requirements specification is a written agreement between a client and a software engineering firm stating the requirements and specifications of the software to be developed.

Section 7.3

1. The traditional waterfall approach dictates that the requirements analysis, design, implementation, and testing phases be performed in a linear manner. The newer models allow for a more relaxed trial-and-error approach.
2. How about the incremental model, the iterative model, and XP?
3. Traditional evolutionary prototyping is performed within the organization developing the software, whereas open-source development is not restricted to an organization. In the case of open-source development the person overseeing the development does not necessarily determine what enhancements will be reported, whereas in the case of traditional evolutionary prototyping the person managing the software development assigns personnel to specific enhancement tasks.
4. This is one for you to think about. If you were an administrator in a software development company, would you be able to adopt the open-source methodology for the development of software to be sold by your company?

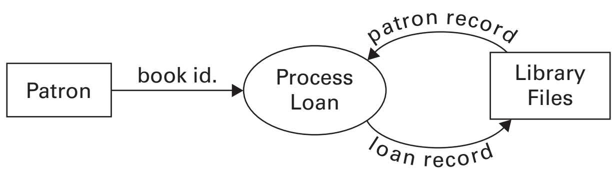
Section 7.4

1. The chapters of a novel build on one another, whereas the sections in an encyclopedia are largely independent. Hence, a novel has more coupling between its chapters than an encyclopedia has between its sections. However, the sections within an encyclopedia probably have a higher level of cohesion than the chapters in a novel.
2. The accumulated score would be an example of data coupling. Other "couplings" that might exist would include fatigue, momentum, knowledge gained about an opponent's strategy, and perhaps self-confidence. In many sports the cohesion of the units is increased by terminating the action and restarting the next unit from a fresh beginning. For example, in baseball each inning starts without any base runners, even though the team might have finished the previous inning with the bases loaded. In other cases the units are scored separately as in tennis where each set is won or lost without regard for the other sets.
3. This is a tough one. From one point of view, we could start by placing everything in a single module. This would result in little cohesion and no coupling at all. If we then begin to divide this single module into smaller ones, the result would be an increase in coupling. We might therefore conclude that increasing cohesion tends to increase coupling. On the other hand, suppose the problem at hand naturally divides into three very cohesive modules, which we will call A, B, and C. If our original design did not observe this natural division (for example, half of task A might be placed with half of task B, and so on), we would expect the cohesion to be low and the coupling high. In this case, redesigning the system by isolating tasks A, B, and C into separate modules would most likely decrease intermodule coupling as intramodule cohesion increases.
4. Coupling is linking between modules. Cohesion is the connectedness within a module. Information hiding is the restriction of information sharing.

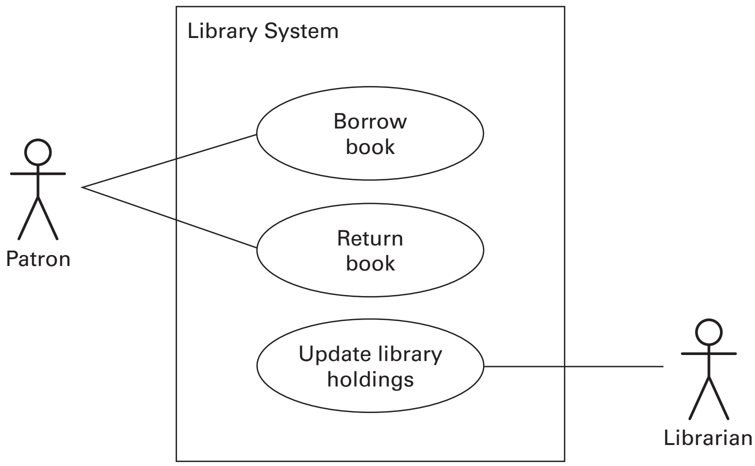
5. You should probably add an arrow indicating that `ControlGame` must tell `UpdateScore` who won the volley and another arrow in the other direction indicating that `UpdateScore` will report the current status (such as "set over" or "match over") when it returns control to `ControlGame`.
6. Delete all the horizontal arrows in Figure 7.5 except for the first and last. That is, the judge should evaluate `Player A's` serve and directly send the `updateScore` message to `Score`. (This, of course, ignores the chance for a second serve. How could you modify the program design to allow for double faults?)
7. A traditional programmer writes programs in terms of statements such as those introduced in Chapter 6. A component assembler builds programs by linking prefabricated blocks called components.
8. There are many answers to this question. One combination is to have the calendar automatically set an alarm in a clock to notifying the user of an upcoming appointment. Furthermore, the calendar application could use the components of a map application to provide the directions to the address of the appointment.

Section 7.5

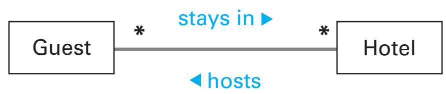
1. Make sure that your diagram deals with the flow of data (not the movement of books). The following diagram indicates that book identifications (from patrons) and patron records (from the library files) are combined to form loan records that are stored in the library files.



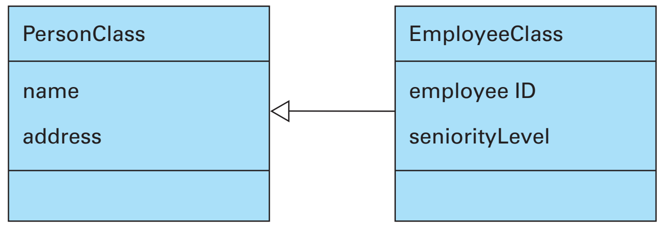
2.



3.



4.



5. Simply draw a rectangle around the figure and add a “sd” label in the upper left-hand corner as in Figure 7.13.
6. Design patterns provide standardized, well-developed approaches for implementing recurring software themes.

Section 7.6

1. The SQA (software quality assurance) group oversees and enforces the quality control systems adopted by the organization.
2. Humans have a tendency not to record the steps (decisions, actions, etc.) that they take during a project. (There are also issues of personality conflicts, jealousies, and ego clashes.)
3. Record keeping and reviewing.
4. The purpose of testing software is to find errors. In a sense, then, a test that does not reveal an error is a failure.
5. One would be to consider the amount of branching in the modules. For instance, a procedural module containing numerous loops and `if-else` statements would probably be more prone to errors than a module with a simple logical structure.
6. Boundary value analysis would suggest that you test the software on a list with 100 entries as well as a list with no entries. You might also perform a test with a list that is already in the correct order.

Section 7.7

1. Documentation takes the form of user documentation, system documentation, and technical documentation. It might appear in accompanying manuals, within the source program in the form of comments and well-written code, through interactive messages that the program itself writes at a terminal, through data dictionaries, and in the form of design documents such as structure charts, class diagrams, dataflow diagrams, and entity-relationship diagrams.
2. In both the development and modification phases. The point is that modifications must be documented as thoroughly as the original program. (It is also true that software is documented while in its use phase. For example, a user of the system might discover problems, which, rather than being fixed, are merely reported in future editions of the system user’s manual.

Moreover, "how to" books are often produced after the software has been in use for an extended period.)

3. Different people will have different opinions on this one. Some will argue that the program is the point of the whole project and thus is naturally the more important. Others will argue that a program is worth nothing if it is not documented, because if you cannot understand a program, you cannot use it or modify it. Moreover, with good documentation, the task of creating the program can be "easily" re-created.

Section 7.8

1. a. How about the ability to adjust the tilt of a display or the shape of a mouse? On smartphones, how about the use of touch screens instead of a mouse, or tilting the phone to provide input?
   b. How about the layout of a window on the display including the design of toolbars, scroll elevators, and pull-down menus? On a smartphone, isn't titling the camera to point at the items of interest in line with the way humans think?
2. a. It would be impractical and inconvenient to use a mouse (or even a stylus) on a smartphone. Furthermore, the reduced size of the display screen requires that nonessential elements of the display be constrained to limited space. For this reason, scrollbars are often omitted. If present, scrollbars are shown as thin lines.
   b. A sliding touch on the display screen is a natural gesture to the way we think. We may move papers or other items by sliding them around on a desk. An augment can be made that this is more natural than the use of scrollbars on a desktop computer. While indeed the scrollbar moves as expected, the area being scrolled moves in the opposite direction. For a user who has never used a computer, this behavior may seem counterintuitive.
3. You could answer "the role of human characteristics." Another good answer would be that interface design focuses on the external, rather than the internal, characteristics of a software system.
4. The three that are discussed in the text are the formation of habits, the narrowness of attention, and limited multiprocessing capabilities. Can you imagine others? How about the tendency to make assumptions?

Section 7.9

1. The copyright notice asserts ownership of the work and identifies personnel authorized to use the work. All works including requirements specifications, design documents, source code, and the final product usually involve a considerable investment to produce. An individual or corporation should take the steps to insure that their ownership rights are reserved and that all intellectual property is not used by undesired parties.
2. Copyright and patent laws benefit society because they encourage creators of new products to make them available to the public. Without such protection, companies would hesitate to make major investments in new products.
3. A disclaimer does not protect a company against negligence.

Chapter 8

Section 8.1

1. List: A listing of the members of a sports team.
   Stack: The stack of trays in a cafeteria.
   Queue: The line at a cafeteria.
   Tree: The organization chart of many governments.
2. Stacks and queues can be thought of as special types of lists. In the case of a general list, entries can be inserted and removed at any location. In the case of a stack, entries can be inserted and removed only at the head. In the case of a queue, entries can be inserted only at the tail, and entries can be removed only at the head.
3. The letters on the stack from top to bottom would be E, D, B, and A. If a letter were popped off the stack, it would be the letter E.
4. The letters in the queue from head to tail would be B, C, D, and E. If a letter were removed from the queue, it would be the letter B.
5. The leaf (or terminal) nodes are D and C. B must be the root node because all the other nodes have parents.

Section 8.2

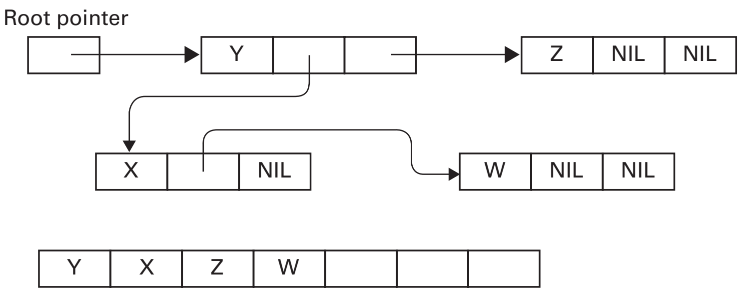
1. Data within a computer's main memory is actually stored in individually addressable memory cells. Structures such as arrays, lists, and trees are simulated to make the data more accessible to the data's users.
2. If you were to write a program for playing a game of checkers, the data structure representing the checkerboard would probably be a static structure because the size of the board does not change during the game. However, if you were to write a program for playing a game of dominoes, the data structure representing the pattern of dominoes constructed on the table would probably be a dynamic structure because this pattern varies in size and cannot be predetermined.
3. A telephone directory is essentially a collection of pointers (telephone numbers) to people. The clues left at the scene of a crime are (perhaps encrypted) pointers to the perpetrator.

Section 8.3

1. 5 3 7 4 2 8 1 9 6
2. If R is the number of rows in the matrix, the formula is $R(J - 1) + (I - 1)$.
3. $(c - i) + j$
4. The head pointer contains the NIL value.
5.
```
def PrintList (List):
   Last = the last name to be printed
   Finished = False
   CurrentPointer = List.Head
```

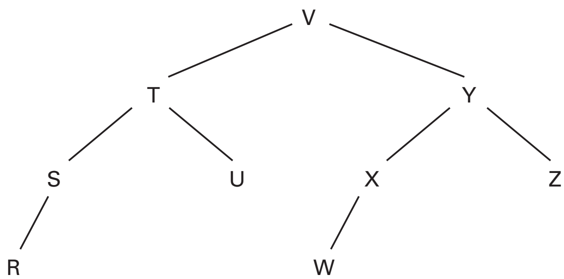
```
while ((CurrentPointer != None) and (Finished != False)):
  print(CurrentPointer.Value)
  if (name just printed == Last):
    Finished = True
  CurrentPointer = CurrentPointer.Next
```

6. The stack pointer points to the cell immediately below the base of the stack.
7. Represent the stack as a one-dimensional array and the stack pointer as a variable of integer type. Then use this stack pointer to maintain a record of the position of the stack's top within the array rather than of the exact memory address.
8. Both empty and full conditions are indicated by the equal head and tail pointers. Thus additional information is required to distinguish between the two conditions.
9.

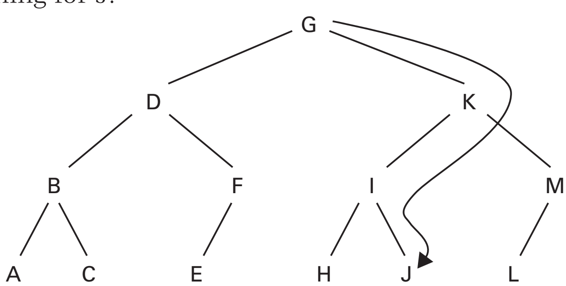


Section 8.4

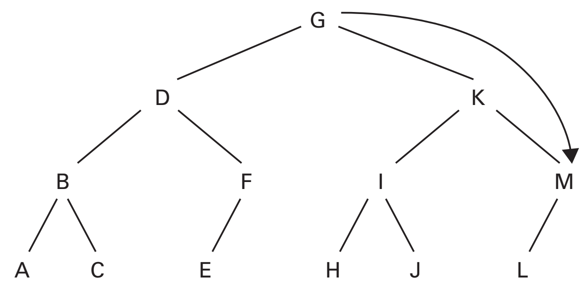
1.



2. When searching for J:



When searching for P:



3.

```
def PrintTree (Tree):
  if (Tree is not None):
    PrintTree(Tree.Left)
    print(Tree.Value)
    PrintTree(Tree.Right)
```

```
def PrintTree (Tree):
  if (Tree is not None):
    PrintTree(Tree.Left)
    print(Tree.Value)
    PrintTree(Tree.Right)
```

Here, when K is printed

4. At each node, each child pointer could be used to represent a unique letter in the alphabet. A word could be represented by a path down the tree along the sequence of pointers representing the spelling of the word. A node could be marked in a special way if it represented the end of a correctly spelled word.

Section 8.5

1. A type is a template; an instance of that type is an actual entity built from that template. As an analogy, dog is a type of animal, whereas Lassie and Rex are instances of that type.
2. A user-defined data type is a description of data organization, whereas an abstract data type includes operations for manipulating the data.
3. A point to be made here is that you have a choice between implementing the list as a contiguous list or a linked list. The choice you make will affect the structure of the functions for inserting new entries, deleting old ones, and finding entries of interest. However, this choice should not be visible to a user of an instance of the abstract data type.
4. The abstract data type would at least contain a description of a data structure for storing the account balance and functions for making a deposit and making a withdrawal via a check.

Section 8.6

1. Both abstract data types and classes are templates for constructing instances of a type. Classes, however, are more general in that they are associated with inheritance and might describe a collection of only functions.

2. A class is a template from which objects are constructed.
3. The class might contain a circular queue along with functions for adding entries, removing entries, testing to see if the queue is full, and testing to see if the queue is empty.

Section 8.7

1. a. A5 b. A5 c. CA
2. D50F, 2EFF, 5FFE
3. 2EA0, 2FB0, 2101, 20B5, D50E, E50F, 5EE1, 5FF1, BF14, B008, C000
4. When traversing a linked list in which each entry consists of two memory cells (a data cell followed by a pointer to the next entry), an instruction of the form DR0S could be used to retrieve the data and DR1S could be used to retrieve the pointer to the next entry. If the form DRTS were used, then the exact memory cell being referenced could be adjusted by modifying the value in register T.

Chapter 9

Section 9.1

1. The purchasing department would be interested in inventory records to place orders for more raw goods, whereas the accounting department would need the information to balance the books.
2. A database model provides an organizational perspective of a database that is more compatible with applications than the actual organization. Thus defining a database model is the first step toward allowing the database to be used as an abstract tool.
3. The application software translates the user's requests from the terminology of the application into terminology compatible with the database model that is supported by the database management system. The database management system in turn converts these requests into actions on the actual database.

Section 9.2

1. a. Jerry Smith b. Cheryl H. Clark c. S26Z
2. One solution is

```
TEMP ← SELECT from JOB
  where Dept = "PERSONNEL"
LIST ← PROJECT JobTitle from TEMP
```

   In some systems this results in a list with a job title repeated, depending on how many times it occurred in the personnel department. That is, our list might contain numerous occurrences of the title secretary. It is more common, however, to design the PROJECT operation so that it removes duplicate tuples from the resulting relation.

3. One solution is

```
TEMP1  ←  JOIN JOB and ASSIGNMENT
                  where JOB.JobId = ASSIGNMENT.JobId
TEMP2  ←  SELECT from TEMP1
                  where TermDate = '*'
TEMP3  ←  JOIN EMPLOYEE and TEMP2
                  where EMPLOYEE.EmplId = TEMP2.EmplId
RESULT  ←  PROJECT Name, Dept from TEMP3
```

4.
```
SELECT JobTitle
  FROM Job
  WHERE Dept = 'PERSONNEL';
```

```
SELECT Employee.Name, Job.Dept
  FROM Job, Assignment, and Employee
  WHERE (Job.Job = Assignment.JobId) and
     (Assignment.EmplId = Employee.EmplID)
      and (Assignment.TermDate = '*');
```

5. The model itself does not provide data independence. This is a property of the data management system. Data independence is achieved by providing the data management system the ability to present a consistent relational organization to the application software even though the actual organization might change.

6. Through common attributes. For instance, the `EMPLOYEE` relation in this section is tied to the `ASSIGNMENT` relation via the attribute `EmplId`, and the `ASSIGNMENT` relation is tied to the `JOB` relation by the attribute `JobId`. Attributes used to connect relations like this are sometimes called connection attributes.

Section 9.3

1. There might be methods for assigning and retrieving the `StartDate` as well as the `TermDate`. Another method might be provided for reporting the total time in service.
2. A persistent object is an object that is stored indefinitely.
3. One approach is to establish an object for each type of product in inventory. Each of these objects could maintain the total inventory of its product, the cost of the product, and links to the outstanding orders for the product.
4. As indicated at the beginning of this section, object-oriented databases appear to handle composite data types more easily than relational databases. Moreover, the fact that objects can contain methods that take an active role in answering questions promises to give object-oriented databases an advantage over relational databases whose relations merely hold the data.

Section 9.4

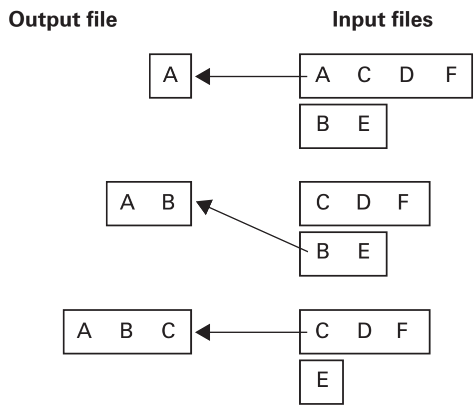
1. Once a transaction has reached its commit point, the database management system accepts the responsibility of seeing that the complete transaction is performed on the database. A transaction that has not reached its commit

point does not have such assurance. If problems arise, it might have to be resubmitted.

2. One approach would be to stop interweaving transactions for an instant so that all current transactions can be completed in full. This would establish a point at which a future cascading rollback would terminate.
3. A balance of $100 would result if the transactions were executed one at a time. A balance of $200 would result if the first transaction were executed after the second transaction retrieved the original balance and before that second transaction stored its new balance. A balance of $300 would result if the second transaction were executed after the first retrieved the original balance and before the first transaction stored its new balance.
4. a. If no other transaction has exclusive access, the shared access will be granted.
   b. If another transaction already has some form of access, the database management system will normally make the new transaction wait, or it could roll back the other transactions and give access to the new transaction.
5. Deadlock would occur if each of two transactions acquired exclusive access to different items and then required access to the other.
6. The preceding deadlock could be removed by rolling back one of the transactions (using the log) and giving the other transaction access to the data item previously held by the first.

Section 9.5

1. You should be led through these initial stages:



2. The idea is to first divide the file to be stored into many separate files containing one record each. Next, group the one-record files into pairs, and apply the merge algorithm to each pair. This results in half as many files, each with two records. Furthermore, each of these two-record files is sorted. We can group them into pairs and again apply the merge algorithm to the pairs. Again we find ourselves with fewer but larger files, each of which is sorted. Continuing in this fashion, we are ultimately left with only one file that consists of all the original records but in sorted order. (If an odd

number of files occurs at any stage of this process, we need merely to set the odd one aside and pair it with one of the larger files in the next stage.)

3. If the file is stored on tape or CD, its physical organization is most likely sequential. However, if the file is stored on magnetic disk, then it is most likely scattered over various sectors on the disk and the sequential nature of the file is a conceptual property that is supported by a pointer system or some form of a list in which the sectors on which the file is stored are recorded.
4. First find the target key in the file's index. From there, obtain the location of the target record. Then retrieve the record at that location.
5. A poorly chosen hash algorithm results in more clustering than normal and thus in more overflow. Because the overflow from each section of mass storage is organized as a linked list, searching through the overflow records is essentially searching a sequential file.
6. The section assignments are as follows:

a. 0	b. 0	c. 3	d. 0	e. 3
f. 3	g. 3	h. 3	i. 3	j. 0

   Thus all the records hash into buckets 0 and 3, leaving buckets 1, 2, 4, and 5 empty. The problem here is that the number of buckets being used (6) and the key values have the common factor of 3. (You might try rehashing these key values using 7 buckets and see what improvement you find.)

7. The point here is that we are essentially applying a hash algorithm to place the people in the group into one of 365 categories. The hash algorithm, of course, is the calculation of one's birthday. The amazing thing is that only twenty-three people are required before the probability is in favor of at least two of the birthdays being the same. In terms of a hashed file, this indicates that when hashing records into 365 available buckets of mass storage, clustering is likely to be present after only twenty-three records have been entered.

Section 9.6

1. Searching for patterns in dynamic data is problematic.
2. Class description—Identify characteristics of subscribers to a certain magazine.
   Class discrimination—Identify features that distinguish between subscribers of two magazines.
   Cluster analysis—Identify magazines that tend to attract similar subscribers.
   Association analysis—Identify links between subscribers to various magazines and different purchasing habits.
   Outlier analysis—Identify subscribers to a magazine who do not conform to the profile of normal subscribers.
   Sequential pattern analysis—Identify trends in magazine subscription.
3. The data cube might allow sales data to be viewed as sales by month, sales by geographic region, sales by product class, etc.
4. Traditional database inquiries retrieve facts stored in the database. Data mining looks for patterns among the facts.

Section 9.7

1. The point here is to compare your answer to this question with that of the next. The two raise essentially the same question but in different contexts.
2. See previous problem.
3. You might receive announcements or advertisements for opportunities that you would not have otherwise received, but you might also become the subject of solicitation or the target of crime.
4. The point here is that a free press can alert the public to abuses or potential abuses and thus bring public opinion into play. In most of the cases cited in the text, it was a free press that initiated corrective action by alerting the public.

Chapter 10

Section 10.1

1. Image processing deals with analyzing two-dimensional images, 2D graphics deals with converting two-dimensional shapes into images, and 3D graphics deals with converting three-dimensional scenes into images.
2. Traditional photography produces images of actual scenes, whereas 3D graphics produces images of virtual scenes.
3. The first is "building" the virtual scene. The second is capturing the image.

Section 10.2

1. The steps are modeling (building the scene), rendering (producing a picture), and displaying (displaying the picture).
2. The image window is the portion of the projection plane that constitutes the image.
3. A frame buffer is a memory area that contains an encoded version of an image.

Section 10.3

1. It is a rhombus (a squashed square).
2. A procedural model is a program segment that directs the construction of an object.
3. The list could include the grass-covered ground, a stone walkway, a gazebo, trees, shrubbery, clouds, sun, and actors. The point here is to emphasize the scope of a scene graph—it can contain a lot of detail.
4. Representing all objects by polygonal meshes provides a uniform approach to the rendering process. (In most cases, rendering is approached as the task of rendering planar patches rather than rendering objects.)
5. Texture mapping is a means of associating a two-dimensional image with the surface of an object.

Section 10.4

1. Specular light is light that is "directly" reflected off a surface. Diffuse light is light that is "scattered" off a surface. Ambient light is light that does not have a precise source.
2. Clipping is the process of discarding those objects (and parts of objects) that do not lie within the view volume.
3. Suppose a highlight should appear in the middle of a patch. That highlight is caused by a specific surface orientation at that point of the patch. Because Gouraud shading considers only the surface orientations along the boundaries of the patch, it will miss the highlight. But, because Phong shading attempts to determine the surface orientations within the patch interior, it may detect the highlight.
4. The rendering pipeline provides a standardized approach to rendering, which ultimately leads to more efficient rendering systems. In particular, the rendering pipeline can be implemented in firmware, meaning that the rendering process can be performed more quickly than if the task were implemented via traditional software.
5. The purpose of this question is to get you to think about the distinctions between local and global lighting models rather than to produce a specific predetermined answer. Potential solutions that you might propose include placing appropriately modified copies of the objects to be reflected behind the mirror while considering the mirror transparent or trying to handle images in the mirror as a form of drop shadows.

Section 10.5

1. We are interested only in the rays that ultimately reach the image window. If we started at the light source, we would not know which rays to follow.
2. Distributed ray tracing tries to avoid the inherent shiny appearance produced by traditional ray tracing by tracing multiple rays.
3. Radiosity is time consuming and fails to capture specular affects.
4. Both ray tracing and radiosity implement a global lighting model, and both are computationally intense. However, ray tracing tends to produce shiny-appearing surfaces, whereas radiosity leads to dull-appearing surfaces.

Section 10.6

1. There is not an exact answer. If an image lingers for 200 milliseconds and we projected five frames per second, each frame would have just faded away by the time the next frame was projected. This would probably result in a pulsating image that would be uncomfortable to watch for an extended time but still produce an animated effect. (In fact, slower rates can produce rough animation.) Note that the rate of five frames per second is well below the motion picture standard of twenty-four frames per second.
2. A storyboard is a "pictorial outline" of the desired animation sequence.
3. In-betweening is the process of creating frames that fill in the gaps between key frames.

4. Dynamics is the branch of mechanics that analyzes motion as the consequence of forces. Kinematics is the branch of mechanics that analyzes motion without regard for the forces that cause the motion.

Chapter 11

Section 11.1

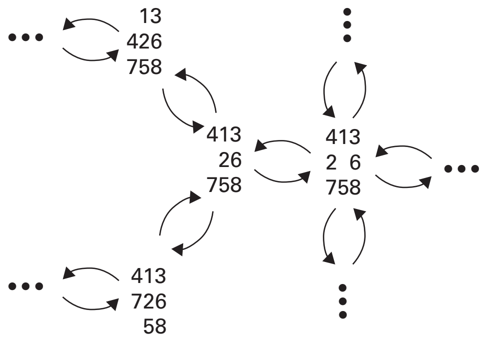
1. Those introduced in the chapter include reflex actions, actions based on real-world knowledge, goal seeking actions, learning, and perception.
2. Our purpose here is not to give a decisive answer to this issue but to use it to show how delicate the argument over the existence of intelligence really is.
3. Although most of us would probably say no, we would probably claim that if a human dispensed the same products in a similar atmosphere, awareness would be present even though we might not be able to explain the distinction.
4. There is not a right or wrong answer. Most would agree that the machine at least appears to be intelligent.
5. There is not a right or wrong answer. It should be noted that chat bots, programs designed to emulate a person chatting, have difficulty carrying on a meaningful conversation for even a short period of time. Chat bots are easily identified as machines.

Section 11.2

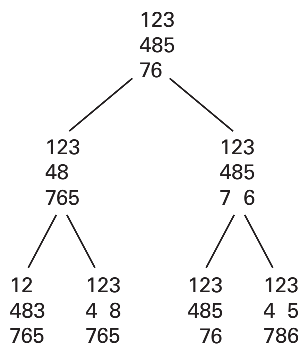
1. In the remote control case, the system needs only to relay the picture, whereas to use the picture for maneuvering, the robot must be able to "understand" the meaning of the picture.
2. The possible interpretations for one section of the drawing do not match any of those of another section. To embed this insight into a program, you might isolate the interpretations allowable for various line junctions and then write a program that tries to find a set of compatible interpretations (one for each junction). In fact, if you stop and think about it, this is probably what your own senses did in trying to evaluate the drawing. Did you detect your eyes scanning back and forth between the two ends of the drawing as your senses tried to piece possible interpretations together? (If this subject interests you, you will want to read about the work of people such as D. A. Huffman, M. B. Clowes, and D. Waltz.)
3. There are four blocks in the stack but only three are visible. The point is that understanding this apparently simple concept requires a significant amount of "intelligence."
4. Interesting, isn't it? Such subtle distinctions in meaning present significant problems in the field of natural language understanding.
5. Is the sentence describing what kind of horses they are, or is it telling what some people are doing?
6. The parsing process produces identical structures, but the semantic analysis recognizes that the prepositional phrase in the first sentence tells where the fence was built, whereas the phrase in the second sentence tells when the fence was built.
7. They are brother and sister.

Section 11.3

1. Production systems provide a uniform approach to a variety of problems. That is, although apparently different in their original form, all problems reformulated into terms of production systems become the problem of finding a path through a state graph.

2. 

3. The tree is four moves deep. The upper portion appears as follows:



4. The task requires too much paper as well as too much time.

5. Our heuristic system for solving the eight-puzzle is based on an analysis of the immediate situation, just as that of the mountain climber. This shortsightedness is what allowed our algorithm to proceed initially along the wrong path in the example of this section just as a mountain climber can be led into trouble by always plotting a course based only on the local terrain. (This analogy often causes heuristic systems based on local or immediate information to be called hill-climbing systems.)

6. The system rotates the 5, 6, and 8 tiles either clockwise or counterclockwise until the goal state is reached.

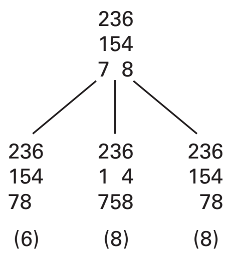
7. The problem here is that our heuristic scheme ignores the value of keeping the hole adjacent to the tiles that are out of place. If the hole is surrounded by tiles in their correct position, some of these tiles must be moved before those tiles still seeking their correct place can be moved. Thus it is incorrect to consider all those tiles surrounding the hole as actually being correct. To fix this flaw, we might first observe that a tile in its correct position but blocking the hole from incorrectly positioned tiles must be moved away from its correct position and later moved back. Thus each correctly positioned tile on a path between the hole and the nearest incorrectly

positioned tile accounts for at least two moves in the remaining solution. We can therefore modify our projected cost calculation as follows:

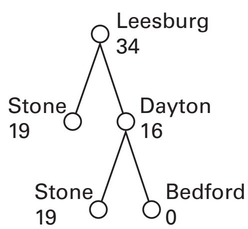
First, calculate the projected cost as before. However, if the hole is totally isolated from the incorrectly positioned tiles, find a shortest path between the hole and an incorrectly positioned tile, multiply the number of tiles on this path by two, and add the resulting value to the previous projected cost.

With this system, the leaf nodes in Figure 11.10 have projected costs of 6, 6, and 4 (from left to right), and thus the correct branch is pursued initially.

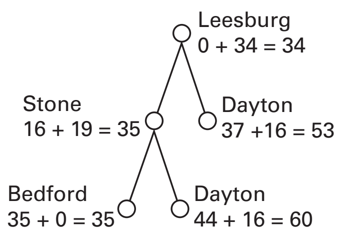
Our new system is not foolproof. For example, consider the following configuration. The solution is to slide the 5 tile down, rotate the top two rows clockwise until those tiles are correct, move the 5 tile back up, and finally move the 8 tile to its correct position. However, our new heuristic system wants us to start by moving the 8 tile, because the state obtained by this initial move has a projected cost of only 6 compared with the other options that have costs of 8.



8. The solution found by the best fit algorithm is the path from Leesburg to Dayton and then to Bedford. This path is not the shortest route.



9. The solution found is the path from Leesburg to Stone, and then to Bedford. This path is the shortest route.



Section 11.4

1. Real-world knowledge is the information about the environment that a human uses to understand and reason. Developing methods for representing, storing, and recalling this information is a major goal of research in artificial intelligence.

2. It uses the closed-world assumption.
3. The frame problem is the problem of correctly updating a machine's store of knowledge as events occur. The task is complicated by the fact that many events have indirect consequences.
4. Imitation, supervised training, and reinforcement. Reinforcement does not involve direct human intervention.
5. Traditional techniques derive a single computer system. Evolutionary techniques involve multiple generations of trial systems from which a "good" system may be discovered.

Section 11.5

1. All patterns produce an output of 0 except for the pattern 1, 0, which produces an output of 1.
2. Assign a weight of 1 to each input, and assign the unit a threshold value of 1.5.
3. A major problem identified in the text is that the training process might oscillate, repeating the same adjustments over and over.
4. The network will wander to the configuration in which the center neuron is excited and all others are inhibited.

Section 11.6

1. Rather than developing a complete plan of action, the reactive approach is to wait and make decisions as options arise.
2. The point here is for you to think about how broad the field of robotics is. It encompasses the entire scope of artificial intelligence as well as numerous topics in other fields. The goal is to develop truly autonomous machines that can move about and react intelligently with their environments.
3. Internal control and physical structure.

Section 11.7

1. There is no right or wrong answer.
2. There is no right or wrong answer.
3. There is no right or wrong answer.

Chapter 12

Section 12.1

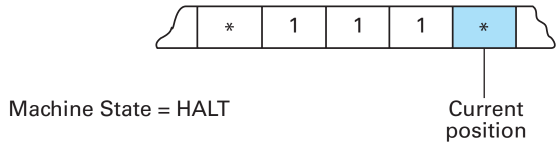
1. How about the boolean operations AND, OR, and XOR. In fact, we used tables in Chapter 1 when introducing these functions.
2. The computation of a loan payment, the area of a circle, or a car's mileage.
3. Mathematicians call such functions transcendental functions. Examples include the logarithmic and trigonometric functions. These particular

examples can still be computed but not by algebraic means. For example, the trigonometric functions can be calculated by actually drawing the triangle involved, measuring its sides, and only then turning to the algebraic operation of dividing.

4. One example is the problem of trisecting an angle. That is, they were unable to construct an angle that was one-third the size of a given angle. The point is that the Greeks' straight-edge and compass computational system is another example of a system with limitations.

Section 12.2

1. The result is the following diagram:



2.

Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	left	STATE 1
STATE 1	0	0	left	STATE 2
STATE 1	1	0	left	STATE 2
STATE 1	*	0	left	STATE 2
STATE 2	0	*	right	STATE 3
STATE 2	1	*	right	STATE 3
STATE 2	*	*	right	STATE 3
STATE 3	0	0	right	HALT
STATE 3	1	0	right	HALT

3.

Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	left	SUBTRACT
SUBTRACT	0	1	left	BORROW
SUBTRACT	1	0	left	NO BORROW
BORROW	0	1	left	BORROW
BORROW	1	0	left	NO BORROW
BORROW	*	*	right	ZERO
NO BORROW	0	0	left	NO BORROW
NO BORROW	1	1	left	NO BORROW
NO BORROW	*	*	right	RETURN
ZERO	0	0	right	ZERO
ZERO	1	0	right	ZERO
ZERO	*	*	no move	HALT
RETURN	0	0	right	RETURN
RETURN	1	1	right	RETURN
RETURN	*	*	no move	HALT

4. The point here is that the concept of a Turing machine is supposed to capture the meaning of "to compute." That is, any time a situation occurs in which computing is taking place, the components and activities of a Turing machine should be present. For example, a person figuring income tax is doing a certain degree of computing. The computing machine is the person and the tape is represented by the paper on which values are recorded.

5. The machine described by the following table halts if started with an even input but never halts if started with an odd input:

Current state	Cell content	Value to write	Direction to move	New state to enter
START	*	*	left	STATE 1
STATE 1	0	0	right	HALT
STATE 1	1	1	no move	STATE 1
STATE 1	*	*	no move	STATE 1

Section 12.3

1.
```
clear AUX
incr AUX
while X not 0:
  clear X
  clear AUX
while AUX not 0:
  incr X
  clear AUX
```

2.
```
while X not 0:
  decr X
```

3.
```
copy X to AUX
while AUX not 0:
  S1
  clear AUX
copy X to AUX
invert AUX (See Question #1)
while AUX not 0:
  S2
  clear AUX
while X not 0:
  clear AUX
  clear X
```

4. If we assume that X refers to the memory cell at address 40 and that each program segment starts at location 00, we have the following conversion table:

```
clear X
incr X
decr X
```

```
while X not 0:
   .
   .
   .
```

5. Just as in a real machine, negative numbers could be dealt with via a coding system. For example, the rightmost bit in each string can be used as a sign but with the remaining bits used to represent the magnitude of the value.
6. The function is multiplication by 2.

Section 12.4

1. Yes. In fact, this program halts regardless of the initial values of its variables, and therefore it must halt if its variables are initialized to the program's encoded representation.
2. The program halts only if the initial value of X ends in a 1. Because the ASCII representation of a semicolon is 00111011, the encoded version of the program must end in a 1. Therefore the program is self-terminating.
3. The point here is that the logic is the same as in our argument that the halting problem does not have an algorithmic solution. If the house painter paints his or her own house, then he or she does not and vice versa.

Section 12.5

1. We could conclude only that the problem has complexity $\Theta(2^{n).}$ If we could show that the "best algorithm" for solving the problem belongs to $\Theta(2^n)$, we could conclude that the problem belongs to $\Theta(2^{n).}$
2. No. As a general rule, the algorithm in $\Theta(n^2)$ will outperform the one in $\Theta(2^n)$, but for small input values an exponential algorithm often outperforms a polynomial algorithm. In fact, it is true that exponential algorithms are sometimes preferred to polynomial ones when the application involves only small inputs.
3. The point is that the number of subcommittees is growing exponentially, and from this point on, the job of listing all the possibilities becomes a laborious task.
4. Within the class of polynomial problems is the sorting problem, which can be solved by polynomial algorithms such as the insertion sort. Within the class of nonpolynomial problems is the task of listing all the subcommittees that could be formed from a given parent committee. Any polynomial problem is an NP problem. The Traveling Salesman problem is an example of an NP problem that has not been shown to be a polynomial problem.
5. No. Our use of the term *complexity* refers to the time required to execute an algorithm—not to how hard the algorithm might be to understand.

Section 12.6

1. $211 - 313 = 66043$
2. The message 101 is the binary representation for 5. $5^e = 5^5 = 15625$. 15625 (mod 91) $= 64$, which is 1000000 in binary notation. Thus, 1000000 is the encrypted version of the message.
3. The message 10 is the binary representation for 2. $2^d = 2^{29} = 536870912$. 536870912 (mod 91) $= 32$, which is 100000 in binary notation. Thus, 100000 is the decrypted version of the message.
4. $n = p - q = 7 - 19 = 133$. To find d we need a positive integer value k such that $k(p - 1)(q - 1) + 1 = k(6 - 18) + 1 = 108k + 1$ is evenly divisible by $e = 5$. The values $k = 1$ and $k = 2$ are not satisfactory, but $k = 3$ produces $108k + 1 = 325$, which is divisible by 5. The quotient 65 is the value of d.