Section A:

1. A = 10 * 16 = 160 + 7 = 167

2. 7 = 0111, .625 = 0.5 + 0.125 = 1010

0111.1010

3. 18 = 00010010

flip bits = 11101101

swap last bit = 11101110

4. Largest possible 8-bit 2's complement binary = 127

5. The number which is subtracted is changed into a negative number, which is then added to the first number to get the 2's complement. For example,

23 - 48 = -25

00010111 (23)

11010000 (-48)

11100111 = -25

6. 11101110

7. 11101011

8. The purpose of the finite state machine is to convert a positive binary value into a 2's complement value.

9.

Input	Original State	Output	New State
0	S0	0	S0
1	S0	1	S1
0	S1	1	S1
1	S1	0	S1

10. Unicode uses more bits for each character than ASCII.

11. Even parity works by counting the number of 1's and 0's in a binary string. If there is an odd number of 1's then the extra bit will make it an even number of 1's.

12. Hamming code can detect more than one error in transmission.

13. 0

14. 1, 4, and 8

15. 16 bit

16.

16 bits per second, 100 seconds, = 1600 bits / 1024 = 1.6 bytes

17. Nyquists Theorem says we should always use a sampling rate double the frequency in the recording, and humans won't be able to hear the higher pitches anyway so quality appears the same.

18. Because of the streaming speed, the computer may not be receiving the packages of data from the website quickly enough.

Section B:

Source Code:

```
for count in range(13):
    ISBN = int(input("Please enter the next digit of ISBN: "))
CalculatedDigit = 0
Count = 1
while Count < 13:
    CalculatedDigit = CalculatedDigit + ISBN
    Count = Count + 1
    CalculatedDigit = CalculatedDigit + ISBN * 3
    Count = Count + 1
while CalculatedDigit >= 10:
    CalculatedDigit = CalculatedDigit - 10
CalculatedDigit = 10 - CalculatedDigit
if CalculatedDigit == 10:
    CalculatedDigit = 10
if CalculatedDigit == ISBN:
    print("Valid ISBN")
else:
    print("Invalid ISBN")
```

Screen Capture 1:

Input: 9, 7, 8, 0, 0, 9, 9, 4, 1, 0, 6, 7, 6

```
Please enter the next digit of ISBN: 9
Please enter the next digit of ISBN: 7
Please enter the next digit of ISBN: 8
Please enter the next digit of ISBN: 0
Please enter the next digit of ISBN: 0
Please enter the next digit of ISBN: 9
Please enter the next digit of ISBN: 9
Please enter the next digit of ISBN: 4
Please enter the next digit of ISBN: 1
Please enter the next digit of ISBN: 0
Please enter the next digit of ISBN: 6
Please enter the next digit of ISBN: 7
Please enter the next digit of ISBN: 6
Valid ISBN
```

Screen Capture 2:

Input: 9, 7, 8, 1, 8, 5, 7, 0, 2, 8, 8, 9, 4

```
Please enter the next digit of ISBN: 9
Please enter the next digit of ISBN: 7
Please enter the next digit of ISBN: 8
Please enter the next digit of ISBN: 1
Please enter the next digit of ISBN: 8
Please enter the next digit of ISBN: 5
Please enter the next digit of ISBN: 7
Please enter the next digit of ISBN: 0
Please enter the next digit of ISBN: 2
Please enter the next digit of ISBN: 8
Please enter the next digit of ISBN: 8
Please enter the next digit of ISBN: 9
Please enter the next digit of ISBN: 4
Valid ISBN
```

Section C:

23. String

24. `int`

25. `Suit`

26. Follower

27. Most wanted holder

28. Gatherer

29. If an integer is input.

30. Validation should be included so that a while loop with the input statement inside is only broken if the input is a string.

31. Linear search