Q.1

4000 x 3000 = 12,000,000 pixels x 3 bytes / pixel = 36,000,000 bytes = 36Mb

a. 36,000,000 / 56,000 = 642.86 seconds
b. 36M / 1M = 36 seconds
c. 36M / 10M =
d. 36M / 500M = 0.072 seconds
e. 36M / 1000M = 0.036 seconds

Q.2

a. 4000-bits = 500bytes, 0.5kb/100kbps = 0.005s = 5ms
b. $6x10^7$ m / $2x10^8$ m/s = $3x10^{-1}$ seconds = 0.3s = 300ms
c. 5ms + 0.3s = 305ms

Q.3

a. 1920-bits = 240bytes, 0.24kb/64kbps = 0.00375s = 3.75ms
b. $3.9x10^7$ / $2x10^8$ m/s = $1.95x10^{-1}$ s = 0.195s = 195ms
c. 195ms + 3.75ms = 198.75ms

Q.4

a. 10^7 bit/s x $50x10^{-3}$ s = $5x10^5$ bits = 0.5Mbit
b. 10^9 bit/s x 10^{-3} s = 10^6 bits = 1Mbit

Q.5

a. 21Gb at 18,000m/3600s = 4.2Gb / m / s
   4.2Gb/m/s > 150Mb/s
   4,200Mb/m > 150Mb
   4,200/150 > m
   28 > m

The pigeon has a higher data for distances up to 28m.

b.
   i. The left side of 4.2Gb/m/s > 150Mb/s doubles with the pigeon speed, so the distances doubles to 56m.
   ii. Again the left side of the initial equation doubles, so the distance doubles to 56m.
   iii. The right side of 4.2Gb/m/s > 150Mb/s doubles, so the distance is halved to 14m.

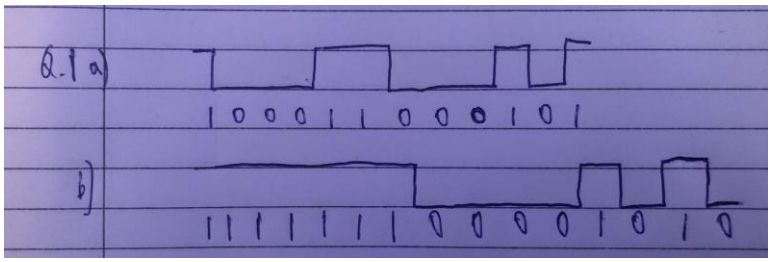
Q.6

80,000Mb/150Mbps = 533.33s - faster

100km/72km/hr = 1.389 hrs = 5,000s

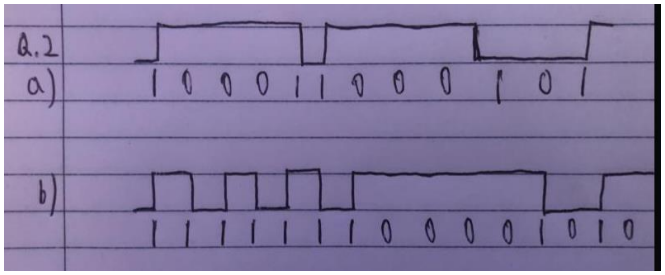
Modulation Schemes

Q.1



Synchronization may be difficult as there can be long series without change, like the 7 ones in part b.

Q.2

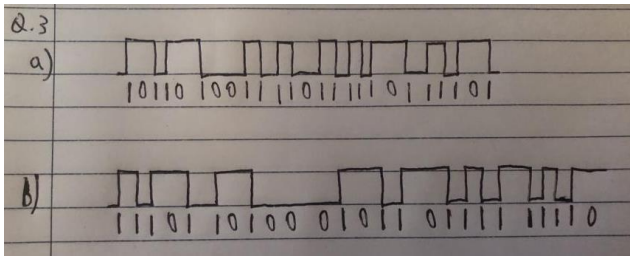


It partially solves it, as it causes long series of ones to have changes, but not long series of zeros.

Q.3

a. A = 10 = 1010
   9 = 1001
   D = 13 = 1101
   F = 15 = 1111
   4b input = 1010 1001 1101 1111 1111
   5b output = 10110 10011 11011 11101 11101

b. F = 1111
   2 = 0010
   5 = 0101
   7 = 0111
   0 = 0000
   4b input = 1111 0010 0101 0111 0000
   5b output = 11101 10100 01011 01111 11110



Yes, it solves the problem as no combination of the 5b outputs has more than 3 consecutive 0's.