COS 109 Problem Set 3

1. Numbers, Numbers, Numbers
   a) Calculation:
      
      \[ 1 \text{ PB} = 10^{15} \text{ B} \]
      
      Speed = 1 B / Sec
      
      Time = 10^{15} \text{ Sec} = 30 \text{ million years.}
      
      Answer:
      
      Just right

   b) Assume my backpack can hold 4 kg, then I can put \( \frac{4000}{2} = 2000 \) SD cards.
      
      i. \( 2000 \times 512 \text{ GB} / 10^6 = 1 \text{ PB} \)
      
      ii. \( 4 \text{ kg.} \)
      
      iii. 14 minutes = 840 sec.
      
      transfer rate = 1.2 TB / sec.

   c) \( 2^{33} = 8.6 \) billion.
      
      Not consistent. Current population is 7.53 billion < 8.6 billion < 10 billion.
      
      The second statement is wrong.

   d) \( R: 1111 \ 1000 \ 0000 \ 0000 = F800 \)
      
      \( G: 0000 \ 0111 \ 1110 \ 0000 = 07E0 \)
      
      \( B: 0000 \ 0000 \ 0001 \ 1111 = 001F \)

2. Bits, Bytes, Bases
   a) \( \log_2 960e9 = 40 \) bits = 5 bytes.
   b) \( \log_2 22.6e12 = 45 \) bits = 6 bytes.
   c) \( \log_2 190e6 = 28 \) bits = 4 bytes
   d) Transform to hex: 31 39 39 39
      
      Lookup in ASCII.
Answer: 1999.

3. Bloody Instructions
   b) Code Screenshot

   ```javascript
   function valid(op) {
       return /get|print|store|load|add|mul|sub|goto|ifpos|ifzero|stop/.test(op);
   }
   ```

   ```javascript
   accumulator = parseFloat(litoradr adr pc);
   else if (opcode[pc] == "add") {
   accumulator = parseFloat(accumulator) + parseFloat(litoradr adr pc);
   } else if (opcode[pc] == "mul") {
   accumulator = parseFloat(accumulator) * parseFloat(litoradr adr pc);
   } else if (opcode[pc] == "sub") {
   accumulator = parseFloat(accumulator) - parseFloat(litoradr adr pc);
   }
   ```

   ```plaintext
   store M store contents of accumulator into memory location called M (accumulator unchanged)
   add M add M to contents of accumulator (M unchanged)
   sub M subtract M from contents of accumulator (M unchanged)
   goto L go to instruction labeled L from contents of accumulator (M unchanged)
   ```

   4. Code:

   ```plaintext
   get
   store TEMP
   mul TEMP
   print
   mul TEMP
   print
   stop
   TEMP 2
   ```

   Code:

   Version 1:
   ```plaintext
   get
   Top print
   ifzero Done
   sub 2
   ```
ifpos Top
Done stop

Version 2:
get
print
CALC sub 2
ifpos OUT
stop
OUT print
goto CALC