Q1

(a)

Number of Cache Blocks(CB): 1K bytes/64 = 1024/64 = 16CB

Number of Memory Blocks(MB) 1G bytes/64 = $2^{30}/2^6 = 2^{24}$MB = 16 777 216MB

1) direct mapping:
   Block: $16 = 2^4$, so it requires 4 bits
   Word: 64 bytes per block, 4 bytes per word, so in each block, there are 64/4 = 16 words. $16 = 2^4$, so it requires 4 bits
   Byte: $4 = 2^2$, so it requires 2 bits
   Tag: 16 CB will map to 2^{24} MB, so each CB will map to $2^{24}/16 = 2^{20}$ MB, so it requires 20 bits
2) associative mapping:
   Tag: There are 2^{24}MB, so it requires 24 bits
   Word: Same as above, it requires 4 bits
   Byte: Same as above, it requires 2 bits
3) two-way set-associative mapping:
   Set: $16/2 = 8 = 2^3$, so it requires 3 bits
   Tag: 8 sets of CB will map to 2^{24} MB, so each set of CB will map to $2^{24}/8 = 2^{21}$ MB, so it requires 21 bits
   Word: Same as above, it requires 4 bits
   Byte: Same as above, it requires 2 bits

(b)

320 consecutive words = 320/16 = 20 MB

1) direct mapping:

Firstly, we load the word at memory location 0, which is a cache miss. However, we will load the whole memory block into the cache block, so words from memory locations 0 to 15 will load to the cache block. Therefore, loading words from memory locations 1 to 15 are cache hit.

Therefore, we can derive there are a total of 20 cache misses.(0,1,...,19)

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
(MB **16 17 18 19** 4 5 6 7 8 9 10 11 12 13 14 15)

Then, we repeat the process one more time. When we load the word at memory location 0, the CB should contain the data from memory locations 4 to 19. So, it is a cache miss.

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
(MB **0 17 18 19** 4 5 6 7 8 9 10 11 12 13 14 15)
...
(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
(MB **0 1 2 3** 4 5 6 7 8 9 10 11 12 13 14 15)

However, when we load words in memory location 4*16 = 64 (in CB 4) again, there will be a cache hit.

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
(MB **0 1 2 3** 4 5 6 7 8 9 10 11 12 13 14 15)
…
(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
(MB **16 17 18 19** 4 5 6 7 8 9 10 11 12 13 14 15)

Therefore, we can derive there are a total of 8 cache misses. (0,1,2,3,16,17,18,19)

Cache hit rate = 640-28 / 640 = 0.95625

2) <u>associative mapping (with the LRU replacement algorithm):</u>

Firstly, in the first loop, there are a total of 20 cache misses. (0,1,...,19)

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 16 17 18 19 4 5 6 7 8 9 10 11 12 13 14 15)

Then, in the second loop, there are also a total of 20 cache misses. (0,1,...,19)

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 16 17 18 19 0 1 2 3 4 5 6 7 8 9 10 11)

...

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 12 13 14 15 16 17 18 19 4 5 6 7 8 9 10 11)

Cache hit rate = 640-40 / 640 = 0.9375

3) <u>two-way set-associative mapping (with the LRU replacement algorithm):</u>

Firstly, we have 20 cache misses. (0,1,...,19)

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 0 1 2 3 4 5 6 7)

...

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 0 8 1 9 2 10 3 11 4 12 5 13 6 14 7 15)

…

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 16 8 17 9 18 10 19 11 4 12 5 13 6 14 7 15)

Then, in the second loop, we have 4 cache misses. (0,1,2,3) The MB 4 is already in the cache, it will be a cache hit.

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 16 **0** 17 **1** 18 **2** 19 **3** 4 12 5 13 6 14 7 15)

Next, we have 4 cache misses. (8,9,10,11)

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB **8** 0 **9** 1 **10** 2 **11** 3 4 12 5 13 6 14 7 15)

Also, we have 4 cache misses. (16,17,18,19)

(CB 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(MB 8 **16** 9 **17** 10 **18** 11 **19** 4 12 5 13 6 14 7 15)

Therefore, we can derive there are a total of 12 cache misses in the second loop. (0,1,2,3,8,9,10,11,16,17,18,19)

Cache hit rate = 640-32 / 640 = 0.95

Q2

(a) L1: hit rate = 0.95, so miss rate = 1 - 0.95 = 0.05
L2: hit rate = 0.8, so miss rate = 1 - 0.8 = 0.2
If L1 and L2 are both miss, miss rate = 0.05 * 0.2 = 0.01

(b) h_1: L1 cache hit rate = 0.95
h_2: L2 cache hit rate = 0.8
C_1: access time to L1 cache = τ
C_2: access time to L2 cache = 10τ
M_{L1}: miss penalty of L1 miss and L2 hit = $C_1 + C_2 + C_1 = 12\tau$
M_{L2}: miss penalty of L1 miss and L2 miss = $C_1 + C_2 + Mem + C_2 + C_1 = 122\tau$
average memory access time
$= h_1 \times C_1 + (1 - h_1) \times [h_2 \times M_{L1} + (1 - h_2) \times M_{L2}]$
$= 0.95 \times \tau + 0.05 \times [0.8 \times 12\tau + 0.2 \times 122\tau]$
$= 2.65\tau$

(c) Let h be the hit rate of L1 cache required
$h \times \tau + (1 - h) \times [0.8 \times 12\tau + 0.2 \times 122\tau] = 1.5\tau$
$h\tau + (1 - h)34\tau = 1.5\tau$
$h + 34 - 34h = 1.5$
$33h = 32.5$
$h = 65/66$

Let k be the hit rate of L2 cache required
$0.95\tau + 0.05 \times [k \times 12\tau + (1 - k) \times 122\tau] = 1.5\tau$
$[k \times 12\tau + (1 - k) \times 122\tau] = 11\tau$
$12k + 122 - 122k = 11$
$110k = 111$
$k = 111/110 > 1$, so impossible