1. (a) The L1 data cache of the AMD Barcelona is a 64KB, 2-way set- associative cache with 64-byte blocks. The Barcelona supports 40-bit physical addresses (i.e., can address 1TB of memory).

   Compute the number of sets and the size of the tag, index, and block offset fields.

   Solution:

   Recall from lecture that if the address has m bits and the cache has 2^s sets with blocks of 2^n bytes, then the address splits up as follows:

Tag = m-s-n	*Index = s*	*Block Offset = n*

   The cache is 2-way set associative, and each block is $64 = 2^6$ bytes. So $n = 6$. Thus each set is $2 \cdot 2^6 = 2^7$ bytes. Now the total cache size is $64KB = 2^{16}$. So the number of sets is $2^{16}/2^7 = 2^9 = 512$ sets. That is, $s = 9$. Putting this all together:

$Tag = m - s - n = 40 - 9 - 6 = 25$	$Index = s = 9$	$Block\ Offset = n = 6$

   (b) The L2 data cache of the Intel Core 2 Duo is a 2MB, 8-way set- associative cache with 64-byte blocks. The Core 2 Duo supports 36- bit physical addresses (i.e., can address 64GB of memory).

   Compute the number of sets and the size of the tag, index, and block offset fields.

   Solution:

   The cache is 8-way set associative, and each block is $64 = 2^6$ bytes. So $n = 6$. Thus each set is $2^3 \cdot 2^6 = 2^9$ bytes. Now the total cache size is $2MB = 2^{21}$. So the number of sets is $2^{21}/2^9 = 2^{12} = 4096$ sets. That is, $s = 12$. Putting this all together:

$Tag = m - s - n = 36 - 12 - 6 = 18$	$Index = s = 12$	$Block\ Offset = n = 6$

2. (a) Given a *direct-mapped* cache with 2 blocks of 2 bytes each, where the address is broken as follows: If the cache was initially empty and the addresses below were accessed

Tag	Index	Offset

in that order, which of the following would be cache misses?
0110 0010 0110 1100 0111 1001 0110 1100 1101
Solution:
1, 2, 3, 4, 6, 8 miss.

Access #5 is a hit because this data was loaded along with 0110 in access #3.

Access #7 is a hit because this data was loaded on a miss in access #3.

Access #9 is a hit because this data was loaded along with 1100 in access #8.

(b) Given a fully associative cache (using a least-recently-used replacement policy) with 3 blocks of 2 bytes each, where the address is broken as follows:

Tag	Offset

If the cache was initially empty and the addresses below were accessed in that order, which of the following would be cache misses?
0110 1101 0111 1001 1110 1100 0110 1111 1101
Solution:
1, 2, 4, 5, 6, 7 miss
Access #3 is a hit because this data was loaded along with 0110 in access #1.

Access #8 is a hit because this data was loaded along with 1110 in access #5.

Access #9 is a hit because this data was loaded along with 1100 in access #6.

3. Give an access sequence for which Cache 1 has strictly more hits than Cache 2, or argue that such a sequence cannot exist.
   Solution:
   Consider the following sequence: 1101, 0010, 1010, 1110, 0110, 1101.

Address	*Cache 1*	*Cache 2*
1101	*Miss*	*Miss*
0010	*Miss*	*Miss*
1010	*Miss*	*Miss*
1110	*Miss*	*Miss*
0110	*Miss*	*Miss*
1101	*Hit*	*Miss*

4. For this question, you are given a 16-byte cache (initially empty) and the sequence of memory accesses (byte loads) shown in the table. For each of the following cache configurations explain whether it can produce this sequence.

address	hit/miss
2	miss
14	miss
0	hit
17	miss
12	hit
3	hit

(a) 4-byte blocks, 2-way set-associative (e.g., 2 sets of 2 blocks each)
Solution:
Yes, here are the final cache contents:

0-3	16-19
12-15	

(b) 4-byte blocks, direct-mapped (4 blocks)
Solution:
No, 3 should miss. Here is the cache before the miss:

16-19
12-15

(c) 8-byte blocks, direct-mapped (2 blocks)
Solution
No, 3 should miss. Here is the cache before the miss:

16-23
12-15

(d) 4-byte blocks, fully associative (4 blocks)
Solution:
Yes, here are the final cache contents:

0-3	12-15	16-19	