You may work in pairs for this assignment. If you choose to work with a partner, make sure only one of you submits a solution, and that the file lists names and PIDs for both of you at the beginning.

Prepare your answers to the following questions in a plain text file. Submit your file to the Curator system by the posted deadline for this assignment. No late submissions will be accepted. For all questions, show supporting work if you want partial credit.

You will submit your answers to the Curator System (www.cs.vt.edu/curator) under the heading `MIPS03`.

1. Consider two caches. Both are 16 cache lines in size, each cache line is 16 bytes and both start out with all lines marked invalid. The only difference is one is two-way associative and another is direct-mapped. Assume DRAM uses 16 bit addresses.

a) [8 points] Find a shortest possible reference stream of addresses where the two-way associative cache would get a hit and the direct-mapped cache would get no hits. Provide addresses of the reference stream in hex.

b) [8 points] Find a shortest possible reference stream of addresses where the direct-mapped cache would get a hit and the two-way associative cache would get no hits. Provide addresses of the reference stream in hex.

2. Consider the following access pattern: *A*, *B*, *C*, *A*. Assume that *A*, *B*, and *C* are memory addresses each of which are in a different block of memory. Suppose *A*, *B*, and *C* are generated in a uniformly random way and that a LRU (least recently used) replacement algorithm is used. LRU replacement means that if we have to choose a block to replace, we replace the one (in the set) that has not been accessed for the longest time; ties are resolved by making a random choice. Further, assume that any given block has an equal chance of being placed in either "way". What is the probability that the second instance of *A* will be a hit if:

a) [6 points] The cache has 2 lines and is fully-associative.

b) [6 points] The cache has 4 lines and is fully-associative.

c) [6 points] The cache has 4 lines and is direct-mapped.

d) [6 points] The cache has 4 lines and is two way associative.

3. [8 points] A system has an (unified) L1 cache and an L2 cache. What is the AMAT (in cycles) for the system with the following parameters?

L1 hit time = 2 cycles
L1 miss rate = 10%
L2 hit time = 5 cycles
L2 miss rate = 5%
DRAM access time = 100 cycles

4. [8 points] A system has a separate L1 cache for instructions (L1i), a L1 cache for data (L1d), and an unified L2 cache. Assume both L1i and L1d access time is 2 clock cycles; the L2 access time is 10 clock cycles; and the DRAM access time is 100 clock cycles. The L1i cache miss rate is 4%; the L1d cache miss rate is 10%; and the L2 cache miss rate is 1%. On average, 30% of instructions are loads or stores. The base CPI, assuming ideal cache performance, is 3.

What is the actual average CPI for this system?

5. A system uses 32-bit addresses, and a direct-mapped cache. Byte offsets are determined using bits 4-0, set numbers using bits 9-5, and tags using bits 31-10. Suppose the following byte-addressed cache references are recorded.

0, 24, 6, 150, 238, 172, 1024, 24, 150, 3100, 190, 2200

a) [8 points] How many bytes of user data can the cache hold? Express your answers in terms of powers of 2

b) [8 points] How many blocks are replaced?

c) [8 points] What is the hit ratio?

6. [20 points] Consider a two-dimensional, N x N array of words A. This array is laid out in memory so that A[0][0] is next to A[0][1], and so on; A[0][N-1] is next to A[1][0], and so on (i.e., in row-major order). Assume that the cache is initially empty, but that A[0][0] maps to the first word of cache line 0.

Consider the following matrix transpose algorithm:

```
int tmp = 0;
for (int i = 0; i < N; i++) {
    for (int j = i+1; j < N; j++) {
        tmp = A[ i ][ j ];
        A[ i ][ j ] = A[ j ][ i ];
        A[ j ][ i ] = tmp;
    }
}
```

For the following questions, assume that the variable tmp is register allocated (i.e., an access to tmp does not trigger a memory access, and thus has no cache effects); assume that the blocks of the array A are mapped into the cache so that block 0 goes into cache set 0, block 1 goes into cache set 1, and so forth.

Suppose the algorithm is executed with a 4 x 4 matrix (4 rows, 4 columns); and suppose we have a direct-mapped cache with 4 sets, indexed 0 to 3, where each cache block encompasses 2 words (i.e., 2 integer values).

For each memory access, find 1) its type (read or write), 2) accessed word, 3) whether it result in cache hit or miss, 4) newly brought cache block (on a miss), and 5) replaced cache block (if a valid block exists on a miss) similar to the following format. Note that the answer below is NOT correct for this question.

R/W	Accessed word	Cache hit/miss	Newly brought cache block	Replaced cache block
R	A[0][1]	miss	A[0][0], A[0][1]	
R	A[0][1]	hit		
W	A[2][0]	miss	A[2][0], A[2][1]	A[0][0], A[0][1]
…	…	…	…	…