Homework#5 for ECE 496
Memory Hierarchies, Caches, Memory (Chapter 5)

Hardcopy is due in class on Tuesday, April 16
Problem 5 must be submitted electronically by 11:55pm on April 16

1) [10 points] Patterson & Hennessy 5.10.1

2) [5] P&H 5.10.4b

3) [10] P&H 5.10.5b

4) [5] P&H 5.11.1b

5) [**100 points**] Write a simulator of a single-level cache, using the high level language of your choice (Java, C++, C). The simulator, called cachesim, takes the following input parameters on the command line: name of the file holding the loads and stores, cache size (not including tags or valid bits) in KB, associativity, and the block size in bytes. The replacement policy is always LRU. For example, "cachesim tracefile 32 4 8" should simulate a cache that is 32KB, 4-way set-associative, has 8-byte blocks, and uses LRU replacement. This cache will be processing the loads and stores in the file called tracefile.

Assumptions: Addresses are 32-bits. The cache size, associativity, block size, and access size will all be powers of 2. No cache access will span multiple blocks (i.e., each cache access fits within a single block). All cache blocks are initially invalid. All cache misses are satisfied by the main memory (and you must track the values written through to memory in case they are subsequently loaded). If a block has never been written before, then its value in main memory is zero. The cache is write-back and write-allocate.

The trace file will be in the following format. There will be some number of lines. Each line will specify a single load or store, the address that is being accessed (in Hex), the size of the access in bytes, and the value to be written if the access is a store (in Hex). For example:

```
store 0x1234ab00 2 19ab
load 0x002a173f 4
```

Your simulator must produce the following output. For each access, it must print out whether it is a hit or a miss. For each load, it must print out the value (in Hex) that is loaded (possibly after satisfying the miss from memory). The output format must be as follows (to help the grader):

```
store miss
load hit 7d228f13
```

Submit this assignment on Sakai. **Please follow the output specs exactly, as an autograder will be used to test your program.** Explain in your README file how to build the simulator, e.g., "g++ cachesim.cpp -o cachesim". If you have any known remaining bugs, please also include those in the README file.