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 you paste a copy of the Partners Template that contains the names and PIDs of both students at the beginning of the file.

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. To receive credit, you must show your computations, and explicitly apply Amdahl's Law where specified.

You will submit your answers to the Curator System (www.cs.vt.edu/curator) under the heading `MIPS04`.

1. Suppose that, in addition to registers in the CPU, a system includes a unified L1 cache, and DRAM, and that the storage levels have the average access times shown below:

L1 cache	1 ns
DRAM	50 ns

The system has a 2GHz clock rate. Also, the system executes machine instructions using a 5-stage pipeline, and the average CPI is 6.5, assuming that there are no memory accesses. (This would take into account the effect of stalls for forwarding and branch decisions, and include the fetch if it's from the L1 cache.)

For the following parts of this question, round time values to the nearest tenth of a nanosecond (e.g., 4.6), and round CPI values to the nearest hundredth of a cycle (e.g., 7.23). Show your computations.

a) **[4 points]** For this part, assume that the hit rate for the L1 cache is (an unlikely) 100%. What is the average memory access time AMAT (in ns)?

b) **[4 points]** Again assuming an L1 cache hit rate of 100%, what would be the average CPI if 20% of the executed instructions require a memory access (besides being fetched)?

c) **[4 points]** Now, suppose the L1 cache hit rate is 97%. What is the average memory access time AMAT now (in ns)?

d) **[4 points]** Under the same assumptions as part **c)**, and assuming that 20% of the executed instructions require a memory access, what would be the average CPI?

e) **[4 points]** Repeat part **d)**, but now assume that 40% of the executed instructions require a memory access.

2. Suppose that, in addition to registers in the CPU, a system includes a unified L1 cache, a unified L2 cache, and DRAM, and that the storage levels have the average access times shown below:

L1 cache	1 ns
L2 cache	10 ns
DRAM	50 ns

The system has a 2GHz clock rate. Also, the system executes machine instructions using a 5-stage pipeline, and the average CPI is 6.5, assuming that there are no memory accesses. (This would take into account the effect of stalls for forwarding and branch decisions, and include the fetch if it's from the L1 cache.)

For the following parts of this question, round time values to the nearest tenth of a nanosecond (e.g., 4.6), and round CPI values to the nearest hundredth of a cycle (e.g., 7.23). Show your computations.

a) **[8 points]** Now, suppose the L1 cache hit rate is 97%, and the L2 cache hit rate is 99%. What is the average memory access time AMAT now (in ns)?

b) **[8 points]** Under the same assumptions as part **a)**, and assuming that 40% of the executed instructions require a memory access (besides being fetched), what would be the average CPI?

3. A system uses 24-bit DRAM addresses and an 8-way set associative cache with 128 sets, with a block size of 32 bytes. The cache sets are, of course, numbered 0 to 127.

a) [4 points] The *capacity* of a cache is the amount of user data the cache can store (so this excludes valid bits, tags. etc.) What is the capacity of this cache? Give your answer in bytes and in KB (kilobytes).

b) [4 points] How many tag comparator circuits would this cache need?

For parts **c)** through **g)**, assume we are loading the block of data from DRAM that corresponds to the address shown below:

0	1	0	1	1	0	0	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	1	1
b_{23}	b_{22}	b_{21}	b_{20}	b_{19}	b_{18}	b_{17}	b_{16}	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

c) [4 points] Which cache set will the block of data be stored in? Give your answer in binary, and in base-10.

d) [4 points] What value will be stored in the tag field of the line containing this block? Give your answer in binary, and in hex.

Be careful. Your understanding of the two previous questions is needed in the remaining parts.

For each of the following parts, suppose that the block corresponding to the address above has been loaded into the cache, and that the cache does not contain any other data blocks. For parts **e)** through **g)**, determine whether an access to each of the addresses below would result in a cache hit or a cache miss. Justify your answers.

Be careful when comparing the addresses below to the address given above.

e) [2 points]

0	1	0	1	1	0	0	0	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0
b_{23}	b_{22}	b_{21}	b_{20}	b_{19}	b_{18}	b_{17}	b_{16}	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

f) [2 points]

0	1	0	1	1	0	0	0	1	1	0	1	1	1	1	0	1	1	0	1	1	0	1	1
b_{23}	b_{22}	b_{21}	b_{20}	b_{19}	b_{18}	b_{17}	b_{16}	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

g) [2 points]

0	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	1	1
b_{23}	b_{22}	b_{21}	b_{20}	b_{19}	b_{18}	b_{17}	b_{16}	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

4. Suppose the C function shown below is used to perform searches on an array storing 2^{20} 64-bit integer values:

```
int32_t countMatches(const int64_t list, int32_t nElements,
                                         int64_t target) {

   int32_t nMatches = 0;

   for (int32_t idx = 0; idx < nElements; idx++) {

      if ( target == list[idx] ) {
          nMatches++;
      }
   }
   return nMatches;
}
```

a) **[8 points]** Suppose this code is executed on hardware that includes a single-level direct-mapped cache, where each cache line uses a block size of 8 bytes, and the total cache capacity is 64KB. Assume that each local automatic variable in the function is stored in a register. What will be the cache hit rate for accesses to the elements in the array? Justify your answer.

b) **[8 points]** Now suppose a single-level direct-mapped cache is used, but the block size is 128 bytes and the total cache capacity is still 64KB. Will the cache hit rate for accesses to elements in the array change? If so, what will the hit rate be. Explain why. If not, explain why the hit rate will not change.

5. **[10 points]** Suppose the C function shown below is used to perform searches on a sorted array storing 2^{20} 64-bit integer values:

```
int32_t findMatches(const int64_t list, int32_t nElements,
                                        int64_t target) {

   int32_t loIdx = 0;
   int32_t hiIdx = nElements;

   while ( loIdx <= hiIdx ) {

      int32_t midIdx = (loIdx + hiIdx) / 2;

      if ( target < list[midIdx] ) {
         hiIdx = midIdx - 1;
      }
      else if ( target > list[midIdx] ) {
         loIdx = midIdx + 1;
      }
      else {
         return midIdx;
      }
   }
   return -1;
}
```

Suppose that the system uses a single-level direct-mapped cache with a total capacity of 64KB. Will the size of the cache blocks have any effect on the cache hit rate for array elements? Explain why or why not. (Be careful, there is a subtle issue here.)

6. Consider two caches. Each cache has 16 cache lines in size, each cache uses 32-byte cache blocks, and each cache starts out with all lines marked invalid. The only difference is one cache is two-way set associative and the other 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 at least one hit and the direct-mapped cache would get no hits. Provide addresses of the reference stream in binary and 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 binary in hex.