CS 203 (2021 Fall) Assignment #2

Student ID #:

Name:

Who else you discussed with when finishing the assignment:

(While you may have your partner do all the work, this will only hurt you when the midterm and final come around so don't do it.)

Please make sure that you explain how you derive each answer clearly. Simply equations with numbers or giving the resulting numbers/graphs will not satisfy the grading rubrics for full credits.

1. Consider the following matrix transpose code

```
int i, j ,k;
double *A, *B, *C;
A = (double *)malloc(sizeof(double)*N*N);
B = (double *)malloc(sizeof(double)*N*N);
init_data(A, N*N);
for(i = 0; i < N; i++)
    for(j = 0; j < N; j++)
              B[i*N+j] = A[j*N+i];
// assume load A[j*N+i] and then store B[i*N+j]
output_data(B, N*N);
```

Assume that the starting address of array A is 0x20000 and array B is 0x40000. Assume N = 128. Assuming that you have an intel Core i7 processor that has a 32K, 8-way, 64-byte blocked L1 data blocked L1 data cache. Please answer the following questions:

(1) How many of the misses within the two-level nested for loop are compulsory misses? How many of them are conflict misses?

(2) Continued from the previous question, what’s the overall miss rate for the nested for-loop?

2. Smith and Goodman found that for a given small size, a direct-mapped instruction cache consistently outperformed a fully associative instruction cache using LRU replacement.

   (1) Explain how this would be possible. (Hint: You can't explain this using the three C's (compulsory, capacity, conflict) model of cache misses because it "ignores" the cache policy.)

   (2) Explain where replacement policy fits into the three C's model, and explain why this means that misses caused by replacement policy are "ignored"--or, more precisely, cannot in general be definitively classified--by the three C's model.

(3) Are there any replacement policies for the fully-associative cache that would outperform the direct-mapped cache? Ignore the policy of "do what a direct mapped cache would do."

3. You are building a system around a single-issue in-order processor running at 2 GHz and the processor has a base CPI of 1 if all memory accesses are hits. The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions). The memory system for this computer is composed of a split L1 cache that imposes no penalty on hits. Both the I-cache and D-cache are direct mapped and hold 32KB each. You may assume the caches use write-allocate and write-back policies. The L1 I-cache has a 2% miss rate and the L1 D-cache has a 5% miss rate. Also, 50% of all blocks replaced from L1 D-cache are dirty. The 512KB write-back, unified L2 cache has an access time of 12ns. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also 25% of all blocks replaced are dirty. The main memory has an access latency of 60ns. What is the overall CPI, including memory accesses?