1. Draw diagrams showing shared caching and private caching for a quad core system. Explain the difference between them.
2. Explain the cache coherence problem for private caches and why multicore systems need to solve it.
3. Define false sharing.
4. For each of the code segments below, explain whether false sharing is likely to occur and any steps that could be taken to reduce or eliminate it. (Note that neither “yes” nor “no” is an explanation.) Recall that when OpenMP runs a sections block, each section sunblock is assigned to a single thread. For all the code segments, assume that `array` and `count` are pre-allocated integer arrays sufficiently long for the way they are used.

```
a. #pragma omp parallel sections
   {
        #pragma omp section
        {
             for(int i=0; i<num/2; i++)
                  if(array[i] == 0)
                       count[i/5]++;
        }

        #pragma omp section
        {
             for(int i=num/2; i<num; i++)
                  if(array[i] == 0)
                       count [i/5]++;
        }
   }
b. #pragma omp parallel sections
   {
        #pragma omp section
        {
             for(int i=0; i<num; i+=2)
                  if(array[i] < 0)
                       count[0]++;
        }

        #pragma omp section
        {
             for(int i=1; i<num; i+=2)
                  if(array[i] < 0)
```

```
                       count[1]++;
        }
   }
c. #pragma omp parallel sections
   {
        #pragma omp section
        {
             int ones = 0;
             for(int i=0; i<num; i++)
                  if(array[i] == 1)
                       ones++;
             printf(“Number of ones: %d\n”, ones);
        }

        #pragma omp section
        {
             int twos = 0;
             for(int i=0; i<num; i++)
                  if(array[i] == 2)
                       twos++;
             printf(“Number of twos: %d\n”, twos);
        }
   }
```