**Question 1:**
Consider the MSI using snoopy protocol for a bus-based system with three processors P₁, P₂ and P₃, each with a direct-mapped cache and assume that the size of a cache line is one word. The following sequence of memory operations access two memory locations (words), A and B, that are mapped to the same cache location:

P₁ writes A = 4
P₃ writes B = 2
P₂ reads A
P₃ reads A
P₃ writes A = 12
P₂ reads A
P₁ reads B
P₁ writes B = 10

Assume that initially, A = B = 3 and that the cache is initially empty. For each memory access, determine:
(a) The content and state of the cache line in each processor (modified, shared, invalid).
(b) The bus operations caused by the MSI protocol (read request, read data, write back).

You can use the table found in the slides as an example of how best to represent this answer.

**Question 2:**
Consider a directory-based coherence protocol for a distributed shared memory system, in which each data block has a “Home” processor. Along with each block, the home processor keeps a directory entry that specifies one of the following states for the block:
- Uncached: no processor has it (invalid in all caches)
- Shared: cached in one or more processors (blocks are clean where cached)
- Exclusive: one processor (owner) has the block (block is dirty in cache)

Assume that a processor, P, is the home for blocks a, b and c and that the directory indicates the following states for the four blocks:
- Block a: Shared by processors Q and R
- Block b: Exclusive with Q as the owner
- Block c: Shared a located at R

Assume also that the cache in each processor uses direct mapping (no associativity) and that both blocks c and b are mapped to the same cache entry (has the same index).
In each of the following two cases, indicate the sequence of protocol messages that should be exchanged and the final state of the blocks involved in the operation in the caches of R and Q. Use an informal format of the form “R sends a message to P to do so and so” and ignore “acknowledgment” messages.

(a) Processor R issues a “write” into block $a$
(b) Processor R issues a “read” into block $b$