1. For each assertion A1..A3, B1..B3, C1, C2, indicate whether it is true or false. If it is false, give the assertion which is true at that point. You must do this assuming synchronous communication. You do not have to provide proofs.

\[
\begin{align*}
\text{P}_1: & \quad x = 1 \\
\text{P}_2: & \quad \text{P}_2 \text{! } x + 1 \\
\text{P}_3: & \quad \{ \text{A1: } x = 2 \} \\
\text{A2: } x = 3 & \\
\text{P}_1: & \quad \text{P}_1 \text{? } y \\
\text{P}_2: & \quad \{ \text{B1: } y = 2 \} \\
\text{B2: } y = 3 & \\
\text{P}_3: & \quad \text{P}_3 \text{! } y + 1 \\
\text{C1: } z = 2 & \\
\text{C2: } z = 3 & \wedge \ x = 4 \\
x = x + 1 & \\
y = y + 1 & \\
z & = 1
\end{align*}
\]

2. Assume that variables x, y and z are assigned some initial values at processes P1, P2 and P3 respectively. It is claimed that the following program calculate the sum of x, y and z, and at the end, each of x, y and z contain the sum of their initial values. Is the claim correct? If not, correct the program to compute the sum. Give reasons for your answer in either case. Assume synchronous communication.

P1: step1 = 0
\[
\text{do} \\
\text{[} \text{step1 = 0, P2? x1 } \rightarrow x = x + x1; \text{ P3 ! x; step1++} \text{]} \\
\text{[} \text{step1 = 1, P3 ? x } \rightarrow \text{step1++} \text{]} \\
\text{[} \text{step1 = 0, P2 ! x } \rightarrow \text{skip} \text{]} \\
\text{od}
\]

P2: step2 = 0
\[
\text{do} \\
\text{[} \text{step2 = 0, P1 ? y1 } \rightarrow y = y + y1; \text{ P3 ! y; step2++} \text{]} \\
\text{[} \text{step2 = 1, P3 ? y } \rightarrow \text{step2++} \text{]} \\
\text{[} \text{step2 = 0, P1 ! y } \rightarrow \text{skip} \text{]} \\
\text{od}
\]

P3: step3 = 0
\[
\text{do} \\
\text{[} \text{step3 < 2, P1 ? z1 } \rightarrow z = z + z1; \text{ step3++;} \text{]} \\
\text{[} \text{step3 < 2, P2 ? z1 } \rightarrow z = z + z1; \text{ step3++;} \text{]} \\
\text{[} \text{step3 = 2 } \rightarrow \text{P1 ! z; P2 ! z} \text{]} \\
\text{od}
\]

3. Do the following executions satisfy sequential consistency

a. P1: w1(y)2, r1(x)1
   P2: w2(x)2, r2(z)1
   P3: w3(z)2, r3(y)1

b. P1: r1(x)1, w1(y)2, r1(x)2
   P2: r2(y)1, w2(z)3, w2(x)2
   P3: r3(y)2, r3(z)3, r3(x)1

All variables are initialized to 1. Give reasons for your answer.
4. Assume that you have three processes, A, B and C and two variables x and y. Process A writes x, B reads x and writes y and C only reads x and y. Each process may perform the operations above multiple times. Each process which reads a variable maintains a local copy. When A writes x, it sends the updated value to both B and C, who update their local copies. When B writes y, it sends the updated value to C which updates its local copy. Whenever B or C wants to perform a read, they simply read their local copies. Does this implementation ensure sequential consistency? If not, present a scenario that violates sequential consistency.

5. In the class, we studied the dining philosophers problem using message passing. Write a solution to the same problem assuming shared memory. Assume that you can only hold lock on at most one variable at any given time.