1. (20 points.) Redo problem #1 of HW #1 by defining and solving a Markov model using sharpe. Submit the sharpe code and program output.

(Hint: Draw a Markov model to describe the M/M/3/10 system in HW #1. Use the state representation \((f, s, q)\) with \(f\) representing the number of clients (0 or 1) being served by the faster server, \(s\) representing the number of clients (0, 1, or 2) being served by the two slower servers, and \(q\) representing the number of clients (0, 1, 2, 3, 4, 5, 6, or 7) waiting to be served in the queue. The initial state is \((0,0,0)\). Your answers should be the same or close to the ones in HW #1.)

2. (20 points.) Consider a system with 2 CPUs and 2 memory modules that requires at least 1 CPU and 1 memory module to be functioning for the system to be functioning. Use SHARPE to calculate the system reliability \(R(t)\) at \(t=500\) hours. Assume that the MTTFs of a CPU and a memory module are 1000 and 1200 hours, respectively. The MTTRs for a CPU and a memory module are 4 and 2 hours, respectively. Show the SHARPE program listing and output. Consider the following three cases separately. For each case, you need to compute \(R(t)\) at \(t=500\) hours. (Hint: define Markov models that allow repairs to occur only when the system is still alive.)

(a) Each component has an independent repair facility.
(b) Each subsystem (CPU or memory) has an independent repair facility that can repair failed components within the subsystem one at a time.
(c) The whole system shares a repair facility which repairs failed components one at a time with the repair priority of CPUs over memory modules.

3. (20 points.) Suppose that a home multimedia server has \(n = 3\) slots to accommodate incoming high and low priority clients, with arrival rates of \(\lambda_h\) and \(\lambda_l\) and departure rates of \(\mu_h\) and \(\mu_l\), respectively. A high priority client must always occupy one full slot. A low priority client, on the other hand, can lower its quality of service (QoS) by occupying only one half of a slot, if necessary. When a low priority client occupies a full slot, we call it a low priority, high QoS client; when it occupies one half of a slot, we call it a low priority, low QoS client. Draw a Markov state transition diagram for modeling the following resource control policy:

(i) If there is at least one full slot available, an incoming client always occupies an empty full slot, regardless of its priority class.
(ii) If case (i) is not true, an incoming high priority client can lower the QoS of two low priority, high QoS clients, if they are available, after which the high priority client occupies one full slot and the two low priority, high QoS clients both become low priority, low QoS clients, with each occupying one half slot.
(iii) If case (i) is not true, an incoming low priority client can lower the QoS of one low priority, high QoS client, if it is available, after which each occupies one half slot.
(iv) A client will turn away if none of the above cases is applicable.
(v) A low priority, low QoS client occupying one half slot can immediately become a low priority, high QoS client occupying one full slot upon a client’s departure.

Use the representation \((a, b, c)\) where \(a\) stands for the number of low-priority, low QoS clients, \(b\) stands for the number of low-priority, high QoS clients, and \(c\) stands for the number of high-priority clients. Organize the Markov model so that when a high priority client arrives, the transition goes right; and when a low priority client arrives, the transition goes down. Label the transition rate of each transition clearly.

Assuming \(\lambda_h = \mu_h = 3\) and \(\lambda_l = \mu_l = 4\), write Sharpe code to obtain:

(a) the rejection probability of high-priority clients;
(b) the average number of high-priority clients in the system;
(c) the average waiting time of a high-priority client. (Hint: the average waiting time is equal to the average 
response time minus the average service time.)
(d) the throughput of low-priority, low QoS clients.

4. (20 points.) Use the MVA solution technique to hand-calculate the average response time per terminal user 
one it enters the central system for \( N = 4 \) for the QNM shown on page 144 of the lecture slide. (Hint: 
treat the terminal center as a center with infinite servers in the closed QNM and calculate the visit ratios 
first based on routing probabilities. Compare the average response time per user hand-calculated with that 
generated by Sharpe code. Note: the terminal service time per user is 25 sec, not 25 msec.)