1. (20 points.) Write a smpl simulation program for an M/M/3/8 queueing system to compute
   the (1) customer turned-away probability, (2) average number of customers waiting in the
   system (excluding the ones in service), (3) response time per client and (4) throughput,
   assuming that the arrival rate $\lambda$ is equal to the to service rate $\mu = 5$ customers/second. Make
   sure that the reported response time is within 95% level of confidence with 5% confidence
   accuracy based on the batch mean analysis method. Also report the confidence accuracy
   obtained from your simulation program. Turn in the smpl program source code and output
   in a hardcopy.

   An example output is:

   batch 1 mean = 0.213
   batch 2 mean = 0.199
   batch 3 mean = 0.217
   batch 4 mean = 0.211
   batch 5 mean = 0.209
   batch 6 mean = 0.219
   batch 7 mean = 0.211
   batch 8 mean = 0.195
   batch 9 mean = 0.208
   batch 10 mean = 0.214, rel. HW = 0.025
   (1)Customer turned-away probability is 0.000148
   (2)Average number of customers waiting is 0.045147
   (3)Mean Response Time is 0.209468 and half width is 0.005339
   Confidence Accuracy: 0.025486
   (4)Throughput is 5.067743

   This can be obtained from the following code:

   /***************************************************************************/
   /* hw1.c */
   /* This program simulates an M/M/3/8 system */
   /* Output: Customer Turned-away Probability*/
   /* Number of customers waiting */
   /* Response time */
   /* throughput */
   /***************************************************************************/

   #include "smpl.h"
#define TOKENS 1000
#define TRUE 1
#define FALSE 0

int main()
{
    real Ta=0.2,Ts=0.2,mean,hw;
    int tk_id=0,customer=0,event,server,nb, n, rejected=0, completed =0;
    real ts[TOKENS]; /* start time stamp */
    real TotalTimeElapsed, rejectProb, queueLenght, X;
    real CA;
    int cont=TRUE;
    int totalCustomersArrived = 0;
    smpl(0,"M/M/3/8 Queue");
    init_bm(200,2000); /* let m0 be 200 and mb be 2000 observations */
    server=facility("server",3);
    schedule(1,0.0,tk_id);
    while (cont)
    {
        cause(&event,&customer);
        switch(event)
        {
            case 1: /* arrival */
                totalCustomersArrived++;
                n = inq(server);
                if(n<5) /* there is still room to accept this client */
                {
                    ts[customer] = time();
                    schedule(2, 0.0, customer);
                }
                else { rejected++; }
                if (++tk_id >= TOKENS) tk_id=0;
                schedule(1, expntl(Ta), tk_id);
                break;
            case 2: /* request server */
                if (request(server,customer,0)==0) then
                    schedule(3,expntl(Ts),customer);
                break;
            case 3: /* release server */
                release(server,customer);
                if (obs(time()-ts[customer]) == 1) cont = FALSE;
                break;
        }
    }
    TotalTimeElapsed = time();
    rejectProb = rejected/(real)(totalCustomersArrived);
2. (10 points.) Assume that four components obeying the exponential failure law have failure rates of $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_4$, respectively; that is, $R_1(t) = e^{-\lambda_1 t}$, $R_2(t) = e^{-\lambda_2 t}$, $R_3(t) = e^{-\lambda_3 t}$, and $R_4(t) = e^{-\lambda_4 t}$. Derive a mathematical expression for the system reliability of a 3-out-of-4 system using these four components based on the minimal path set method. Note: you must use the minimal path set method covered in the lecture.

**Answer:**

Let $\phi(t)$, $X_1(t)$, $X_2(t)$, $X_3(t)$, and $X_4(t)$, be binary random variables representing the status of the system, component 1, component 2, component 3 and component 4, respectively, at time $t$ such that the value is 1 if the corresponding entity is alive and 0 otherwise.

Based on the minimal path set method, the system consists of four subsystems (1,2,3), (1,2,4), (1,3,4) and (2,3,4) connected in parallel. Therefore,

$$
\phi(t) = 1 - [(1 - (X_1X_2X_3))(1 - (X_1X_2X_4))(1 - (X_1X_3X_4))(1 - (X_2X_3X_4))]
$$

$$
= X_1X_2X_3 + X_1X_2X_4 + X_1X_3X_4 + X_2X_3X_4 - 3X_1X_2X_3X_4
$$

Consequently,

$$
E[\phi(t)] = R(t)
$$

$$
= E[X_1X_2X_3] + E[X_1X_2X_4] + E[X_1X_3X_4] + E[X_2X_3X_4] - 3E[X_1X_2X_3X_4]
$$

$$
$$

$$
= e^{-(\lambda_1+\lambda_2+\lambda_3)t} + e^{-(\lambda_1+\lambda_2+\lambda_4)t} + e^{-(\lambda_1+\lambda_3+\lambda_4)t} + e^{-(\lambda_2+\lambda_3+\lambda_4)t} - 3e^{-(\lambda_1+\lambda_2+\lambda_3+\lambda_4)t}
$$

3. (15 points.) Consider a fully-connected network topology shown above with 3 nodes (1, 2 and 3) and 3 links (a, b, and c). Two nodes can communicate with each other as long as there exists a communication path between them. For example, nodes 1 and 2 can communicate with each other via the following two communication paths: (a) link a only; and (b) a path consisting of links c and b passing through node 3. The system is designed such that a failed node can be bypassed without blocking any communication path in which it is an intermediate node. For example, if link a and node 3 fail, then nodes 1 and 2 can still...
communicate with each other by the path consisting of links c and b bypassing the failed node 3.

Assume that all nodes (all links) are indistinguishable. The failure and repair rates of each node are $\lambda_n = 0.0005 \ h^{-1}$ and $\mu_n = 0.005 \ h^{-1}$, respectively, while those for each link are $\lambda_l = 0.00001 \ h^{-1}$ and $\mu_l = 0.0001 \ h^{-1}$, respectively. The system requires that at least two nodes must be alive and be able to communicate with each other for the system to be operational. Use a fault tree model to compute the availability of the system at $t=2000$ hours.

(a) (10 points.) Show your fault tree model.

(b) (5 points.) Write a Sharpe code based on your fault tree model to compute the system availability.

Ans (a): The fault tree model below is based on the minimal path set method to deal with repeated components. An alternative solution is based on minimal cut set.

Ans (b):
poly abar(lambda, mu) gen\nlambda/(lambda+mu),0,0\n-lambda/(lambda+mu),0,-(lambda+mu)

ftree 3ring
* nodes are labeled with 1 2 3 and links are labeled with a, b and c
repeat 1 abar(lambdan,mun)
repeat 2 abar(lambdan,mun)
repeat 3 abar(lambdan,mun)
repeat a abar(lambdal,mul)
repeat b abar(lambdal,mul)
repeat c abar(lambdal,mul)
*enuerminating all minimal path sets in the 3-ring topology
or path1 1 2 a
or path2 2 3 b
or path3 3 1 c
or path4 1 2 b c
or path5 2 3 a c
or path6 3 1 a b
and top path1 path2 path3 path4 path5 path6
end

bind
lambdan 0.0005
mun 0.005
lambdal 0.00001
mul 0.0001
end

*print the availability at time=2000hr
expr 1-value(2000;3ring)

end