1. (20 points.) Consider an M/M/2/8 queuing system equipped with one fast server with the service rate of $\mu_1 = 6$ customers/sec and one slow server with the service rate of $\mu_2 = 3$ customers/sec. Assume that the customer arrival rate $\lambda$ is 15 customers/second. Write an smpl simulation program for this M/M/2/8 queueing system to compute (1) the customer turned-away probability, (2) average number of customers waiting in the system (excluding the ones in service), (3) response time per client for those served by the fast server, (4) response time per client for those served by the slow server, and (5) throughput. Make sure that the reported response time for those served by the fast server 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. Show the smpl program source code and output.

Smpl source code:

```c
#include "smpl.h"
#define TOKENS 1000
#define TRUE 1
#define FALSE 0
main()
{
    real Ta=1.0/15.0, Tfs=1.0/6.0, Tss=1.0/3.0, mean, hw;
    int tk_id = 0, customer = 0, event, mm28_facility, nb, n, rejected = 0;
    real totalCompleted = 0, fsCompleted = 0, ssCompleted = 0;
    real ts[TOKENS]; /* start time stamp */
    real TotalTimeElapsed, rejectProb, queueLength, X;
    real ssTotalResponseTime;
    int cont = TRUE;
    real totalCustomersArrived = 0;
    int fsBusy = FALSE, ssBusy = FALSE;

    smpl(0, "M/M/2/8 Queue");
    init_bm(200, 2000); /* let m0 be 200 and mb be 2000 observations */
    mm28_facility = facility("mm28_facility", 2);
    schedule(1, 0.0, tk_id);
    while (cont)
    {
        cause(&event, &customer);
        switch (event)
        {
        case 1: /* arrival */
            totalCustomersArrived++;
            if (++tk_id >= TOKENS) tk_id = 0;
            n = inq(mm28_facility);
            if (n < 6) /* there is room to accept this client */
            {
                ts[customer] = time();
                schedule(2, 0.0, customer);
            }
            else { rejected++; }
            schedule(1, expntl(Ta), tk_id);
            break;
        case 2: /* request service */
            if (request(mm28_facility, customer, 0) == 0)
            {
                if (!fsBusy)
                {
                    schedule(3, expntl(Tfs), customer);
                }
            }
        } /* end switch */
    } /* end while */
```
fsBusy=TRUE;
}
else if (!ssBusy)
{
    schedule(4, expntl(Tss), customer);
    ssBusy=TRUE;
}
else printf("Error:\n");
} break;
case 3: /* release fast server */
    release(mm28_facility, customer);
    fsCompleted++;
    fsBusy=FALSE;
    if (obs(time() - ts[customer]) == 1) cont = FALSE;
    break;
case 4: /* release fast server */
    release(mm28_facility, customer);
    ssCompleted++;
    ssTotalResponseTime += time() - ts[customer];
    ssBusy=FALSE;
    break;
}
} TotalTimeElapsed = time();
rejectProb = rejected/totalCustomersArrived;
printf("(1) Customer turned-away probability is %f\n", rejectProb);
queueLength = Lq(mm28_facility);
printf("(2) Average number of customers waiting is %f\n", queueLength);
civals(&mean, &hw, &nb);
printf("(3) Response time for those served by fast server is %f; half width is %f; confidence accuracy is %f\n", mean, hw, hw/mean);
printf("(4) Response time for those served by slow server is %f\n", ssTotalResponseTime/ssCompleted);
totalCompleted = totalCustomersArrived - rejected; /*Serviced customers*/
X = totalCompleted/TotalTimeElapsed;
printf("(5) Throughput is %f\n", X);
}

Output:
batch 1 mean = 0.711
batch 2 mean = 0.693
batch 3 mean = 0.657
batch 4 mean = 0.705
batch 5 mean = 0.713
batch 6 mean = 0.656
batch 7 mean = 0.674
batch 8 mean = 0.696
batch 9 mean = 0.681
batch 10 mean = 0.704, rel. HW = 0.022
(1) Customer turned-away probability is 0.404132
(2) Average number of customers waiting is 4.641454
(3) Response time for those served by fast server is 0.689223; half width is 0.015062; confidence accuracy is 0.021854
(4) Response time for those served by slow server is 0.845672
(5) Throughput is 8.930777
2. (25 points.) Assume that 5 components (1, 2, 3, 4 and 5) obeying the exponential failure law have the failure rates of \( \lambda_1 = 0.01, \lambda_2 = 0.02, \lambda_3 = 0.03, \lambda_4 = 0.04, \) and \( \lambda_5 = 0.05, \) respectively and they are connected by a bridge structure as shown on slide #57.

(a) (10 points.) Derive an analytical expression for the system reliability of this bridge system at time t based on the minimal path set method and then plug in \( t=12 \) and component failure rates to compute the system reliability at \( t=12. \) Note: you must use the minimal path set method covered in the lecture.

\[
\varphi(t) = 1-[(1 - X_1X_2X_3)(1 - X_1X_4)(1 - X_2X_5)(1 - X_2X_4)]
\]

\[
= -X_2X_3X_4 + 2X_1X_2X_3X_5 -X_2X_3X_4X_5 -X_1X_3X_4X_5 + X_1X_4X_5 + X_2X_4X_5 -X_2X_3X_4X_5 + X_2X_5
\]

\[
E[\varphi(t)] = R(t) =
\]

\[
E[\varphi(t)] = 1 - \sum \text{events}
\]

\[
E[X_1X_3X_5] + E[X_1X_4] + E[X_2X_3X_4] + E[X_1X_2X_3X_5] + E[X_1X_2X_4X_5] + E[X_2X_3X_4X_5] + E[X_2X_5]
\]

\[
R(12) = (e^{-\lambda_112} + e^{-\lambda_212} + e^{-\lambda_312} + e^{-\lambda_412} + e^{-\lambda_512} + e^{-\lambda_1212} + e^{-\lambda_2212} + e^{-\lambda_3212} + e^{-\lambda_4212} + e^{-\lambda_5212} + e^{-\lambda_1312} + e^{-\lambda_2312} + e^{-\lambda_3312} + e^{-\lambda_4312} + e^{-\lambda_5312} + e^{-\lambda_1412} + e^{-\lambda_2412} + e^{-\lambda_3412} + e^{-\lambda_4412} + e^{-\lambda_5412} + e^{-\lambda_1512} + e^{-\lambda_2512} + e^{-\lambda_3512} + e^{-\lambda_4512} + e^{-\lambda_5512})
\]

\[
= 0.78854
\]

(b) (5 points.) Write a sharpe code based on a reliability graph to compute the system reliability at time \( t=12 \) units. Show the sharpe code and output.

```
relgraph bridge(v1,v2,v3,v4,v5)
a b exp(v1)
a c exp(v2)
b d exp(v4)
c d exp(v5)
bidirect
b c exp(v3)
end
expr 1-value(12;bridge;0.01,0.02,0.03,0.04,0.05)
end

Output: 1-value(12;bridge;0.01,0.02,0.03,0.04,0.05): 7.8854e-01
```

(c) (5 points.) Suppose you wish to solve the problem using fault trees. Show a fault tree model for the bridge system and then write a sharpe program based on your fault tree model to compute the system reliability at time \( t=12 \) units. Show the sharpe code and output. Note that your solutions from parts (a), (b) and (c) should be the same.
(d) (5 points.) Suppose that these 5 components also obey the exponential repair law with the repair rates of $\mu_1 = 0.1$, $\mu_2 = 0.2$, $\mu_3 = 0.3$, $\mu_4 = 0.4$, and $\mu_5 = 0.5$, respectively. Write a Sharpe program based on your fault tree model to compute the instantaneous system availability at time $t=12$ units and the steady state availability. Show the Sharpe code and output.

```sharpe
poly unavail(mu, lambda)\n  gen\n  1, 0, 0;\n  - mu/(lambda+mu), 0, 0;\n  - lambda/(lambda+mu), 0, - (lambda+mu)\n
ftree system
repeat c1 unavail (0.1, 0.01)
repeat c2 unavail (0.2, 0.02)
repeat c3 unavail (0.3, 0.03)
repeat c4 unavail (0.4, 0.04)
repeat c5 unavail (0.5, 0.05)
or c1_3_5 c1 c3 c5
or c1_4 c1 c4
or c2_5 c2 c5
or c2_3_4 c2 c3 c4
and top c1_3_5 c1_4 c2_5 c2_3_4
end
expr 1-value(12;system)
end
```

Output: 1-value(12;system): 7.8854e-01

```sharpe
expr pinf(system)
end
```

Output:

<table>
<thead>
<tr>
<th>1-value(12;system)</th>
<th>9.8518e-01</th>
<th>//instantaneous availability at time t=12</th>
</tr>
</thead>
<tbody>
<tr>
<td>pinf(system)</td>
<td>9.8230e-01</td>
<td>//steady state availability</td>
</tr>
</tbody>
</table>