Problem 1:

batch 1 mean = 0.212
batch 2 mean = 0.214
batch 3 mean = 0.210
batch 4 mean = 0.205
batch 5 mean = 0.212
batch 6 mean = 0.209
batch 7 mean = 0.199
batch 8 mean = 0.210
batch 9 mean = 0.202
batch 10 mean = 0.208, rel. HW = 0.016

(1) Customer turned-away probability is 0.000297
(2) Average number of customers waiting is 0.045123
(3) Mean Response Time is 0.207993 and half width is 0.003422
(4) Throughput is 4.934838

#include "smpl.h"
define TOKENS 1000
define TRUE 1
define FALSE 0
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 */
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);
bake;
case 3: /* release server */
release(server,customer);
completed++;
if (obs(time()-ts[customer]) == 1) cont = FALSE;
bake;
}
}
TotalTimeElapsed = time();
rejectProb = rejected/(real)(totalCustomersArrived);
printf("(1) Customer turned-away probability is %f\n",rejectProb);
queueLenght =Lq(server);
printf("(2) Average number of customers waiting is %f\n",queueLenght);
civals(&mean, &hw, &nb);
printf("(3) Mean Response Time is %f and half width is %f\n", mean, hw);
X = (real)(completed)/TotalTimeElapsed;
printf("(4) Throughput is %f\n", X);
}
Problem 2:
The reliability of the system is:

\[
R(t) = R_0 \left[ 1 - (1-R_f)^3 \right] = e^{-\lambda_f t} \left[ 1 - (1-e^{-\lambda_f t})^3 \right] \\
= e^{-\lambda_f t} \left[ 3e^{-\lambda_f t} - 3e^{-2\lambda_f t} + e^{-3\lambda_f t} \right] \\
= 3e^{-(\lambda_w+\lambda_f) t} - 3e^{-(\lambda_w+2\lambda_f) t} + e^{-(\lambda_w+3\lambda_f) t}
\]

The mean time to failure (MTTF) is:

\[
MTTF = \int_0^\infty R(t)\,dt = \int_0^\infty 3e^{-(\lambda_w+\lambda_f) t} - 3e^{-(\lambda_w+2\lambda_f) t} + e^{-(\lambda_w+3\lambda_f) t} \,dt \\
= \frac{3}{\lambda_w + \lambda_f} - \frac{3}{\lambda_w + 2\lambda_f} + \frac{1}{\lambda_w + 3\lambda_f}
\]

Problem 3:

a. Write a Sharpe code based on a reliability graph model to compute the system reliability after 5 weeks of operation.

2. Source code:

```
bind
v1 0.00007
v2 0.00006
v3 0.00005
v4 0.00004
v5 0.00003
v6 0.00002
v7 0.00001
end
relgraph graph1
a b exp(v1)
a c exp(v3)
b d exp(v4)
d e exp(v6)
c e exp(v7)
bidirect
b c exp(v2)
c d exp(v5)
end
expr 1-value(24*7*5;graph1)
End
```

b. Find the minimal path and minimal cut sets.

the minimal paths are: \{1,4,5,7\}, \{1,4,6\}, \{1,2,5,6\}, \{1,2,7\}, \{3,7\}, \{3,5,6\}, \{3,2,4,6\}
the minimal cuts are: \{2,3,4\}, \{2,3,5,6\}, \{1,3\}, \{4,5,7\}, \{6,7\}, \{1,2,5,7\}

c. Build a fault tree model based on the minimal cut set identified in (b) and then write a Sharpe code based on your fault tree model to compute the system reliability after 5 weeks of operation. The answer obtained here should be the same as that obtained from part (a).

1. fault tree model

```
  Failure
  \ OR \n  \ AND \n  \ AND \n  2  3  4
  \ AND \n  2  3  5  6
  \ AND \n  1  3
  \ AND \n  4  5  7
  \ AND \n  6
  \ AND \n  7
  \ AND \n  2  5  7
  or top s1 s2 s3 s4 s5 s6
  end
```


3. Source code:
```
ftree system
repeat 1 exp(0.00007)
repeat 2 exp(0.00006)
repeat 3 exp(0.00005)
repeat 4 exp(0.00004)
repeat 5 exp(0.00003)
repeat 6 exp(0.00002)
repeat 7 exp(0.00001)
and s1 2 3 4
and s2 2 3 5 6
and s3 1 3
and s4 4 5 7
and s5 6 7
and s6 1 2 5 7
or top s1 s2 s3 s4 s5 s6
end

*print reliability after 5 weeks
expr 1-value(24*7*5; system)
end
```