1. (1) The customer turned away probability is 0.28.
   (2) The average number of customers waiting in the system is 4.79.
   (3) The response time per client for those served by the fast server is 0.64 s.
   (4) The response time per client for those served by the two slow servers is 0.78 s.
   (5) The throughput is 10.80.
   (6) The report confidence accuracy is $H/Y = \frac{0.003165}{0.642219} = 4.9\%$.

**Source Code:**

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

main()
{
    real Ta=1.0/15,Ts_fast=0.2,Ts_slow=1.0/3,mean,hw,nq;
    real fast=0.0,slow=0.0;
    real tk_total=0;
    int tk_id=0,customer=0,event,server,nb;
    int flg; /*a flag to choose one of the two slow servers*/
    int fast_cus=0,slow_cus=0; /*number of customers servered by fast server or slow servers*/
    int lost_cus=0; /*number of customers turned away by the system*/
    int que_cus; /*number of tokens in queue*/
    int server1=0,server2=0,server3=0; /*server availability, 0 means available, 1 means unavailable*/
    real ts[TOKENS];
    int cont=TRUE;
    smpl(0, "M/M/3/10 Quene with BMA");
    init_bm(200,20000);
    server=facility("server",3);
    schedule(1,0.0,tk_id);
    while (cont)
    {
        cause(&event,&customer);
        switch(event)
        {
        case 1:/*arrival*/
            tk_total++;
            if(++tk_id>=TOKENS) tk_id=0;
            schedule(1,expntl(Ta),tk_id);
            que_cus=inq(server);
```

```c
```
if (que_cus >= 7) lost_cus++;
else {
    schedule(2,0,0,customer);
    ts[customer]=time();
}
break;
case 2: /*request server*/
if (request(server,customer,0)==0){
    if (server1==0){
        schedule(3,expntl(Ts_fast),customer);
        server1=1;
    } else if (server2==0){
        schedule(4,expntl(Ts_slow),customer);
        server2=1;
    } else {
        schedule(5,expntl(Ts_slow),customer);
        server3=1;
    }
}
break;
case 3: /*departure from the fast server*/
release(server,customer);
server1=0;
fast_cus++;
fast+=time()-ts[customer];
if (obs(time()-ts[customer])==1)
    cont=FALSE;
break;
case 4: /*departure from the first slow server*/
release(server,customer);
server2=0;
slow_cus++;
slow+=time()-ts[customer];
break;
case 5: /*departure from the second slow server*/
release(server,customer);
server3=0;
slow_cus++;
slow+=time()-ts[customer];
break;
}
/*end while*/
civals(&mean,&hw,&nb);
/*answer for Q1*/
printf("Customer turnedaway probability: %f
", lost_cus/tk_total);
/*answer for Q2*/
printf("Average number of customers waiting in the system: %f\n", Lq(server));  
/*answer for Q3*/
printf("Response time per client served by fast server: %f\n", fast/fast_cus);  
/*answer for Q4*/
printf("Response time per client served by slow servers: %f\n", slow/slow_cus);  
/*answer for Q5*/
printf("Throughput: %f\n", (fast_cus + slow_cus)/time());  
/*answer for Q6*/
printf("Y=%f,H=%f after %d batches\n",mean,hw,nb);
}

2. The system structure is shown as follows:

\[
R(t) = R_w(t)[1 - F_{f1}(t)F_{f2}(t)F_{f3}(t)] = R_w(t)[1 - (1 - R_f(t))^3] \\
= e^{-\lambda_{\omega}t} \left[1 - (1 - e^{-\lambda_{f}t})^3\right] = 3e^{-\lambda_{\omega}t - \lambda_{f}t} - 3e^{-2\lambda_{\omega}t - 2\lambda_{f}t} + e^{-3\lambda_{\omega}t - 3\lambda_{f}t} \\
\text{MTTF} = \int_0^\infty R(t) dt = \frac{3}{\lambda_{\omega} + \lambda_{f}} - \frac{3}{\lambda_{\omega} + 2\lambda_{f}} + \frac{1}{\lambda_{\omega} + 3\lambda_{f}}
\]
3. (a) The system reliability after 12 weeks of operation is 0.98279.

The vertices of the network marked with characters a, b, c, d, e are shown as follows:

![Network Diagram]

Source Code:

```plaintext
bind
r1 0.00001
r2 0.00002
r3 0.00003
r4 0.00004
r5 0.00005
r6 0.00006
r7 0.00007
end

relgraph bridge
a b exp(r1)
a c exp(r3)
b d exp(r4)
c e exp(r6)
d e exp(r6)
c e exp(r7)
end

bidirect
b c exp(r2)
c d exp(r5)
end

expr 1-value(2016; bridge)
pqcdf (bridge)
end
```

(b) 7 minimal path sets in total:
\{1,2,5,6\}, \{1,2,7\}, \{1,4,5,7\}, \{1,4,6\}, \{2,3,4,6\}, \{3,5,6\}, \{3,7\}

6 minimal cut sets in total:
\{1,3\}, \{6,7\}, \{2,3,4\}, \{4,5,7\}, \{1,2,5,7\}, \{2,3,5,6\}

The structure using parallel connection of the **minimal path sets** is shown as follows:

![Parallel Connection Structure]
The structure using parallel connection of the **minimal cut sets** is shown as follows:

(c) The system reliability after 12 weeks of operation is **0.98279**.

The fault tree model built based on the **minimal path sets** is shown as follows:

```
bind
r1 0.00001
r2 0.00002
r3 0.00003
r4 0.00004
r5 0.00005
r6 0.00006
r7 0.00007
end

ftrace network
repeat p1 exp(r1)
repeat p2 exp(r2)
repeat p3 exp(r3)
repeat p4 exp(r4)
repeat p5 exp(r5)
repeat p6 exp(r6)
repeat p7 exp(r7)
OR forkl p1 p2 p3 p6
OR forkl p1 p2 p7
OR fork2 p1 p4 p5 p7
OR fork4 p1 p4 p6
OR fork5 p2 p3 p4 p6
OR fork6 p3 p7
OR fork7 p3 p4 p6
AND top forkl fork2 fork3 fork4 fork5 fork6 fork7
end

expr 1-value(2016;network)
pqodf (network)
end
```

Output

```
1-value(2016;network): 9.8279e-001
CDF for system network:

[Q(p1)*Q(p3)]+
[Q(p6)*Q(p7)*(1-Q(p1)*Q(p3))]+
[P(p1)*Q(p3)]*
[Q(p4)*Q(p5)*(1-Q(p6)*Q(p7))]+
[P(p2)*Q(p3)]*
[Q(p4)*Q(p5)*P(p6)*Q(p7)*(1-Q(p1))]+
[Q(p4)*Q(p5)*P(p6)*Q(p7)*(1-Q(p3))]+
[Q(p1)*Q(p2)]*
[P(p3)*P(p4)*Q(p5)*P(p6)*Q(p7)]+
[P(p1)*Q(p2)]*
[Q(p3)*P(p4)*Q(p5)*Q(p6)*P(p7)]
```