CS 350 Homework #1 Tim Duffy

1) a.

x	P(X=x)	xP (X = x)	$(x - \mu)^2 * P(X = x)$
1	0.25	0.25	$(1 - 2.08)^2 * (0.25) = 0.2916$
2	0.42	0.84	$(2 - 2.08)^2 * 0.42 = 0.002688$
3	0.33	0.99	$(3 - 2.08)^2 * 0.33 = 0.279312$

The mean time for a single transaction is μ = (1 * 0.25) + (2 * 0.42) + (3 * .33) = **2.08**

The standard deviation for a single transaction is:

σ = √(0.2916 + 0.002688 + 0.279312) = √(0.5736) = **0.757363849**

b. The probability mass function for three transactions can be defined as follows:

Total Time	Number of Events	Probability
0	none	0
1	none	0
2	none	0
3	(1,1,1)	0.015625
4	(1,1,2), (1,2,1), (2,1,1)	0.07875
5	(1,1,3), (1,3,1), (1,2,2), (2,1,2), (2,2,1), (3,1,1)	0.194175
6	(1,2,3), (1,3,2), (2,1,3), (2,2,2), (2,3,1), (3,1,2), (3,2,1)	0.281988
7	(1,3,3), (2,2,3), (2,3,2), (3,1,3), (3,2,2), (3,3,1)	0.256311
8	(2,3,3), (3,2,3), (3,3,2)	0.137214
9	(3,3,3)	0.035937

c.

x	P(X=x)	xP (X = x)	$(x - \mu)^2 * P(X = x)$
3	0.015625	0.046875	$(3 - 6.24)^2 * (0.015625) = 0.164025$
4	0.07875	0.315	$(4 - 6.24)^2 * (0.07875) = 0.395136$
5	0.194175	0.970875	$(5 - 6.24)^2 * (0.194175) = 0.298563$
6	0.281988	1.691928	$(6 - 6.24)^2 * (0.281988) = 0.016243$
7	0.256311	1.794177	$(7 - 6.24)^2 * (0.256311) = 0.148045$
8	0.137214	1.097712	$(8 - 6.24)^2 * (0.137214) = 0.425034$
9	0.035937	0.323433	$(9 - 6.24)^2 * (0.035937) = 0.273754$
total	1	6.24	1.7208

The mean of the total time for three transactions is the sum of the third column, **6.24.**

The standard deviation for three transactions is σ = √(1.7208) **= 1.3118**

d. The CDF(x) can be defined as CDF(x) = (P(X <= x)), where CDF(x) = CDF(x-1) + P(X=X)

CDF(x) is defined by the following chart:

0	P(X<=x)
>3	0
3	0.015625
4	0.094375
5	0.288550
6	0.570538
7	0.826849
8	0.964063
9	1

e. There is a 28.8% chance that three transactions will take 5 seconds or less. (Assuming that the units of time are seconds).

f. The PMF for at least 1 event taking maximum time is defined as follows:

Total Time	Events	Probability
5	(1,1,3), (1,3,1), (3,1,1)	0.061875
6	(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)	0.207900
7	(1,3,3), (2,2,3), (2,3,2), (3,1,3), (3,2,2), (3,3,1)	0.256311
8	(2,3,3), (3,2,3), (3,3,2)	0.137214
9	(3,3,3)	0.035937

g.

x	P(X=x)	xP (X = x)	$(x - \mu)^2 * P(X = x)$
5	0.061875	0.309375	$(5 - 6.824)^2 * (0.061875) = 0.205857$
6	0.207900	1.247400	$(6 - 6.824)^2 * (0.207900) = 0.14115911$
7	0.256311	1.794177	$(7 - 6.824)^2 * (0.256311) = 0.007939$
8	0.137214	1.097712	$(8 - 6.824)^2 * (0.137214) = 0.189764$
9	0.035937	0.323433	$(9 - 6.824)^2 * (0.035937) = 0.170161$
total	0.699237	4.772097	0.7149

The mean of any one transaction taking the maximum time is **6.82472037**
Standard deviation for three transactions is: $\sigma = \sqrt{(0.7149)}$ = **0.170161**

2) Given: $q = 8$ $T_s = 0.2$ msec $T_q = 1$ msec

a. We can determine the rate at which packets arrive at the router (λ) by using the average number of packets in the system at any time (q), and the time to process and transmit a single packet (T_q). Using Little's formula, we get $T_q = q/\lambda \Rightarrow \lambda = q/T_q$.

$\lambda = q/T_q \Rightarrow \lambda = 8$ packets/1 msec = **8 packets/msec**

b. The average number of packets waiting to be processed can be derived using T_s, T_q and λ.

We know that $\lambda * T_w = w$ = the average number of packets waiting.
We also know that $T_q = T_s + T_w$. So,

$T_w = T_q - T_s = 1$ msec - 0.2 msec = 0.8 msec
and $\lambda * T_w = w = 8$ packets/msec * 0.8 msec = an average of 10 packets waiting to be processed

c. A single processor can process a packet at a rate of 5 packets/msec. Since there is an incoming rate of 8 packets/msec, there must be a minimum of 2 processors in this router.

d. The answer above hold true if the processors can process packets at a rate of 5 packets/msec or higher. If they slow down too much, then the router may need an additional processor to cut down on traffic buildup.

3) Given: P(server is available) = 0.98 => P(server is down) = 1 - 0.98 = 0.02

a. P(out of 100 connections, none will be refused) = $(0.98)^{100}(0.02)$ = **0.002652**
This assumes that the 101st connection is refused.

b. P(exactly 1 of 100 connections are refused) = (100 choose 1) * $(0.98)^{99}$*(0.02)
= 100 * 0.00270652 = **0.270652**

c. P(At most, 3 of 100 connections are refused)
= P(0 refused) + P(1 refused) + P(2 refused) + P(3 refused)
= 0.002652 + 0.270652 + [(100 choose 2) * $(0.98)^{98}$ * $(0.02)^2$]
+ [(100 choose 3) *$(0.98)^{97}$*$(0.02)^3$] +

= 0.002652 + 0.270652 + 0.273414 + 0.182275941 = **0.728993941**

d. P(50 or more consecutive requests are not refused) = P(at least 50 are accepted)
= $(0.98)^{50}$ = 0.36416968.

Probability that 50 or more connections are successful <= 0.36416968

e. The average number of requests successfully served before a connection is refused is the number of connections served before probability of success drops below 50% (average).

$0.5 = (0.98)^n$ where n is the number of successes
$\log(0.5) = n \log(0.98)$ => $-0.301029996 = n * (-0.00877392431)$

=> $n = (-0.301029996)/(-0.00877392431) = 34.31 \approx$ **34 successes on average**

f. If the probability the server is available is 98%, then 98% of the time the request will be served successfully. 0.98 * 500 = 490.

On average, **490 requests out of 500** will be served.

4) Given: $\mu = 20$ msec $\sigma = 5$ msec

a. $f(x) = (1/\ \sigma\sqrt{(2\pi)}) * e\text{^} (-0.5 * ((x-\mu*100)/(\sigma*10))^2)$
$f(x) = (1/\ 5\sqrt{(2\pi)}) * e\text{^} (-0.5 * ((x-20)/5)^2)$

b. Probability that the 100 requests will take between 2 and 2.1 seconds

$z1 = z(2000 \text{ msecs}) = (2000 - 20*100) / (5\sqrt{(100)}) = 0.00$
$z2 = z(2100 \text{ msecs}) = (2100 - 20*100) / (5\sqrt{(100)}) = 100/(5*10) = 2.00$
$P(z1) = 0.50$
$P(z2) = 0.9772$
$P(z2) - P(z1) = 0.9772 - 0.5 =$ **0.4772 = 47.22 %**

c. There is 0% chance that 100 requests will take over 2.5 seconds to complete. This is, of course assuming that the server maintains its average response time of 20 msecs, with 5 msec standard deviation.

Even in the worst case, a single request will take 20 + 5 msecs = 25 msecs. For 100 requests, this maximum would be 0.025 sec * 100 = 2.5 seconds. If 2.5 seconds is the maximum time the server could spend on these requests, there is no chance it will take longer.

5)

```
import java.util.Random;
import static java.lang.Math.*;
import java.io.*;
import java.util.Scanner;

class Exponential {
   public static void main(String[] args) {
     // Random number generator
     Random rndm = new Random();

     Scanner console = new Scanner(System.in);
     System.out.println("Please enter an integer value for the
                           mean of the exponential distribution");
     Double mean = console.nextDouble();
     Double lambda = 1.0/mean;
     Double sum = 0.0;

     for(int i = 1; i <= 100; i++){
       /*The Random class, along with the nextFloat function returns the next
      * pseudorandom, uniformly distributed float value between 0.0 and 1.0
      * from this random number generator's sequence */
       Double U = rndm.nextDouble();
       Double V = (-1* log(1-U)) / lambda;

       System.out.println(V);
       sum += V;
     }

     System.out.print("The mean of these 100 numbers is actually ");
     System.out.println(sum/100);
   }
}
```

The CDF obtained empirically is similar to, but not exactly the same as the CDF obtained analytically. This is because the 100 values generated by this program do not fall perfectly around the mean we input. Since the generated values are not perfectly distributed, the mean and standard deviation differ slightly for each trial.