Mini Project # 1
Name: Niveditha Varadha Chandrasekaran

Section 1

Given:
A program is divided into 3 blocks that are being compiled on 3 parallel computers. Each block takes an Exponential amount of time, 5 minutes on the average, independently of other blocks.

Steps:
Let X1, X2 and X3 be the random variables representing the first, second and third block of the program respectively and X be the variable representing the time taken to compile the whole program.

Step 1: The block compilation times X1, X2 and X3 follow an exponential distribution, where lambda is given as 0.2 (1/5). Simulate X1, X2 and X3 by using rexp() function and compute the time taken to compile the whole program, i.e. X = max(X1, X2, X3).

Step 2: Repeat step 1 for 10000 times (N = 10000) to get 10,000 draws from the distribution of X and construct a histogram of X and superimpose the theoretical density function of X using curve() function.

Theoretical density function of X:

$$f(x) = 3\lambda(1 - e^{-\lambda x})^2 e^{-\lambda x}$$

Step 3: Use step 2 and calculate the mean of X given by Mean(X) and variance of X given by Var(X) and calculate the value of $E(X^2)$ using the formula $E(X^2) = Var(X) + [Mean(X)]^2$.

Step 4: Compare the estimated value of $E(X^2)$ from the simulation with the theoretical value of $E(X^2)$ which is given as 118.1962.

Section 2

1. **Simulate the block compilation times X1, X2 and X3. Use the simulated values to simulate X, the compilation time of the whole program.**

   E(X) = 1/ λ
   Given: E(X) = 5
   λ = 1/ E(X) = 1/5 = 0.2

   Time taken by the first module to be complied:
   X1 = rexp(1, 0.2)

Time taken by the second module to be complied:
X2 = rexp(1, 0.2)
Time taken by the third module to be complied:
X3 = rexp(1, 0.2)

Then time taken by the whole program to be compiled:
X = max(X1, X2, X3)

The R code is given in Section 3.

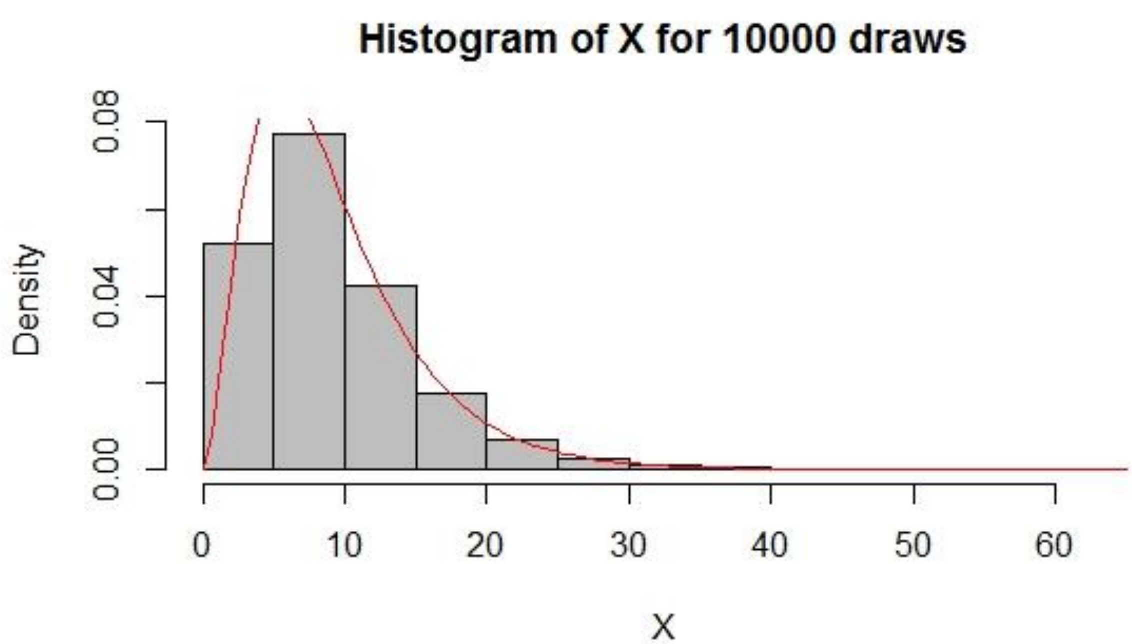
2. **Repeat the previous step 10,000 times. This will give you 10,000 draws from the distribution of X.**

   Consider N = 10000, use the replicate function to get the distribution of X.

   The R code is given in Section 3.

3. **Make a histogram of the draws of X. Superimpose the theoretical density function of X. Try using the R function `curve' for drawing the density. Note what you see.**
   The R code is given in Section 3.
   Form the figure below, X follows Normal Distribution with the curve skewed to the right.
   The Density function of X is:

$$f(x) = 3\lambda(1 - e^{-\lambda x})^2 e^{-\lambda x}$$



4. **Use the draws to estimate $E(X^2)$. Compare your answer with the exact answer of $E(X^2)$. Note what you see.**

The R code is given in Section 3.

The given theoretical value for $E(X^2)$ is 118.1962

The value of $E(X^2)$ estimated from the simulation using the formula $E(X^2) = Var(X) + [Mean(X)]^2$ is: 116.2526

The estimated value varies from the theoretical value of $E(X^2)$.

5. **Repeat the process of obtaining an estimate of $E(X^2)$ five times. Compare each estimate with the exact value. Note what you see.**

Attempts	1	2	3	4	5
$E(X^2)$	116.2526	118.3709	116.9677	119.0796	117.2117

Since the value n (= 10000) is large the estimated value is closer to the theoretical value for each time the experiment is repeated.

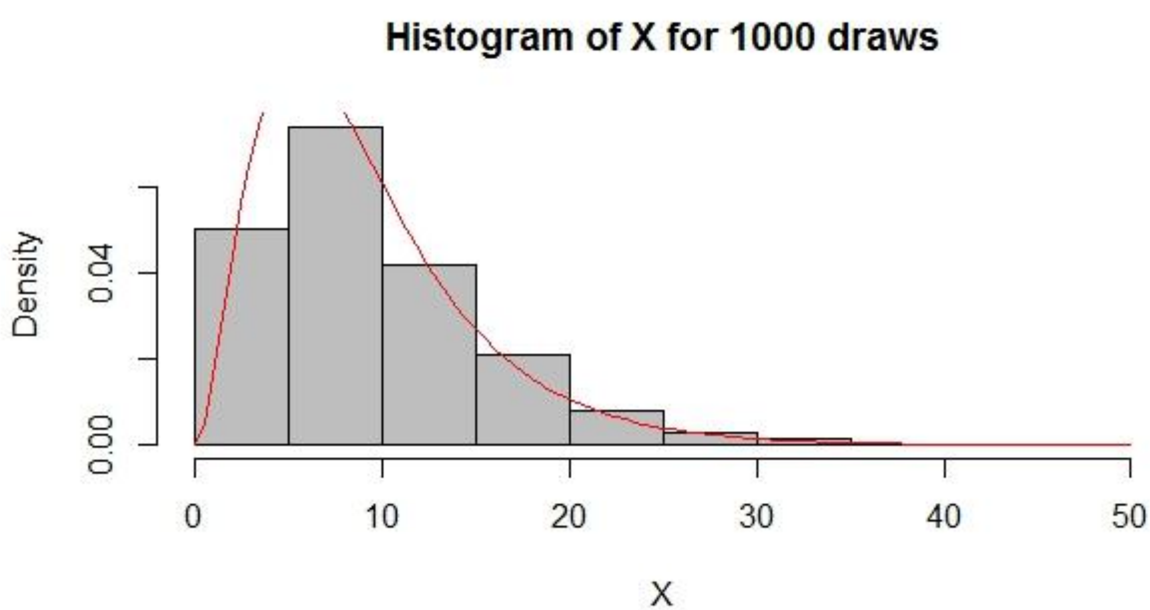
6. **Comment on how your results would change if you use 1,000 Monte Carlo replications instead of 10,000. What if you use 100,000 replications? Justify your answers.**

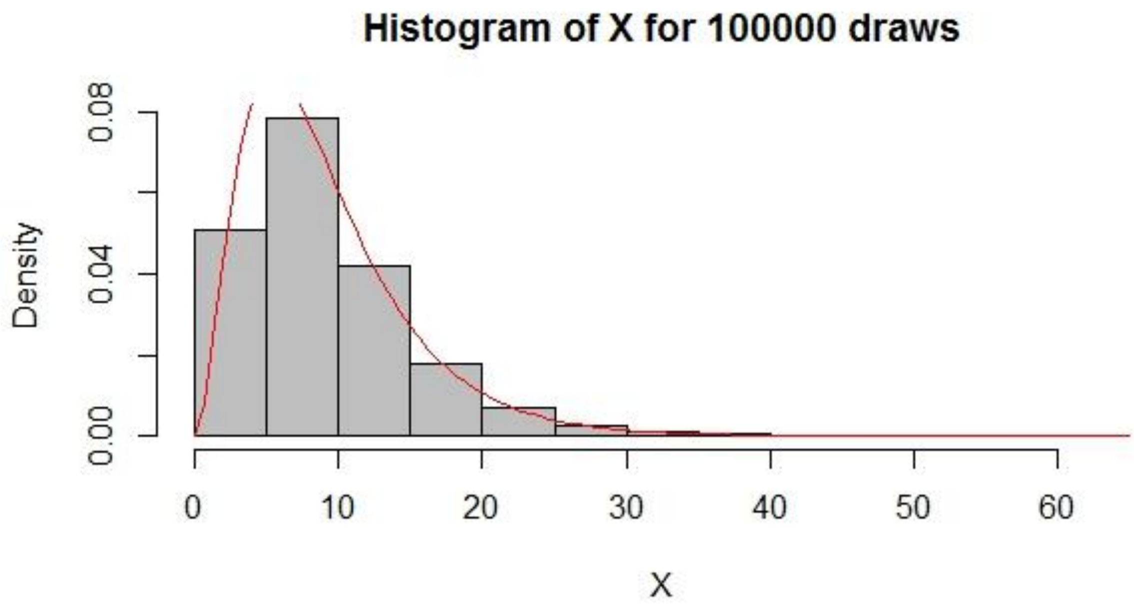
N = 1000

Attempts	1	2	3	4	5
$E(X^2)$	113.3796	125.2162	112.2896	118.934	122.6561

N = 100000

Attempts	1	2	3	4	5
$E(X^2)$	118.4143	118.0157	118.0954	117.5704	118.2620





On comparing the estimated values of $E(X^2)$ for the distribution of X with 1000, 10000, and 100000 draws we can see that as number of draws (N) increases, the estimated value of $E(X^2)$ is closer to that of the theoretical value of $E(X^2)$ (118.1962).

Section 3

R Code:

```
> lambda <- 0.2 # rate of the expotential function.
> #X1, X2, X3 - random variables representing the compilation time for first,
second and third block respectively.
> X1 <- rexp(1, lambda)
> X2 <- rexp(1, lambda)
> X3 <- rexp(1, lambda)
> X <- 0  # X is the random variable representing the compilation time for th
e whole program.
> X <- max(X1, X2, X3)
> X1
[1] 2.177446
> X2
[1] 0.6191411
> X3
[1] 6.585034
> X
[1] 6.585034


> #---------------------------------------------------------------------------
```

```
> # for 10000 draws
> n <- 10000 # number of replications.
> X <- replicate(n, max(X1 = rexp(1, lambda), X2 = rexp(1, lambda), X3 = rexp
(1, lambda)))
> #Histogram of X
> hist(X, probability = TRUE, col = "grey",main = "Histogram of X for 10000 d
raws")
> #Density curve of X
> curve(3*lambda*(1-exp(-lambda*x))^2*(exp(-lambda*x)), ylab="Density",xlab="
X",col = "red", add = TRUE)
> Mean_X <- mean(X) #Gives the mean of the distribution X
> VAR_X <- var(X)   #Gives the variance of the distribution X
> Exp_X_Square <- Mean_X^2 + VAR_X #E(X2) = Var(X) + [Mean(X)]^2
> Exp_X_Square #Expected value of X^2 from the simulation.
[1] 116.1898


> #-----------------------------------------------------------------------------


> # for 1000 draws
> n <- 1000 # number of replications.
> X <- replicate(n, max(X1 = rexp(1, lambda), X2 = rexp(1, lambda), X3 = rexp
(1, lambda)))
> #Histogram of X
> hist(X, probability = TRUE, col = "grey", main = "Histogram of X for 1000 d
raws")
> #Density curve of X
> curve(3*lambda*(1-exp(-lambda*x))^2*(exp(-lambda*x)), ylab="Density",xlab="
X",col = "red", add = TRUE)
> Mean_X <- mean(X) #Gives the mean of the distribution X
> VAR_X <- var(X) #Gives the variance of the distribution X
> Exp_X_Square <- Mean_X^2 + VAR_X #E(X2) = Var(X) + [Mean(X)]^2
> Exp_X_Square #Expected value of X^2 from the simulation.
[1] 126.7381


> #-----------------------------------------------------------------------------


> # for 100000 draws
> n <- 100000 # number of replications.
> X <- replicate(n, max(X1 = rexp(1, lambda), X2 = rexp(1, lambda), X3 = rexp
(1, lambda)))
> #Histogram of X
> hist(X, probability = TRUE, col = "grey", main = "Histogram of X for 100000
draws")
> #Density curve of X
> curve(3*lambda*(1-exp(-lambda*x))^2*(exp(-lambda*x)), ylab="Density", xlab=
"X",col = "red", add = TRUE)
> Mean_X <- mean(X) #Gives the mean of the distribution X
> VAR_X <- var(X) #Gives the variance of the distribution X
> Exp_X_Square <- Mean_X^2 + VAR_X #E(X2) = Var(X) + [Mean(X)]^2
> Exp_X_Square #Expected value of X^2 from the simulation.
[1] 117.6574
```