```r
a <- 1103515245
b <- 12345
m <- 2^31
seed1 <- 12345
seed2 <- 54321
x <- seed1
y <- seed2
x.s <- NULL
y.s <- NULL
N <- 10000

#Generate random numbers using seed1
for(i in 1:N) {
 #modulus is %%
 x <- (a*x + b) %% m
 x.s <- c(x.s,x/m)
}

#Generate random numbers using seed2
for(i in 1:N) {
 #modulus is %%
 y <- (a*y + b) %% m
 y.s <- c(y.s,y/m)
}

#3
mean(exp(exp(x.s)))

#4
mean((1-x.s^2)^(3/2))

#5
mean(exp((x.s*4-2) + (x.s*4-2)^2)*4)

#6
mean((1/x.s-1)*(1+(1/x.s-1)^2)^-2*((1/x.s)^2))

#7
mean(2*exp(-(1/x.s-1)^2)*((1/x.s)^2))

#8
mean(exp((x.s+y.s)^2))

#9
i <- y.s < x.s
mean(exp(-(1/x.s-1))*(1/x.s^2)*exp(-(1/y.s-1))*(1/y.s^2)*i)
```

```
#10
mean(x.s)
mean(exp(x.s))
mean((x.s-.5)*(exp(x.s)-1.7177))

#11
#a
mean(sqrt(1-x.s^2))
mean(x.s*(sqrt(1-x.s^2)))
mean((x.s-.5)*(sqrt(1-x.s^2)-.7845051))

#b
mean(x.s^2)
mean((x.s^2-.3338223)*(sqrt(1-x.s^2)-.7845051))

#12
#T sets of means used to determine E[N] where N is the minimum number of random
 numbers whose sum exceeds 1
T <- 3
#C sets of random numbers whose sum just exceeds 1
C <- 100
#var to store the running count of random numbers in each set in C until the sum exceeds 1
setCount <- 0
#var to store the sum of random numbers in each set in C until the sum exceeds 1
sum <- 0
#count vector to store N, the number of random numbers in each set in C whose sum just
 exceeds 1
c.s <- NULL
#mean vector to store the T 'means,' i.e, the mean of each c.s vector (mean of N, the
 minimum number of random numbers whose sum exceeds 1)
m.s <- NULL
x <- seed1

#Generate T sets of means
for(i in 1:T) {
 #Generate C sets of random numbers such that the sum of each set just exceeds 1
 for(j in 1:C) {
  repeat{
 x <- (a*x + b) %% m
 #sum the numbers until the sum exceeds 1, include the number that made the sum exceed
 1
 sum = sum + x/m
 #keep a running count of the random numbers in each set
 setCount <- setCount + 1
 if(sum > 1){
  #store the number of random numbers in the current 'T' set vector c.s
        c.s <- c(c.s,setCount)
        #reset setCount - the running count var
```

```
        setCount <- 0
        #reset the sum
        sum <- 0
        break
 }
  }
 }
 #calculate the mean value of the current 'T' set of counts
 m.s <- c(m.s,mean(c.s))
 #reset the count vector for the next 'T'
 c.s <- NULL
 #reset x to the seed
 x <- seed1
 #calculate a new 'C' for the next 'T' round
 C <- C*10
}
m.s
```

#a-c

[1] 2.6600 2.7000 2.7448

#d

2.75

#13

#a

```
#C sets of random numbers whose product is greater than or equal to exp(-3)
C <- 10000
#var to store the running count of random numbers in each set in C until the product is
 greater than or equal to exp(-3)
setCount <- 1
#var to store the product of random numbers in each set in C until the product is greater
 than or equal to exp(-3)
prod <- 1
#count vector to store N, the number of random numbers in each set in C whose product is
 greater than or equal to exp(-3)
c.s <- NULL
#set the seed
x <- seed1

#Generate C sets of random numbers such that the product of each set is greater than or
 equal to exp(-3)
 for(j in 1:C) {
  repeat{
 x <- (a*x + b) %% m
 #multiply the numbers until the product is less than exp(-3)
 prod = prod * x/m
```

```
 if(prod < exp(-3)){
        #store the number of random numbers in the vector c.s
        c.s <- c(c.s,setCount)
        #reset setCount - the running count var
        setCount <- 1
        #reset the product
        prod <- 1
        break
 }
 else {
        #keep a running count of the random numbers in each set, does not include the factor
 that made the product less than exp(-3)
        setCount <- setCount + 1
 }
  }
}
 #calculate the mean value of the counts
 mean(c.s)
[1] 3.9703
sd(c.s)
[1] 1.73618
```

#b

```
h <- hist(c.s)
h

$density
 [1] 0.2074 0.2177 0.2338 0.1602 0.0982 0.0471 0.0237 0.0084 0.0030 0.0004
[11] 0.0001
```