```
#starting problem 3
#P(A)= 0.0174
#P(B)= 0.000751
#P(A and B)= 0.000204

simulations<- 100000

#Function to simulate spinning the spinner 4 times and returning 4 numbers greater than 6
ProbA <- function(){
 Results <- sample(x=0:10,
                   size=4,
                   replace=T)
 number <- sum(Results>6)
return(number ==4)
}

## Function to simulate spinning the spinner 4 times and returning exacly three 7's
ProbB <- function(){
Results <-sample (x=0:10,
        size=4,
        replace=T)
number<- sum(Results ==7)
return(number==3)
}

# Simulate the experiment 100,000 times and calculate probabilities
A_count <- sum(replicate(simulations, ProbA()))
B_count <- sum(replicate(simulations, ProbB()))
A_and_B_count <- sum(replicate(simulations, ProbA() & ProbB()))

# Calculate probabilities
P_A <- A_count / simulations
P_B <- B_count / simulations
P_A_and_B <- A_and_B_count /simulations

# Output probabilities
cat("Probability of event A (obtaining 4 numbers all more than 6):", P_A, "\n")
cat("Probability of event B (obtaining exactly three 7's):", P_B, "\n")
cat("Probability of event A ∩ B (both events A and B happening together):", P_A_and_B,
"\n")

#starting problem 4

# Function to simulate tossing 18 fair 6-sided dice and returning the count of 6's
myFLIPS <- sample(x= 1:6,
                  size = 18,
                  replace = TRUE)
myResults <- NA

# Simulate the experiment 100,000 times and calculate probability of at least three 6's
for(i in 1:100000) {
  myFLIPS <- sample( x=1:6,
                     size=18,
                     replace =T,
                     )
  myResults[i] <- sum(myFLIPS ==6)
}
# Output probability
Prob_at_least_3_6s <- sum(myResults >= 3)/100000

# Output probability
cat("Probability of getting at least three 6's when tossing 18 fair 6-sided dice:",
Prob_at_least_3_6s, "\n")
```

```
#starting problem 5

#Function to simulate no two consecutive rolls
simulate_no_consecutive <- function() {
  rolls <- sample(1:6, 10, replace = TRUE)
  consecutive_are_diff <- diff(rolls)
  return(all(consecutive_are_diff != 0))
}

# Simulate experiment 100,000 times and calculate probability
count_no_consecutive <- sum(replicate(100000, simulate_no_consecutive()))

# Calculate the probability
Pno_consecutive <- count_no_consecutive / 100000

# Output probability
cat("Probability of no two consecutive rolls being the same when rolling a fair 6-sided
die 10 times:", Pno_consecutive, "\n")
```