1. Write a `pois.prob()` function that computes $P(X = x)$, $P(X \neq x)$, $P(X < x)$, $P(X \leq x)$, $P(X > x)$, and $P(X \geq x)$. Enable the user to specify the rate parameter λ.

```r
pois.prob <- function(x,             #x = number of event and occurrences
                      lambda,        #lambda = the rate parameter
                      type = "<=") { #type = determines the type of probability calculated
  #function purpose:
  #uses dpois and ppois to conditionally return the correct probability

  if(type == "="){
    #P(X=x), calculates the probability of exactly observing x occurrences within a poisson distribution
    prob = dpois(x = x, lambda = lambda)
  } else if(type == "!="){
    #P(X!=x), calculates the probability of observing all occurrences except x within a poisson distribution
    prob = 1 - dpois(x = x, lambda = lambda)
  } else if(type == ">") {
    #P(X>x), calculates the probability of observing more than x occurrences within a poisson distribution
    prob = 1 - ppois(q = x, lambda = lambda)
  } else if(type == "<") {
    #P(X<x), calculates the probability of observing less than x occurrences within a poisson distribution
    prob = ppois(q = x - 1, lambda = lambda)
  } else if(type == ">=") {
    #P(X>=x), calculates the probability of observing at least x occurrences within a poisson distribution
    prob = 1 - ppois(q = x - 1, lambda = lambda)
  } else if(type == "<=") {
    #P(X<=x), calculates the probability of observing at most x occurrences within a poisson distribution
    prob = ppois(q = x, lambda = lambda)
  } else {
    #If parameters entered are incorrect, stop the function
    stop("Enter valid parameters (numericals for x and lambda, inequality for type).")
  }

  return(prob) #return the probability
}
```

2. Write a `beta.prob()` function that computes $P(X = x)$, $P(X \neq x)$, $P(X < x)$, $P(X \leq x)$, $P(X > x)$, and $P(X \geq x)$ for a beta distribution. Enable the user to specify the shape parameters α and β.

```r
beta.prob <- function(x,           #x = number of event/occurences
                      alpha,       #alpha = the alpha parameter
                      beta,        #beta = the beta parameter
                      type = "<=") { #type = the type of probability to be calculated
  #function purpose:
  #Use dbeta and pbeta to conditionally return the correct probability

  if(type == "="){
    #P(X=x), calculates the probability of exactly observing x occurrences within a beta distribution
    #(This js always 0)
    prob = 0
  } else if(type == "!="){
    #P(X!=x), calculates the probability of observing all occurrences except x within a beta distribution
    #(This is always 1)
    prob = 1
  } else if(type == ">") {
    #P(X>x), calculates the probability of observing more than x occurrences within a beta distribution
    prob = 1 - pbeta(q = x, shape1 = alpha, shape2 = beta)
  } else if(type == "<") {
    #P(X<x), calculates the probability of observing less than x occurrences within a beta distribution
    prob = pbeta(q = x, shape1 = alpha, shape2 = beta)
  } else if(type == ">=") {
    #P(X>=x), calculates the probability of observing at least x occurrences within a beta distribution
    prob = 1 - pbeta(q = x, shape1 = alpha, shape2 = beta)
  } else if(type == "<=") {
    #P(X<=x), calculates the probability of observing at most x occurrences within a beta distribution
    prob = pbeta(q = x, shape1 = alpha, shape2 = beta)
  } else {
    #If parameters entered are incorrect, stop the function
    stop("Enter valid parameters (numericals for x; alpha; and beta, inequality for type).")
  }

  return(prob) #return the probability
}
```