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.
   Solution: Below is the code for computing various probabilities for the Poisson Distribution. The `prob` parameter was removed as a parameter of the function, as it would only be used in calculations involving `qpois` (inverse CDF), which was not used in the function. The default type was set as $\leq$. λ was passed in directly as a parameter for the Poisson Distribution. λ must be a non-negative number, and the support of `x` is all non-negative numbers, so $x \geq 0$. It's key to note that the Poisson Distribution is a discrete distribution.

```r
# x >= 0
# lambda >= 0
pois.prob <- function(x, lambda, type="<="){
  if (type == "="){
    # definition of PMF
    return(dpois(x, lambda))
  }
  if (type == "!="){
    # complement rule
    return(1-dpois(x, lambda))
  }
  if(type == "<="){
    # definition of CDF
    return(ppois(x, lambda))
  }
  if (type == "<"){
    # P(X < x) = P(X <= x-1)
    return(ppois(x-1, lambda))
  }
  if (type == ">="){
    # P(X >= x) = 1 - P(X <= x-1)
    return(1-ppois(x-1, lambda))
  }
  if (type == ">"){
    # P(X > x) = 1 - P(X <= x)
    return(1-ppois(x, lambda))
  }
}
```

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.
   Solution: Below is the code for computing various probabilities for the Beta Distribution. The `prob` parameter was removed as a parameter of the function, as it would only be used in calculations involving `qbeta` (inverse CDF), which was not used in the function. The default type was set as $\leq$. α and β were passed in directly as a parameters for the Beta Distribution. α and β must be non-negative numbers, and $0 \leq x \leq 1$. $P(X = x)$ and $P(X \neq)$ were hardcoded into the function, as they have the same answer for any x. It's key to note that the Beta Distribution is a continuous distribution.

```r
# 0 <= x <= 1
# alpha, beta > 0
beta.prob <- function(x, alpha, beta, type="<="){
  if (type == "="){
    # P(X = x) = 0 for all x
    return(0)
  }
  if (type == "!="){
    # P(X != x) = 1 for all x
    return(1)
  }
  if(type == "<" | type == "<="){
    # P(X < x) = P(X <= x) for continuous distributions
    # definition of pbeta
    return(pbeta(x, alpha, beta))
  }
  if (type == ">" | type == ">="){
    # P(X > x) = P(X >= x) = 1 - P(X < x)for continouous distributions
    return(1 - pbeta(x, alpha, beta))
  }
}
```