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`, the inverse CDF, which was not used in the function. λ was passed in directly as a parameter for the Poisson Distribution, and statements were added to make sure λ was ≥ 0 and `x` was in the support.

```
pois.prob <- function(x, lambda, type="<="){
  #
  if (lambda < 0){
    # lambda must be <= 0
    lambda = 0
  }
  if (x < 0){
    # x must be <= 0
    x = 0
  }
  if (type == "="){
    # definition of PMF
    return(dpois(x, lambda))
  }
  if (type == "!="){
    # complement rule
    return(1-dpois(x, lamda))
  }
  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.

```
beta.prob <- function(x, size, prob, type="<="){
  # Use dbeta and pbeta to conditionally return the correct probability
}
```