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.

```
pois.prob <- function(x, lambda, type){
  if(type == "="){
    #pmf of x
    prob = dpois(x, lambda)
  }
  else if(type == "!="){
    #1 - pmf of x
    prob = 1 - dpois(x, lambda)
  }
  else if(type == "<"){
    #cdf of x-1
    prob = ppois(x-1, lambda)
  }
  else if(type == "<="){
    #cdf of x
    prob = ppois(x, lambda)
  }
  else if(type == ">"){
    #1 - cdf of x
    prob = 1 - ppois(x, lambda)
  }
  else if(type == ">="){
    #1 - cdf of x-1
    prob = 1 - ppois(x-1, lambda)
  }
  else{
    print("Invalid input. Please enter a valid operator for the 'type' parameter.")
    return(0)
  }
  return(prob)
}
```

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, alpha, beta, type){
  if(type == "="){
    #For a continuous distribution, probability is zero at any given point
    prob = 0
  }
  else if(type == "!="){
    #The compliment of "="
    prob = 1
  }
  else if(type == "<" | type == "<="){
    #These two operators are equivalent for continuous distributions
    prob = pbeta(x, alpha, beta)
  }
  else if(type == ">" | type == ">="){
    #These two operators are equivalent for continuous distributions
    prob = 1 - pbeta(x, alpha, beta)
  }
  else{
    print("Invalid input. Please enter a valid operator for the 'type' parameter.")
    return(0)
  }
  return(prob)
}
```