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, lambda, type="<=") {

  # Initialize result variable
  result <- NA

  # Compute probabilities based on the type argument
  if (type == "=") {
    result <- dpois(x, lambda)  # P(X = x)
  } else if (type == "<") {
    result <- ppois(x - 1, lambda)  # P(X < x)
  } else if (type == "<=") {
    result <- ppois(x, lambda)  # P(X <= x)
  } else if (type == ">") {
    result <- 1 - ppois(x, lambda)  # P(X > x)
  } else if (type == ">=") {
    result <- 1 - ppois(x - 1, lambda)  # P(X >= x)
  }
  return(result)
}
```

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, alpha, beta, type="<=") {

  # Calculate the probability based on the specified type
  if (type == "=") {
    # P(X = x) is the PDF at x, which is 0 since beta distribution is continuous
    return(0)
  }
  else if (type == "<") {
    # P(X < x) is the CDF at x P(X=x)=0 so "<" = "<="
    return(pbeta(x, shape1 = alpha, shape2 = beta))
  }
  else if (type == "<=") {
    # P(X <= x) is the CDF at x
    return(pbeta(x, shape1 = alpha, shape2 = beta))
  }
  else if (type == ">") {
    # P(X > x) is 1 minus the CDF at x
    return(1 - pbeta(x, shape1 = alpha, shape2 = beta))
  }
  else if (type == ">=") {
    # P(X >= x) is 1 minus the CDF at x since ">=" and ">" are equal
    return(1 - pbeta(x, shape1 = alpha, shape2 = beta))
  }
}
```