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.

```
suppressPackageStartupMessages(library(tidyverse))
pois_probabilities <- function(x, lambda, type = "<=") {
  prob <- case_when(
    type == "="  ~ dpois(x, lambda),  # P(X = x)
    type == "!=" ~ 1 - dpois(x, lambda),  # P(X \neq x)
    type == "<"  ~ ppois(x - 1, lambda),  # P(X < x)
    type == "<=" ~ ppois(x, lambda),  # P(X \leq x)
    type == ">"  ~ ppois(x, lambda, lower.tail = FALSE),  # P(X > x)
    type == ">=" ~ ppois(x - 1, lambda, lower.tail = FALSE)  # P(X \geq x)
  )
  return(tibble(Type = type, Probability = prob))
}
pois_probabilities(3, lambda = 5, type = ">=")


## # A tibble: 1 x 2
##   Type  Probability
##   <chr>       <dbl>
## 1 >=          0.875
```

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_probabilities <- function(x, alpha, beta, type = "<=") {
    prob <- case_when(
      type == "="  ~ 0,  #always = 0 for continuous distribution
      type == "!=" ~ 1, #always = 1 for continuous distribution
      type == "<"  ~ pbeta(x, alpha, beta),  # P(X < x)
      type == "<=" ~ pbeta(x, alpha, beta),  # P(X \leq x) (same for continuous)
      type == ">"  ~ pbeta(x, alpha, beta, lower.tail = FALSE),  # P(X > x)
      type == ">=" ~ pbeta(x, alpha, beta, lower.tail = FALSE)  # P(X \geq x)
    )

  return(tibble(Type = type, Probability = prob))
}

beta_probabilities(0.3, alpha = 2, beta = 5)


## # A tibble: 1 x 2
##   Type  Probability
##   <chr>       <dbl>
## 1 <=          0.580
```