CHALLENGE: PROB CALCULATOR

Write a function that calculates $P(x_1 \leq X \leq x_2)$ where X follows a normal distribution with parameters μ_x, σ_x. The function input is a matrix with four columns. The first column refers to x_1, the second to x_2, the third to μ_x, the fourth to σ_x. Each row can contain different values. The output of the function is a vector having the same length as there are rows in the matrix.

##Testing the prob calculator:

```
my.value.matrix <- matrix(nrow=2, ncol=4)
my.value.matrix[1,] <- c(-1, 1, 0, 1)
my.value.matrix[2,] <- c(-1, 2, 0, 1)
colnames(my.value.matrix) <- c("x1", "x2", "mean", "sigma")
my.value.matrix
```

```
##      x1 x2 mean sigma
## [1,] -1  1    0     1
## [2,] -1  2    0     1
```

```
my.prob.calc(my.value.matrix)
```

```
## [1] 0.6826895 0.8185946
```

##Solution

```
my.prob.calc <- function(my.matrix) {
  #step 1: obtain a vector of Prob(X <= x_1)
  my.lower.cum.prob <- pnorm(my.matrix[,1], mean=my.matrix[,3], sd=my.matrix[,4])
  #step 2: obtain a vector of Prob(X <= x_2)
  my.upper.cum.prob <- pnorm(my.matrix[,2], mean=my.matrix[,3], sd=my.matrix[,4])
  #step 3: calculate difference
  my.result <- my.upper.cum.prob - my.lower.cum.prob
  return(my.result)

}
```

##Improved code using names

```
my.prob.calc2 <- function(my.matrix) {
  my.lower.cum.prob <- pnorm(my.matrix[,'x1'], mean=my.matrix[,'mean'], sd=my.matrix[,'sigma'])
  my.upper.cum.prob <- pnorm(my.matrix[,'x2'], mean=my.matrix[,'mean'], sd=my.matrix[,'sigma'])
  my.result <- my.upper.cum.prob - my.lower.cum.prob
  return(my.result)
  }
```

```
my.prob.calc2(my.value.matrix)
```

```
## [1] 0.6826895 0.8185946
```