Number To Word Function for European System

```
numbers2words <- function(x){

  # HELPER FUNCTION
  helper <- function(x){
    # Convert the number into digits and reverse the same
    digits <- rev(strsplit(as.character(x), "")[[1]])

    # Count the number of Digits
    nDigits <- length(digits)

    # If number of Digits is 1, then the number is a 1-digit number
    if (nDigits == 1) as.vector(ones[digits])

    # If number of Digits is 2, then the number is a 2-digit number
    else if (nDigits == 2) {
      # If number < 19, then the number is between 10 and 19
      if (x <= 19) as.vector(teens[digits[1]])

      # Else, the number is between 20 and 99
      else {
        # If last digit is "0", then do not write "Zero" in the end
        if (digits[1][1] == "0") {
          trim(paste(tens[digits[2]]))
        }
        else {
          trim(paste(tens[digits[2]],Recall(as.numeric(digits[1]))))
        }
      }
    }

    # If number of Digits is 3, then the number is a 3-digit number
    else if (nDigits == 3)
      trim(paste(ones[digits[3]], "hundred and",
                                  Recall(makeNumber(digits[2:1]))))

    # Else, it is a larger number
    else {
      # Find out if it a "Thousand" or "Million" or "Billion", etc
      nSuffix <- ((nDigits + 2) %/% 3) - 1

      # If the number if too large to handle, quit the program
      if (nSuffix > length(suffixes))
          stop(paste(x, "is too large!"))

      # Else, form the words in thousands at a time
      trim(paste(Recall(makeNumber(digits[
        nDigits:(3*nSuffix + 1)])),
        suffixes[nSuffix],
```

```
        Recall(makeNumber(digits[(3*nSuffix):1])))
    }
  }

  # TRIM FUNCTION
  trim <- function(text){
    #Tidy leading/trailing whitespace, space before comma
    text=gsub("^\ ", "", gsub("\ *$", "", gsub("\ ,",",",text)))

    #Clear any trailing " and"
    gsub(" and$","",text)
  }

  # MAKENUMBER FUNCTION
  makeNumber <- function(...) as.numeric(paste(..., collapse=""))

  #Disable scientific notation
  opts <- options(scipen=100)
  on.exit(options(opts))

  ones <- c("zero", "one", "two", "three", "four", "five", "six",
            "seven", "eight", "nine")
  names(ones) <- 0:9

  teens <- c("ten", "eleven", "twelve", "thirteen", "fourteen",
             "fifteen", "sixteen", " seventeen", "eighteen",
             "nineteen")
  names(teens) <- 0:9

  tens <- c("twenty", "thirty", "forty", "fifty", "sixty",
            "seventy", "eighty", "ninety")
  names(tens) <- 2:9

  suffixes <- c("thousand", "million", "billion", "trillion")

  #Remove the decimals if any
  x <- round(x)

  #Call the HELPER Function
  if (length(x) > 1) return(trim(sapply(x, helper)))

  helper(x)
}
```

Number To Word Function for Indian System

```
numbers2words <- function(x){
  helper <- function(x){
    if (x < 0) { print(paste(x, "is negative!")); return }

    digits <- rev(strsplit(as.character(x), "")[[1]])
    nDigits <- length(digits)
    if (nDigits == 1) as.vector(ones[digits])
    else if (nDigits == 2)
      convert2DigitNumbers(x)
    else if (nDigits == 3)
      convert3DigitNumbers(x)
    else if (nDigits == 4 || nDigits == 5)
      convertThousands(x)
    else if (nDigits == 6 || nDigits == 7)
      convertLakhs(x)
    else if (nDigits == 8 || nDigits == 9)
      convertCrores(x)
    else
      trim(paste(
             numbers2words(floor(x/10000000)), "crore",
             convertCrores(x %% 10000000)))
  }

  convert2DigitNumbers <- function(x) {
    if ( x > 0 && x <= 99 ) {
      digits2DigitNumber <- rev(strsplit(as.character(x), "")[[1]])
      if (x <= 9) as.vector(ones[digits2DigitNumber])
      else if (x <= 19) as.vector(teens[digits2DigitNumber[1]])
      else {
        if (digits2DigitNumber[1][1] == "0") {
          trim(paste(tens[digits2DigitNumber[2]]))
        }
        else {
          trim(paste(
               tens[digits2DigitNumber[2]],
               as.vector(ones[digits2DigitNumber[1]])))
        }
      }
    }
  }

  convert3DigitNumbers <- function(x) {
    if ( x > 0 && x <= 999 ) {
      if ( x < 100 ) convert2DigitNumbers(x)
      else {
        digits3DigitNumber <-
                 rev(strsplit(as.character(x), "")[[1]])
        trim(paste(
         ones[digits3DigitNumber[3]], "hundred and",
         convert2DigitNumbers(makeNumber(digits3DigitNumber[2:1]))))
```

```
    }
  }
}

convertThousands <- function(x) {
  if ( x > 0 && x <= 99999 ) {
    if ( x < 1000 ) convert3DigitNumbers(x)
    else {
      digitsThousands <- rev(strsplit(as.character(x), "")[[1]])
      if ( x <= 9999 )
        trim(paste(
          ones[digitsThousands[4]],
          "thousand",
          convert3DigitNumbers(makeNumber(digitsThousands[3:1]))))
      else
        trim(paste(
          convert2DigitNumbers(makeNumber(digitsThousands[5:4])),
          "thousand",
          convert3DigitNumbers(makeNumber(digitsThousands[3:1]))))
    }
  }
}

convertLakhs <- function(x) {
  if ( x > 0 && x <= 9999999 ) {
    if ( x < 100000 ) convertThousands(x)
    else {
      digitsLakhs <- rev(strsplit(as.character(x), "")[[1]])
      if ( x <= 999999 )
        trim(paste(
                  ones[digitsLakhs[6]], "lakh",
                  convertThousands(makeNumber(digitsLakhs[5:1]))))
      else
        trim(paste(
              convert2DigitNumbers(makeNumber(digitsLakhs[7:6])),
              "lakh",
              convertThousands(makeNumber(digitsLakhs[5:1]))))
    }
  }
}

convertCrores <- function(x) {
  if ( x > 0 && x <= 999999999 ) {
    if ( x < 10000000 ) convertLakhs(x)
    else {
      digitsCrores <- rev(strsplit(as.character(x), "")[[1]])
      if ( x <= 99999999 )
        trim(paste(ones[digitsCrores[8]], "crore",
                    convertLakhs(makeNumber(digitsCrores[7:1]))))
      else
        trim(paste(
              convert2DigitNumbers(makeNumber(digitsCrores[9:8])),
              "crore",
              convertLakhs(makeNumber(digitsCrores[7:1]))))
    }
  }
```

```
  }

  trim <- function(text) {
    #Tidy leading/trailing whitespace, space before comma
    text=gsub("^\ ", "", gsub("\ *$", "", gsub("\ ,",",",text)))

    #Clear any trailing " and"
    gsub(" and$","",text)
  }

  makeNumber <- function(...) as.numeric(paste(..., collapse=""))

  #Disable scientific notation
  opts <- options(scipen=100)
  on.exit(options(opts))

  ones <- c("zero", "one", "two", "three", "four", "five", "six",
            "seven", "eight", "nine")
  names(ones) <- 0:9

  teens <- c("ten", "eleven", "twelve", "thirteen", "fourteen",
             "fifteen", "sixteen", " seventeen", "eighteen",
             "nineteen")
  names(teens) <- 0:9

  tens <- c("twenty", "thirty", "forty", "fifty", "sixty",
            "seventy", "eighty", "ninety")
  names(tens) <- 2:9

  x <- round(x)
  if (length(x) > 1 && x >= 0) return(trim(sapply(x, helper)))

  helper(x)
}
```