```
# Navdeep Gill Assigment 2

# 1

# Bubble Sort

sort.b <- function(x) {
    if (length(x) < 2) {
        stop("vector is not long enough")
    }
    if (!is.vector(x)) {
        stop("parameter must be a vector")
    }
    if (!is.numeric(x)) {
        stop("parameter must be numeric")
    }

    n = length(x)
    v = x

    for (j in 1:(n - 1)) {
        for (i in 1:(n - j)) {
            if (v[i + 1] < v[i]) {
                t = v[i + 1]
                v[i + 1] = v[i]
                v[i] = t
            }
        }
    }
    print(v)
    x = v
}

# Test
x <- c(2, 1, 7, 9, 3, 6, 20, 30, 3, 5, 8, 6, 3)
sort.b(x)
```

```
##  [1]  1  2  3  3  3  5  6  6  7  8  9 20 30
```

```
# 2

# Straight Insertion Sort

sort.sis <- function(x, z) {
    if (!is.vector(x)) {
        stop("Parameter must be a vector")
    }
    if (!is.numeric(z)) {
        stop("Parameter must be a numeric")
    }
    n = length(x)
    y = numeric(n + 1)

    for (i in 1:n) {
        j = i
        while (x[j] <= z) {
            y <<- append(x, z, after = j)
            j = j + 1
        }
    }
}

# Test
x = seq(1:10)
z = 4
sort.sis(x, z)
y
```

```
##  [1]  1  2  3  4  4  5  6  7  8  9 10
```

```
# Merge Sort
merge.sort <- function(in1, in2) {

    if (!is.vector(in1)) {
        stop("Parameter must be a vector")
    }
    if (!is.vector(in2)) {
        stop("Parameter must be a vector")
    }
    if (!is.numeric(in1)) {
        stop("Parameter must be a numeric")
    }
    if (!is.numeric(in2)) {
        stop("Parameter must be a numeric")
    }

    end.vector <- c()

    while (length(in1) > 0 && length(in2) > 0) {
        if (in1[1] <= in2[1]) {
            end.vector <- c(end.vector, in1[1])
            in1 <- in1[-1]
        } else {
            end.vector <- c(end.vector, in2[1])
            in2 <- in2[-1]
        }
    }
    if (length(in1) > 0) {
        end.vector <- c(end.vector, in1)
    }
    if (length(in2) > 0) {
        end.vector <- c(end.vector, in2)
    }
    end.vector
}

# Test
x <- c(1, 2, 3, 4)
y <- c(1.5, 3, 5)
merge.sort(x, y)
```

```
## [1] 1.0 1.5 2.0 3.0 3.0 4.0 5.0
```