**Assignment 5.3**

**Problem Statement**

**1. Test whether two vectors are exactly equal (element by element).**

**vec1 = c(rownames(mtcars[1:15,]))**

**vec2 = c(rownames(mtcars[11:25,]))**

**Answer**:

vec1 = c(rownames(mtcars[1:15,]))

vec2 = c(rownames(mtcars[11:25,]))

is.element(vec1,vec2)

identical(vec1,vec2)

setequal(vec1,vec2)

vec1 %in% vec2

**# 2 Question: Sort the character vector in ascending order and descending order.**

**vec1 = c(rownames(mtcars[1:15,]))**

**vec2 = c(rownames(mtcars[11:25,])).**

**Output:**

vec1 = c(rownames(mtcars[1:15,]))

vec2 = c(rownames(mtcars[11:25,]))

vec1

sort(vec1, decreasing = T)

**3.What is the major difference between str() and paste() show an example.**

**Output :**

library(tidyverse)

library(stringr)

paste("foo", "bar")

paste0("foo", "bar")

## The function paste separates strings by spaces by default, while paste0 does not separate strings with spaces by default.

str\_c("foo", "bar")

str\_c("foo", NA)

paste("foo", NA)

paste0("foo", NA)

## Since str\_c does not separate strings with spaces by default it is closer in behavior to paste0.

str\_c("foo", "bar")

## However, str\_c and the paste function handle NA differently. The function str\_c propagates NA, if any argument is a missing value, it returns a missing value. This is in line with how the numeric R functions, e.g. sum, mean, handle missing values. However, the paste functions, convert NA to the string "NA" and then treat it as any other character vector.

str\_c("foo", NA)

paste("foo", NA)

paste0 ("foo", NA)

**4. Introduce a separator when concatenating the strings.**

**Answer:**

paste("Hello", "world", sep=" ")

x <- c("Hello", "World")

x

paste(x, collapse="--")

paste(x, "and some more", sep="|-|", collapse="--")