ASSIGNMENT 5.3

5. 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: Using == we can find the individual equal elements

vec1 = c(rownames(mtcars[1:15,]))
vec2 = c(rownames(mtcars[11:25,]))
vec1==vec2

Output:

```
> vec1 = c(rownames(mtcars[1:15,]))
> vec2 = c(rownames(mtcars[11:25,]))
> vec1==vec2
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[14] FALSE FALSE
>
```

2. Sort the character vector in ascending order and descending order

vec1 = c(rownames(mtcars[1:15,]))
vec2 = c(rownames(mtcars[11:25,]))

Answer: Using sort() function , we can find the ascending and descending order of the vectors

vec1 = c(rownames(mtcars[1:15,]))
vec2 = c(rownames(mtcars[11:25,]))
#ascending
sort(c(vec1,vec2))
#Descending

sort(c(vec1,vec2), decreasing = T)

Output:

```
> #ascending
> sort(c(vec1,vec2))
 [1] "AMC Javelin"          "Cadillac Fleetwood"  "Cadillac Fleetwood"
 [4] "Camaro Z28"           "Chrysler Imperial"   "Datsun 710"
 [7] "Dodge Challenger"     "Duster 360"          "Fiat 128"
[10] "Honda Civic"          "Hornet 4 Drive"      "Hornet Sportabout"
[13] "Lincoln Continental"  "Mazda RX4"           "Mazda RX4 Wag"
[16] "Merc 230"             "Merc 240D"           "Merc 280"
[19] "Merc 280C"            "Merc 280C"           "Merc 450SE"
[22] "Merc 450SE"           "Merc 450SL"          "Merc 450SL"
[25] "Merc 450SLC"          "Merc 450SLC"         "Pontiac Firebird"
[28] "Toyota Corolla"       "Toyota Corona"       "Valiant"
> #Descending
> sort(c(vec1,vec2), decreasing = T)
 [1] "Valiant"              "Toyota Corona"       "Toyota Corolla"
 [4] "Pontiac Firebird"     "Merc 450SLC"         "Merc 450SLC"
 [7] "Merc 450SL"           "Merc 450SL"          "Merc 450SE"
[10] "Merc 450SE"           "Merc 280C"           "Merc 280C"
[13] "Merc 280"             "Merc 240D"           "Merc 230"
[16] "Mazda RX4 Wag"        "Mazda RX4"           "Lincoln Continental"
[19] "Hornet Sportabout"    "Hornet 4 Drive"      "Honda Civic"
[22] "Fiat 128"             "Duster 360"          "Dodge Challenger"
[25] "Datsun 710"           "Chrysler Imperial"   "Camaro Z28"
[28] "Cadillac Fleetwood"   "Cadillac Fleetwood"  "AMC Javelin"
>
```

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

Answer: There is **no separator** in the *str_c function* and we can only add the sep="", if we need to between the strings,

Whereas, *paste()* function has a **space separator** by default.

Example: Using str_c()

```
        str_c("Data","Science","with","Python","and","R")
[1] "DataSciencewithPythonandR"
```

Using paste()

```
        paste("Data","Science","with","Python","and","R")
[1] "Data Science with Python and R"
```

As seen, the **paste()** has space has separator, but **str_c()** doesn't.

4. Introduce a separator when concatenating the strings.

Answer: Using str_c or paste() with a separator.

Output: