

Problem Statement

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

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

Answer :

```
> isTRUE(all.equal(vec1,vec2))
[1] FALSE
> identical(vec1,vec2)
[1] FALSE
> all.equal(vec1,vec2)
[1] "Lengths (15, 23) differ (string compare on first 15)"
[2] "15 string mismatches"

>
```

The output of the R-Script (from Console window) is given as follows:

```
> vec1 = c(rownames(mtcars[1:15,]))
> vec2 = c(rownames(mtcars[11:25,]))
>
> isTRUE(all.equal(vec1,vec2)) # returns true/false
[1] FALSE
> identical(vec1,vec2)        # returns true/false
[1] FALSE
> all.equal(vec1,vec2)        # returns number of differences
[1] "15 string mismatches"
> |
```

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

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

Answer :

sort(vec1) ascending order

sort(vec1,decreasing = TRUE) descending order

sort(vec2) ascending order

sort(vec2,decreasing = TRUE) descending order

```
> sort(vec1)
[1] "Cadillac Fleetwood" "Datsun 710"      "Duster 360"      "Hornet 4 Drive"
[5] "Hornet Sportabout"  "Mazda RX4"       "Mazda RX4 Wag"   "Merc 230"
[9] "Merc 240D"          "Merc 280"        "Merc 280C"       "Merc 450SE"
[13] "Merc 450SL"         "Merc 450SLC"     "Valiant"
> sort(vec1,decreasing = TRUE)
[1] "Valiant"          "Merc 450SLC"     "Merc 450SL"      "Merc 450SE"
[5] "Merc 280C"        "Merc 280"       "Merc 240D"       "Merc 230"
```

```

[9] "Mazda RX4 Wag"      "Mazda RX4"      "Hornet Sportabout" "Hornet 4 Drive"
[13] "Duster 360"        "Datsun 710"     "Cadillac Fleetwood"
> sort(vec2)
[1] "AMC Javelin"      "Cadillac Fleetwood" "Camaro Z28"
[4] "Chrysler Imperial" "Dodge Challenger"  "Ferrari Dino"
[7] "Fiat 128"         "Fiat X1-9"         "Ford Pantera L"
[10] "Honda Civic"      "Lincoln Continental" "Lotus Europa"
[13] "Maserati Bora"    "Merc 280"          "Merc 280C"
[16] "Merc 450SE"       "Merc 450SL"        "Merc 450SLC"
[19] "Pontiac Firebird" "Porsche 914-2"     "Toyota Corolla"
[22] "Toyota Corona"   "Volvo 142E"
> sort(vec2,decreasing = TRUE)
[1] "Volvo 142E"      "Toyota Corona"    "Toyota Corolla"
[4] "Porsche 914-2"   "Pontiac Firebird" "Merc 450SLC"
[7] "Merc 450SL"      "Merc 450SE"       "Merc 280C"
[10] "Merc 280"        "Maserati Bora"    "Lotus Europa"
[13] "Lincoln Continental" "Honda Civic"     "Ford Pantera L"
[16] "Fiat X1-9"       "Fiat 128"        "Ferrari Dino"
[19] "Dodge Challenger" "Chrysler Imperial" "Camaro Z28"
[22] "Cadillac Fleetwood" "AMC Javelin"

```

```

Console C:/Users/Vikram/Desktop/Acad/
> isTRUE(all.equal(vec1,vec2))
[1] FALSE
> identical(vec1,vec2)
[1] FALSE
> all.equal(vec1,vec2)
[1] "Lengths (15, 23) differ (string compare on first 15)"
[2] "15 string mismatches"
> sort(vec1)
[1] "Cadillac Fleetwood" "Datsun 710"      "Duster 360"      "Hornet 4 Drive"
[5] "Hornet Sportabout" "Mazda RX4"      "Mazda RX4 Wag"   "Merc 230"
[9] "Merc 240D"         "Merc 280"       "Merc 280C"       "Merc 450SE"
[13] "Merc 450SL"        "Merc 450SLC"    "Valiant"
> sort(vec1,decreasing = TRUE)
[1] "Valiant"          "Merc 450SLC"     "Merc 450SL"      "Merc 450SE"
[5] "Merc 280C"        "Merc 280"       "Merc 240D"       "Merc 230"
[9] "Mazda RX4 Wag"    "Mazda RX4"      "Hornet Sportabout" "Hornet 4 Drive"
[13] "Duster 360"       "Datsun 710"     "Cadillac Fleetwood"
> sort(vec2)
[1] "AMC Javelin"      "Cadillac Fleetwood" "Camaro Z28"
[4] "Chrysler Imperial" "Dodge Challenger"  "Ferrari Dino"
[7] "Fiat 128"         "Fiat X1-9"         "Ford Pantera L"
[10] "Honda Civic"      "Lincoln Continental" "Lotus Europa"
[13] "Maserati Bora"    "Merc 280"          "Merc 280C"
[16] "Merc 450SE"       "Merc 450SL"        "Merc 450SLC"
[19] "Pontiac Firebird" "Porsche 914-2"     "Toyota Corolla"
[22] "Toyota Corona"   "Volvo 142E"
> sort(vec2,decreasing = TRUE)
[1] "Volvo 142E"      "Toyota Corona"    "Toyota Corolla"
[4] "Porsche 914-2"   "Pontiac Firebird" "Merc 450SLC"
[7] "Merc 450SL"      "Merc 450SE"       "Merc 280C"
[10] "Merc 280"        "Maserati Bora"    "Lotus Europa"
[13] "Lincoln Continental" "Honda Civic"     "Ford Pantera L"
[16] "Fiat X1-9"       "Fiat 128"        "Ferrari Dino"
[19] "Dodge Challenger" "Chrysler Imperial" "Camaro Z28"
[22] "Cadillac Fleetwood" "AMC Javelin"
> |

```

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

Answer :

str() gives the class of variable, number of values and the elements whereas paste() prints or displays the actual elements .

For example:

str(mtcars\$mpg) gives the class of mtcars\$mpg as num, number of values as 32(1:32) and the elements as 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...

whereas **paste(mtcars\$mpg)** prints the actual elements present in mtcars\$mpg.

```
> str(mtcars$mpg)
```

```

num [1:32] 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
> paste(mtcars$mpg)
[1] "21" "21" "22.8" "21.4" "18.7" "18.1" "14.3" "24.4" "22.8" "19.2" "17.8" "16.4"
[13] "17.3" "15.2" "10.4" "10.4" "14.7" "32.4" "30.4" "33.9" "21.5" "15.5" "15.2" "13.3"
[25] "19.2" "27.3" "26" "30.4" "15.8" "19.7" "15" "21.4"
>

```

D. Introduce a separator when concatenating the strings

```

> paste(rownames(mtcars[1,]), rownames(mtcars[2,]), sep = " ")
[1] "Mazda RX4 Mazda RX4 Wag"
> paste(rownames(mtcars[1,]), rownames(mtcars[2,]), sep = " ")
[1] "Mazda RX4 Mazda RX4 Wag"
> paste(rownames(mtcars[1,]), rownames(mtcars[4,]), sep = ",")
[1] "Mazda RX4,Hornet 4 Drive"
> paste(rownames(mtcars[2,]), rownames(mtcars[1,]), sep = "--")
[1] "Mazda RX4 Wag--Mazda RX4"
> paste(rownames(mtcars[3,]), rownames(mtcars[10,]), sep = "$")
[1] "Datsun 710$Merc 280"
> paste("hello","world",sep="@ ")
[1] "hello @ world"
> paste("Assignment","5","3",sep="_")
[1] "Assignment_5_3"

```

The above R-script shows 6 examples where separators are introduced while concatenating the strings. The separators are introduced by setting the parameter “sep” as “ ” or “,” or “--” or “\$” or “@” or “_” or any other separator .

- **paste(rownames(mtcars[1,]), rownames(mtcars[2,]), sep = " ")** introduces a separator ,a single blank " " between the strings rownames(mtcars[1,]) and rownames(mtcars[2,]).
- **paste(rownames(mtcars[1,]), rownames(mtcars[4,]), sep = ",")** introduces a separator comma ",", " between the strings rownames(mtcars[1,]) and rownames(mtcars[4,]).
- **paste(rownames(mtcars[2,]), rownames(mtcars[1,]), sep = "--")** introduces a separator "--" between the strings rownames(mtcars[2,]) and rownames(mtcars[1,]).
- **paste(rownames(mtcars[3,]), rownames(mtcars[10,]), sep = "\$")** introduces a separator dollar "\$" between the strings rownames(mtcars[3,]) and rownames(mtcars[10,]).
- **paste("hello","world",sep="@")** introduces a separator "@" between the strings “hello” and “world”
- **paste("Assignment","5","3",sep="_")** introduces a separator underscore "_" between the strings "Assignment", "5" and "3".