# 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,]))

|  |
| --- |
| > vec1 = c(rownames(mtcars[1:15,]))  > vec1  [1] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710"  [4] "Hornet 4 Drive" "Hornet Sportabout" "Valiant"  [7] "Duster 360" "Merc 240D" "Merc 230"  [10] "Merc 280" "Merc 280C" "Merc 450SE"  [13] "Merc 450SL" "Merc 450SLC" "Cadillac Fleetwood"  > vec2 = c(rownames(mtcars[11:25,]))  > vec2  [1] "Merc 280C" "Merc 450SE" "Merc 450SL"  [4] "Merc 450SLC" "Cadillac Fleetwood" "Lincoln Continental"  [7] "Chrysler Imperial" "Fiat 128" "Honda Civic"  [10] "Toyota Corolla" "Toyota Corona" "Dodge Challenger"  [13] "AMC Javelin" "Camaro Z28" "Pontiac Firebird"  > # returns true/false  > identical(vec1,vec2)  [1] FALSE  > identical(vec1,vec2)  [1] FALSE  > isTRUE(all.equal(vec1,vec2))  [1] FALSE  > setequal(vec1,vec2)  [1] FALSE  > # returns number of differences  > all.equal(vec1,vec2)  [1] "15 string mismatches"  > # Compare one by one  > vec1 %in% vec2  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE  [13] TRUE TRUE TRUE  > vec2 %in% vec1  [1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  [13] FALSE FALSE FALSE |
|  |
| |  | | --- | | > | |

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

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

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

|  |
| --- |
| > vec1 = c(rownames(mtcars[1:15,]))  > vec1  [1] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710"  [4] "Hornet 4 Drive" "Hornet Sportabout" "Valiant"  [7] "Duster 360" "Merc 240D" "Merc 230"  [10] "Merc 280" "Merc 280C" "Merc 450SE"  [13] "Merc 450SL" "Merc 450SLC" "Cadillac Fleetwood"  > vec2 = c(rownames(mtcars[11:25,]))  > vec2  [1] "Merc 280C" "Merc 450SE" "Merc 450SL"  [4] "Merc 450SLC" "Cadillac Fleetwood" "Lincoln Continental"  [7] "Chrysler Imperial" "Fiat 128" "Honda Civic"  [10] "Toyota Corolla" "Toyota Corona" "Dodge Challenger"  [13] "AMC Javelin" "Camaro Z28" "Pontiac Firebird"  > #sort in ascending order by default  > sort(vec1)  [1] "Cadillac Fleetwood" "Datsun 710" "Duster 360"  [4] "Hornet 4 Drive" "Hornet Sportabout" "Mazda RX4"  [7] "Mazda RX4 Wag" "Merc 230" "Merc 240D"  [10] "Merc 280" "Merc 280C" "Merc 450SE"  [13] "Merc 450SL" "Merc 450SLC" "Valiant"  > sort(vec2)  [1] "AMC Javelin" "Cadillac Fleetwood" "Camaro Z28"  [4] "Chrysler Imperial" "Dodge Challenger" "Fiat 128"  [7] "Honda Civic" "Lincoln Continental" "Merc 280C"  [10] "Merc 450SE" "Merc 450SL" "Merc 450SLC"  [13] "Pontiac Firebird" "Toyota Corolla" "Toyota Corona"  > #sort in descending order  > sort(vec1,decreasing = T)  [1] "Valiant" "Merc 450SLC" "Merc 450SL"  [4] "Merc 450SE" "Merc 280C" "Merc 280"  [7] "Merc 240D" "Merc 230" "Mazda RX4 Wag"  [10] "Mazda RX4" "Hornet Sportabout" "Hornet 4 Drive"  [13] "Duster 360" "Datsun 710" "Cadillac Fleetwood"  > sort(vec2,decreasing = T)  [1] "Toyota Corona" "Toyota Corolla" "Pontiac Firebird"  [4] "Merc 450SLC" "Merc 450SL" "Merc 450SE"  [7] "Merc 280C" "Lincoln Continental" "Honda Civic"  [10] "Fiat 128" "Dodge Challenger" "Chrysler Imperial"  [13] "Camaro Z28" "Cadillac Fleetwood" "AMC Javelin" |
|  |
| |  | | --- | |  | |

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

#str()

#display the structure of an arbitrary object

#ex:

#it comactly display the internal structure of an R object

#a diagnostic function and an alternative to summary

#it displays many useful things

a<- c("1","2","3","hey")

a

str(a)

#paste()

#used for Concatenate Strings

#paste (., sep = " ", collapse = NULL)

#ex:

x <- c('My.name.is.Raman.Kumar','learning.Data.Analytics')

x

con\_str<- paste(x[1],x[2],sep = ",")

con\_str

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

str(vec1)

#returns the value, class and number of elements

paste(vec1)

#returns the value only(or just prints)

#similarly

mode(str(vec1))

mode(paste(vec1))

class(str(vec1))

class(paste(vec1))

1. Introduce a separator when concatenating the strings.

2

x <- c('My.name.is.RAMAN.KUMAR','learning.Data.Analytics')

x

y<- c(gsub(".","-",x,fixed = TRUE))

y

#use of paste() function to concatenate strings

con\_str<- paste(y[1],y[2],sep = ",")

con\_str

#another example

x<-c("a","b","c")

y<-c("A","B","C")

paste(x,y)

paste(x,y,sep = ",")

paste(x,y,sep = "-")

|  |
| --- |
| > x <- c('My.name.is.RAMAN.KUMAR','learning.Data.Analytics')  > x  [1] "My.name.is.RAMAN.KUMAR" "learning.Data.Analytics"  > y<- c(gsub(".","-",x,fixed = TRUE))  > y  [1] "My-name-is-RAMAN-KUMAR" "learning-Data-Analytics"  > #use of paste() function to concatenate strings  > con\_str<- paste(y[1],y[2],sep = ",")  > con\_str  [1] "My-name-is-RAMAN-KUMAR,learning-Data-Analytics"  > x<-c("a","b","c")  > y<-c("A","B","C")  > paste(x,y)  [1] "a A" "b B" "c C"  > paste(x,y,sep = ",")  [1] "a,A" "b,B" "c,C"  > paste(x,y,sep = "-")  [1] "a-A" "b-B" "c-C" |
|  |
| |  | | --- | | > | |