Task 1:

1. States = rownames(US Arrests)
   Get States names with ‘w’.
   Get States names with ‘W’.

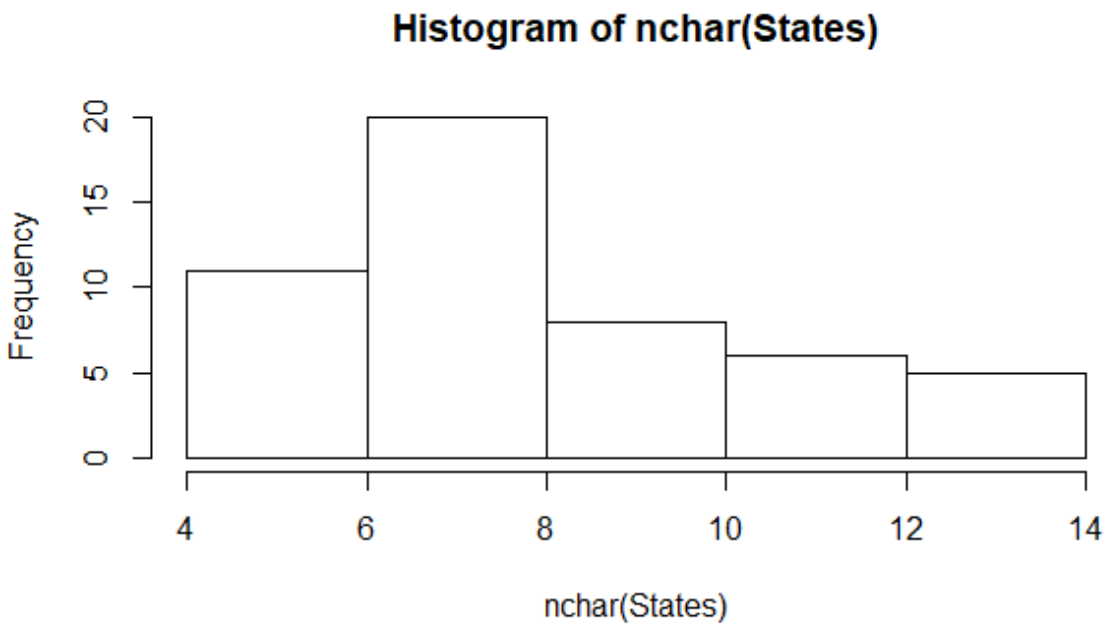
Ans:-

```
> States = rownames(USArrests)
> grep(pattern = "[w]", States, value = TRUE)

[1] "Delaware"      "Hawaii" "Iowa"  "New Hampshire" "New Jersey"
[6] "New Mexico"    "New York"

> grep(pattern = "[W]", States, value = TRUE)

[1] "Washington"    "West Virginia" "Wisconsin"     "Wyoming"
```

2. Prepare a Histogram of the number of characters in each US state.

Ans:-

```
> States=rownames(USArrests)
> hist(nchar(States))
```



Task 2:

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

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

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

Ans:-

```
> 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,]))
```

Ans:-

```
#Ascending Order

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

 [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, decreasing = FALSE)

 [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"

> #Decending Order
> vec1 = c(rownames(mtcars[1:15,]))
> vec2 = c(rownames(mtcars[11:25,]))
> sort(vec1, decreasing = TRUE)

[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 = TRUE)

 [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"
```

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

Ans:- str() is used to display the data type of an argument but paste() converts all the arguments into string and concatenates them.

For example:

```
str():
> str("a")
chr "a"

paste()
> paste("a", 1)
[1] "a 1"
```

4. Introduce a separator when concatenating the strings.

Ans:- paste() converts its arguments to character strings, and concatenates them. If the arguments are vectors, they are concatenated term-by-term to give a character vector result. If a value is specified for *collapse*, the values in the result are then concatenated into a single string, with the elements being separated by the value of collapse.

For example:

```
> paste(1:5)
[1] "1" "2" "3" "4" "5"
> paste("A", 1:6, sep = "")
[1] "A1" "A2" "A3" "A4" "A5" "A6"
> paste("A", 1:6, sep = "", colapse=1)
[1] "A11" "A21" "A31" "A41" "A51" "A61"
> paste("Today is", date())
[1] "Today is Thu Apr 18 23:10:04 2019"
```