EXPT NO : 6	VECTORS, LISTS, MATRICES, ARRAYS, FACTORS
DATE : 19.02.25	

AIM:

To Implement vectors, lists, matrices, arrays, factors using r programming.

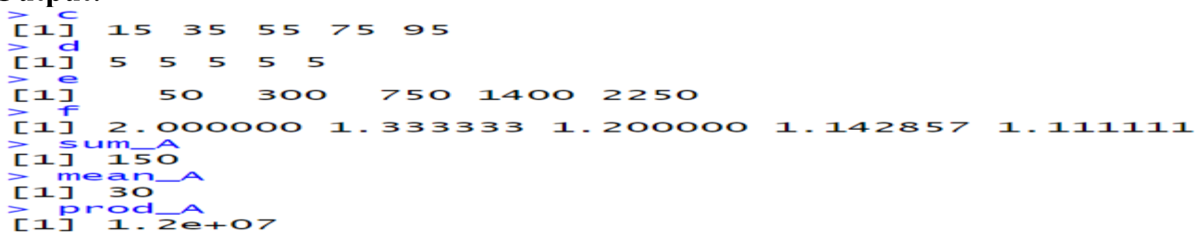
1. Vectors:

- **Create two numeric vectors A and B of length 5.**
- **Perform element-wise addition, subtraction, multiplication, and division.**
- **Find the sum, mean, and product of all elements in A.**
- **Sort vector B in ascending order.**
- **Find and print the maximum and minimum values in vector A.**

Code:

```
A <- c(10, 20, 30, 40, 50)
B <- c(5, 15, 25, 35, 45)
c <- A + B
d <- A - B
e <- A * B
f <- A / B
sum_A <- sum(A)
mean_A <- mean(A)
prod_A <- prod(A)
sorted_B <- sort(B)
max_A <- max(A)
min_A <- min(A)
c d e f
sum_A
mean_A
prod_A
max_A``
min_A
```

Output:



```
> c
[1]  15  35  55  75  95
> d
[1]  5  5  5  5  5
> e
[1]    50   300   750  1400  2250
> f
[1] 2.000000 1.333333 1.200000 1.142857 1.111111
> sum_A
[1] 150
> mean_A
[1] 30
> prod_A
[1] 1.2e+07
```

2. Lists:

• Create a list containing.

o A numeric vector of 5 elements

o A character vector with your first and last name

o A logical value (TRUE or FALSE)

• Print the list.

• Access and print the second element of the list.

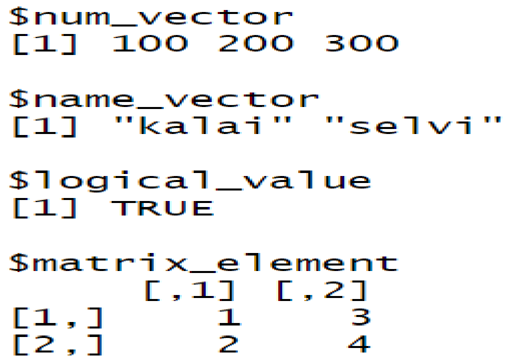
• Modify the first element of the list by replacing it with a new numeric vector of length3.

• Append a new element (a matrix of size 2×2) to the list.

Code:

```
my_list <- list(
  num_vector = c(10, 20, 30, 40, 50),
  name_vector = c("kalai", "selvi"),
  logical_value = TRUE  )
print(my_list)
print(my_list$name_vector)
my_list$num_vector <- c(100, 200, 300)
print(my_list)
my_list$matrix_element <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2)
print(my_list)
```

Output:



```
$num_vector
[1] 100 200 300

$name_vector
[1] "kalai" "selvi"

$logical_value
[1] TRUE

$matrix_element
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

3. Matrices:

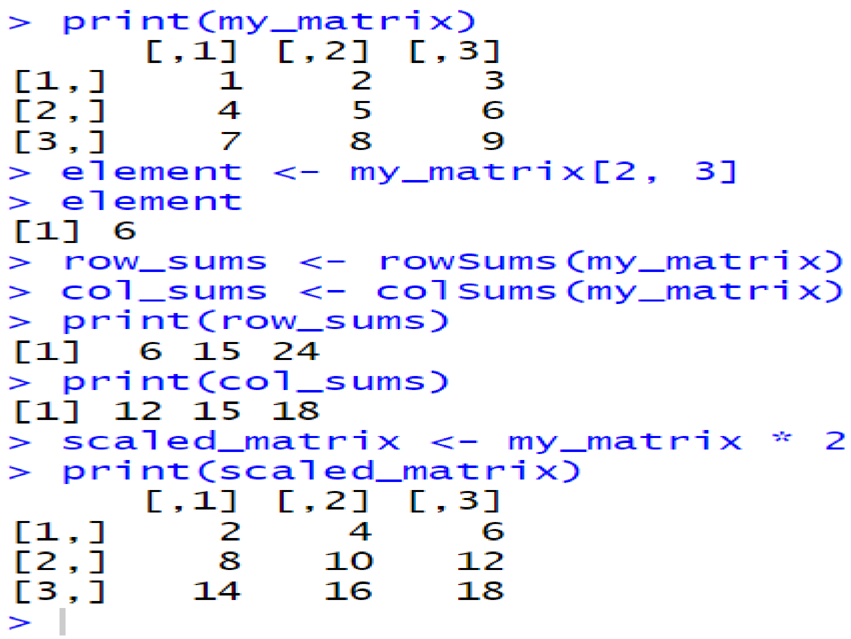
• Create a 3×3 matrix with numbers from 1 to 9 filled by row.

- **Display the matrix.**
- **Access and print the element at row 2, column 3.**
- **Find the row-wise and column-wise sums of the matrix.**
- **Multiply the matrix by a scalar value of 2.**

Code:

```
my_matrix <- matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE)
print("3×3 Matrix:")
print(my_matrix)
element <- my_matrix[2, 3]
element
row_sums <- rowSums(my_matrix)
col_sums <- colSums(my_matrix)
print(row_sums)
print(col_sums)
scaled_matrix <- my_matrix * 2
print(scaled_matrix)
```

Output:



```
> print(my_matrix)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> element <- my_matrix[2, 3]
> element
[1] 6
> row_sums <- rowSums(my_matrix)
> col_sums <- colSums(my_matrix)
> print(row_sums)
[1]  6 15 24
> print(col_sums)
[1] 12 15 18
> scaled_matrix <- my_matrix * 2
> print(scaled_matrix)
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    8   10   12
[3,]   14   16   18
> 
```

4. Arrays:

- **Create a 3×3×2 array with numbers from 1 to 18.**

• Print the entire array.

• Extract the first 3×3 matrix from the array.

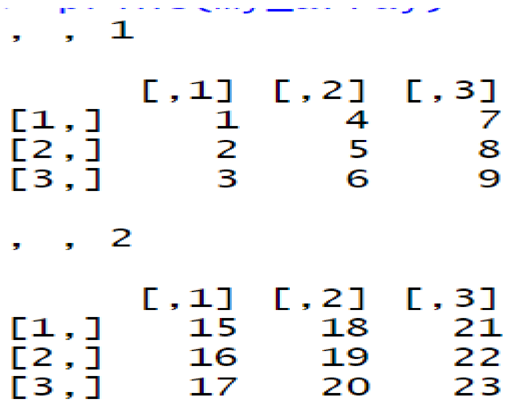
• Find and print the sum of all elements in the array.

• Modify the second matrix of the array by adding 5 to each element and print the updated array.

Code:

```
my_array <- array(1:18, dim = c(3, 3, 2))
print(my_array)
first_matrix <- my_array[, , 1]
print(first_matrix)
sum_elements <- sum(my_array)
sum_elements
my_array[, , 2] <- my_array[, , 2] + 5
print(my_array)
```

Output:



```
, , 1

     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

, , 2

     [,1] [,2] [,3]
[1,]   15   18   21
[2,]   16   19   22
[3,]   17   20   23
```

5. Factors:

• Create a factor variable representing the following colors: "Red", "Green", "Blue", "Red", "Green", "Blue", "Red".

• Display the factor and its levels.

• Find the number of occurrences of each level.

• Change the levels to "R", "G", "B" and print the modified factor.

• Convert the factor into a character vector and print it.

Code:

```
color_factor <- factor(c("Red", "Green", "Blue", "Red", "Green", "Blue", "Red"))
```

print(color_factor)

print(levels(color_factor))

print(table(color_factor))

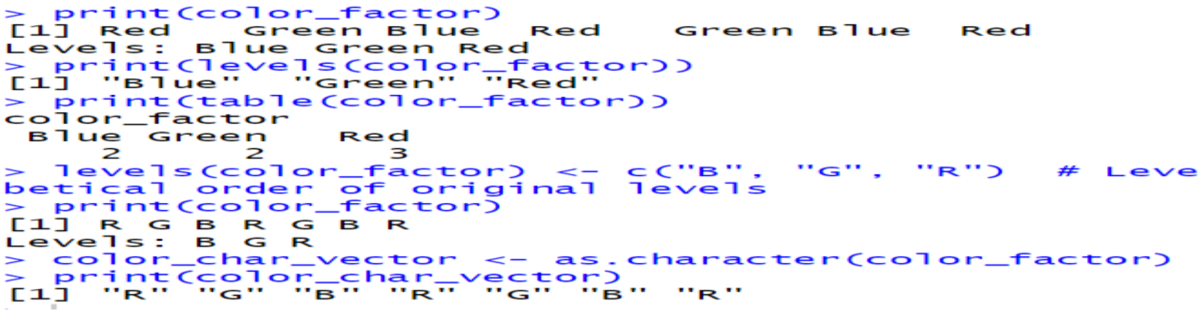
levels(color_factor) <- c("B", "G", "R") # Levels are modified in alphabetical order of original levels

print(color_factor)

color_char_vector <- as.character(color_factor)

print(color_char_vector)

Output:



```
> print(color_factor)
[1] Red   Green Blue  Red   Green Blue  Red
Levels: Blue Green Red
> print(levels(color_factor))
[1] "Blue"  "Green" "Red"
> print(table(color_factor))
color_factor
 Blue Green   Red 
    2     2     3 
> levels(color_factor) <- c("B", "G", "R")  # Leve
betical order of original levels
> print(color_factor)
[1] R G B R G B R
Levels: B G R
> color_char_vector <- as.character(color_factor)
> print(color_char_vector)
[1] "R" "G" "B" "R" "G" "B" "R"
```

RESULT:

Thus , the vectors, lists, matrices, arrays, factors is implemented and output is verified successfully.