WEEK-3

TASK-1:

Programs on vectors and list

1. Write a R program to reverse the order of given vector

```r
original_vector <- c(1, 2, 3, 4, 5)

reversed_vector <- rev(original_vector)

cat("Original vector:", original_vector, "\n")
cat("Reversed vector:", reversed_vector, "\n")
```

OUTPUT:

```
> original_vector <- c(1, 2, 3, 4, 5)
> reversed_vector <- rev(original_vector)
> cat("Original vector:", original_vector, "\n")
Original vector: 1 2 3 4 5
> cat("Reversed vector:", reversed_vector, "\n")
Reversed vector: 5 4 3 2 1
>
```

2. . Write a R program to concatenate a vector with other

1.R - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

readxl Addins

Untitled1*

Source on Save

```r

vector1 <- c(1, 2, 3)
vector2 <- c(4, 5, 6)

concatenated_vector <- c(vector1, vector2)

cat("Concatenated vector:", concatenated_vector, "\n")

```

OUTPUT:

8:1 (Top Level)

Console Terminal Background Jobs

R 4.3.2 · ~/1.R/

```
> vector1 <- c(1, 2, 3)
> vector2 <- c(4, 5, 6)
> concatenated_vector <- c(vector1, vector2)
> cat("Concatenated vector:", concatenated_vector, "\n")
Concatenated vector: 1 2 3 4 5 6
>
```

3. Write a R program to count number of values in a range in a given vector.

```r
given_vector <- c(10, 15, 20, 25, 30, 35, 40, 45, 50)

lower_bound <- 20
upper_bound <- 40

values_in_range <- given_vector[given_vector >= lower_bound & given_vector <= upper_bound]
count <- length(values_in_range)

cat("Number of values in the range [", lower_bound, "-", upper_bound, "]:", count, "\n")
```

OUTPUT:

```
> given_vector <- c(10, 15, 20, 25, 30, 35, 40, 45, 50)
> lower_bound <- 20
> upper_bound <- 40
> values_in_range <- given_vector[given_vector >= lower_bound & given_vector <= upper_bound]
> count <- length(values_in_range)
> cat("Number of values in the range [", lower_bound, "-", upper_bound, "]:", count, "\n")
Number of values in the range [ 20 - 40 ]: 5
>
>
```

4. Write a R program to combines two given vectors
   By row
   By column

```r
vector1 <- c(1, 2, 3)
vector2 <- c(4, 5, 6)

combined_column <- cbind(vector1, vector2)

combined_row <- rbind(vector1, vector2)

cat("Combined by column:\n", combined_column, "\n")
cat("Combined by row:\n", combined_row, "\n")
```

OUTPUT:

```
> vector1 <- c(1, 2, 3)
> vector2 <- c(4, 5, 6)
> combined_column <- cbind(vector1, vector2)
> combined_row <- rbind(vector1, vector2)
> cat("Combined by column:\n", combined_column, "\n")
Combined by column:
 1 2 3 4 5 6
> cat("Combined by row:\n", combined_row, "\n")
Combined by row:
 1 4 2 5 3 6
>
>
```

5. Write a R program to test whether the value of the element of a given vector greater than 10 or not. Return TRUE or False

```
given_vector <- c(5, 12, 8, 15, 10)

greater_than_10 <- given_vector > 10

cat("For the given vector:", given_vector, "\n")
cat("Whether each element is greater than 10:", greater_than_10, "\n")
```

OUTPUT:

```
8:1    (Top Level)
Console   Terminal   Background Jobs
R 4.3.2 · ~/1.R/
> given_vector <- c(5, 12, 8, 15, 10)
> greater_than_10 <- given_vector > 10
> cat("For the given vector:", given_vector, "\n")
For the given vector: 5 12 8 15 10
> cat("Whether each element is greater than 10:", greater_than_10, "\n")
Whether each element is greater than 10: FALSE TRUE FALSE TRUE FALSE
>
>
```

6. Write a R program to create a list containing strings, numbers, vectors and a logical values.

```
File  Edit  Code  View  Plots  Session  Build  Debug  Profile  Tc
readxl
Untitled1*
Source on Save
1
2  my_list <- list(
3    string_element = "Hello, world!",
4    numeric_element = 123,
5    vector_element = c(1, 2, 3),
6    logical_element = TRUE
7  )
8
9  print(my_list)
10
```

OUTPUT:

```
Console   Terminal   Background Jobs
R 4.3.2 · ~/1.R/
+   logical_element = TRUE
+ )
> print(my_list)
$string_element
[1] "Hello, world!"

$numeric_element
[1] 123

$vector_element
[1] 1 2 3

$logical_element
[1] TRUE
```

7. Write a R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access the first and second element of the list.

```r

my_vector <- c(1, 2, 3, 4, 5)

my_matrix <- matrix(1:9, nrow = 3, ncol = 3)

inner_list <- list("a" = "apple", "b" = "banana", "c" = "cherry")

my_main_list <- list("vector" = my_vector,
                     "matrix" = my_matrix,
                     "inner_list" = inner_list)

first_element <- my_main_list[[1]]
second_element <- my_main_list[[2]]

cat("First element (vector):\n")
print(first_element)

cat("\nSecond element (matrix):\n")
print(second_element)
```

OUTPUT:

```
> second_element <- my_main_list[[2]]
> cat("First element (vector):\n")
First element (vector):
> print(first_element)
[1] 1 2 3 4 5
> cat("\nSecond element (matrix):\n")

Second element (matrix):
> print(second_element)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
```

8. Write a R program to create a list containing a vector, a matrix and a list and add element at the end of the list.

File Edit Code View Plots Session Build Debug Profile Tools Help

readxl Addins

Untitled1*

Source on Save

```r

my_vector <- c(1, 2, 3, 4, 5)

my_matrix <- matrix(1:9, nrow = 3, ncol = 3)

inner_list <- list("a" = "apple", "b" = "banana", "c" = "cherry")

my_main_list <- list("vector" = my_vector,
                     "matrix" = my_matrix,
                     "inner_list" = inner_list)

additional_element <- "extra"
names(additional_element) <- "additional_element"
my_main_list[["additional_element"]] <- additional_element

print(my_main_list)

```

OUTPUT:

```
$vector
[1] 1 2 3 4 5

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

$inner_list
$inner_list$a
[1] "apple"

$inner_list$b
[1] "banana"

$inner_list$c
[1] "cherry"


$additional_element
additional_element
           "extra"
```

9. Write a R program to select second element of a given nested list.

File Edit Code View Plots Session Build Debug Profile Tools Help

readxl

Untitled1*

Source on Save

```r

nested_list <- list(
  element1 = c(1, 2, 3),
  element2 = matrix(1:4, nrow = 2),
  element3 = list("a", "b", "c")
)

second_element <- nested_list[[2]]

print(second_element)
```

OUTPUT:

Console Terminal Background Jobs

R 4.3.2 · ~/1.R/

```
> second_element <- nested_list[[2]]
> print(second_element)
     [,1] [,2]
[1,]    1    3
[2,]    2    4
>
```

Task:2

Programs on Arrays and Matrix

1. Write a R program to create a matrix from a list of given vectors.

```r
list_of_vectors <- list(
  vector1 = c(1, 2, 3),
  vector2 = c(4, 5, 6),
  vector3 = c(7, 8, 9)
)
matrix_from_list <- do.call(cbind, list_of_vectors)
print(matrix_from_list)
```

OUTPUT:

```
     vector1 vector2 vector3
[1,]       1       4       7
[2,]       2       5       8
[3,]       3       6       9
>
```

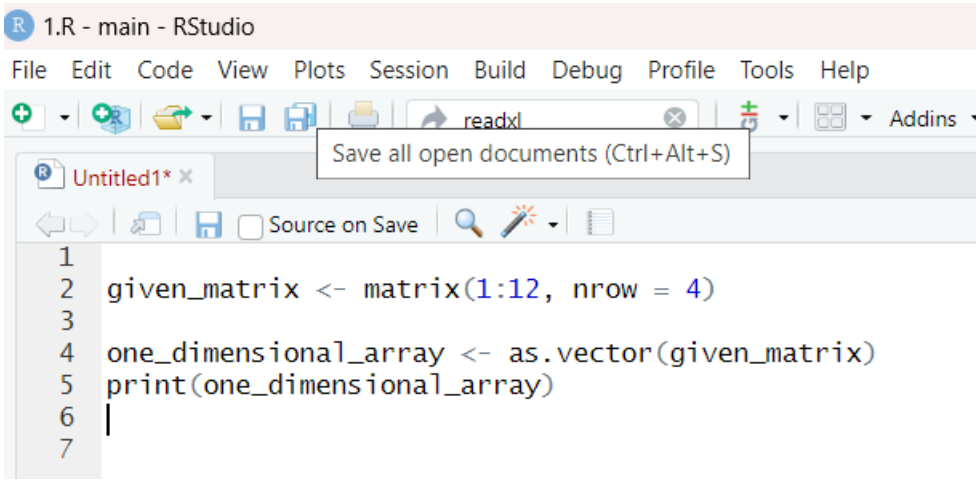
2. Write a R program to extract the submatrix whose rows have column value > 7 from a given matrix.

```r
given_matrix <- matrix(1:12, nrow = 4, byrow = TRUE)
colnames(given_matrix) <- c("A", "B", "C")
rownames(given_matrix) <- c("Row1", "Row2", "Row3", "Row4")

submatrix <- given_matrix[given_matrix[, "B"] > 7, ]

print("Given Matrix:")
print(given_matrix)

print("Submatrix whose rows have column value > 7:")
print(submatrix)
```

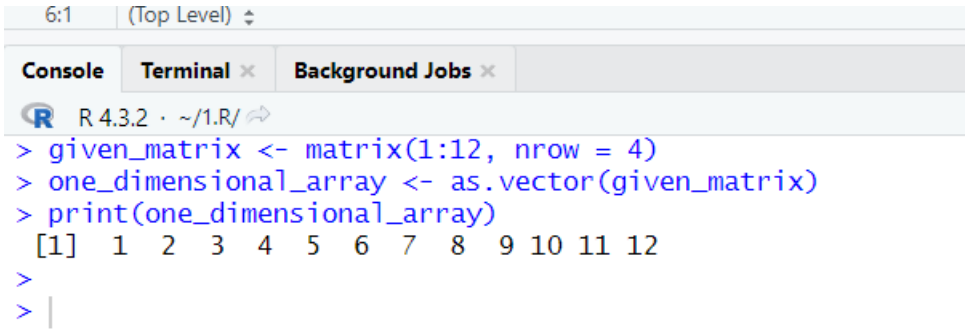
OUTPUT:

```
> print("Given Matrix:")
[1] "Given Matrix:"
> print(given_matrix)
      A  B  C
Row1  1  2  3
Row2  4  5  6
Row3  7  8  9
Row4 10 11 12
> print("Submatrix whose rows have column value > 7:")
[1] "Submatrix whose rows have column value > 7:"
> print(submatrix)
      A  B  C
Row3  7  8  9
Row4 10 11 12
>
```

3. Write a R program to convert a matrix to a 1 dimensional array.



OUTPUT:



4. Write a R program to find row and column index of maximum and minimum value in a given matrix.

```r
given_matrix <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2)
max_index <- which(given_matrix == max(given_matrix), arr.ind = TRUE)

min_index <- which(given_matrix == min(given_matrix), arr.ind = TRUE)
cat("Row and column index of maximum value:", max_index[1], ",", max_index[2], "\n")
cat("Row and column index of minimum value:", min_index[1], ",", min_index[2], "\n")
```

OUTPUT:

```
> given_matrix <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2)
> max_index <- which(given_matrix == max(given_matrix), arr.ind = TRUE)
> min_index <- which(given_matrix == min(given_matrix), arr.ind = TRUE)
> cat("Row and column index of maximum value:", max_index[1], ",", max_index[2], "\n")
Row and column index of maximum value: 2 , 3
> cat("Row and column index of minimum value:", min_index[1], ",", min_index[2], "\n")
Row and column index of minimum value: 1 , 1
>
```

5. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors.

```r
vector1 <- 1:9
vector2 <- 10:18

array_of_matrices <- array(c(vector1, vector2), dim = c(3, 3, 2))

print(array_of_matrices)
```

OUTPUT:

```
R 4.3.2 · ~/1.R/
> print(array_of_matrices)
, , 1

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

, , 2

     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18
```

6. Write a R program to create an 3 dimensional array of 24 elements using the dim() function.

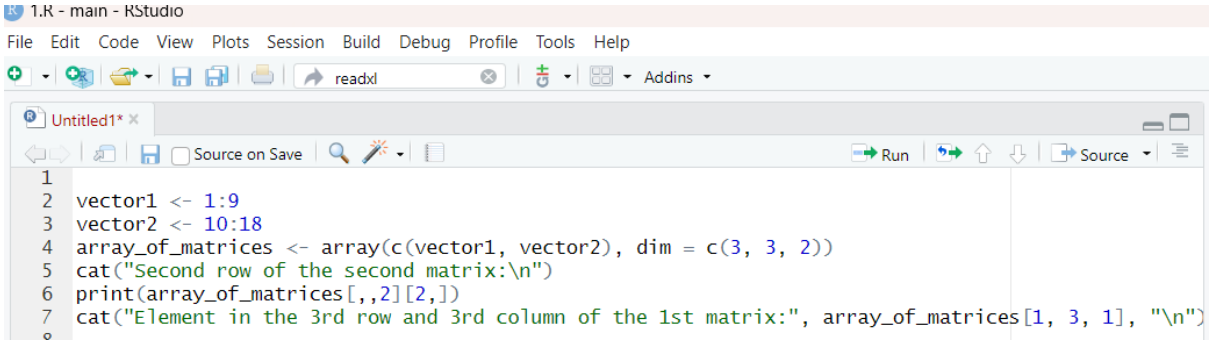
```
1.R - main - RStudio
File  Edit  Code  View  Plots  Session  Build  Debug  Profile  Tools  Help
readxl                                                    Addins
Untitled1*
Source on Save
1
2  dimensions <- c(2, 3, 4)
3  array_3d <- array(1:24, dim = dimensions)
4  print(array_3d)
5
```

OUTPUT:

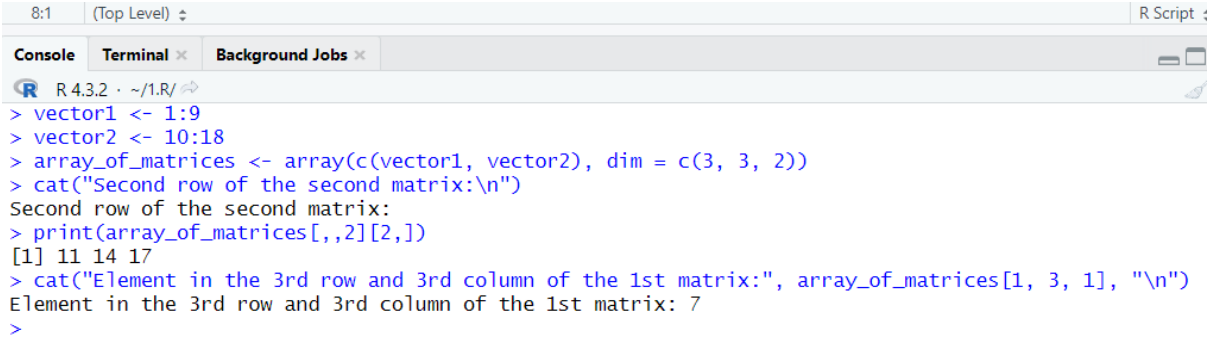
```
Console   Terminal   Background Jobs
R 4.3.2 · ~/1.R/
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12
```

```
, , 3

     [,1] [,2] [,3]
[1,]   13   15   17
[2,]   14   16   18

, , 4

     [,1] [,2] [,3]
[1,]   19   21   23
[2,]   20   22   24
```

7. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.



OUTPUT:



8. Write a R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then first row of the third array.

1.R - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

readxl Addins

Untitled1*

Source on Save

```
array1 <- matrix(1:9, nrow = 3, byrow = TRUE)
array2 <- matrix(10:18, nrow = 3, byrow = TRUE)
array3 <- matrix(19:27, nrow = 3, byrow = TRUE)
combined_array <- rbind(array1[1, ], array2[1, ], array3[1, ])
print(combined_array)
```

OUTPUT:

Console Terminal Background Jobs

R 4.3.2 · ~/1.R/

```
> array1 <- matrix(1:9, nrow = 3, byrow = TRUE)
> array2 <- matrix(10:18, nrow = 3, byrow = TRUE)
> array3 <- matrix(19:27, nrow = 3, byrow = TRUE)
> combined_array <- rbind(array1[1, ], array2[1, ], array3[1, ])
> print(combined_array)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]   10   11   12
[3,]   19   20   21
>
```

Task3:

Programs on Data Frames

1. Write a R program to create a data frame from four given vectors.

```r
vector1 <- c(1, 2, 3, 4)
vector2 <- c("A", "B", "C", "D")
vector3 <- c(TRUE, FALSE, TRUE, FALSE)
vector4 <- c(10.5, 20.5, 30.5, 40.5)
my_dataframe <- data.frame(
  Column1 = vector1,
  Column2 = vector2,
  Column3 = vector3,
  Column4 = vector4
)
print(my_dataframe)
```

OUTPUT:

```
+ )
> print(my_dataframe)
  Column1 Column2 Column3 Column4
1       1       A    TRUE    10.5
2       2       B   FALSE    20.5
3       3       C    TRUE    30.5
4       4       D   FALSE    40.5
>
```

2. Write a R program to get the statistical summary and nature of the data of a given data frame.

```r
my_dataframe <- data.frame(
  ID = c(1, 2, 3, 4),
  Name = c("John", "Alice", "Bob", "Emily"),
  Age = c(25, 30, 28, 35),
  Score = c(85, 92, 88, 79)
)
cat("Statistical Summary:\n")
print(summary(my_dataframe))
cat("\nStructure of the Data Frame:\n")
print(str(my_dataframe))
```

OUTPUT:

```
Console   Terminal   Background Jobs
R 4.3.2 · ~/1.R/
Statistical Summary:
> print(summary(my_dataframe))
       ID           Name                Age            Score
 Min.   :1.00   Length:4           Min.   :25.00   Min.   :79.0
 1st Qu.:1.75   Class :character   1st Qu.:27.25   1st Qu.:83.5
 Median :2.50   Mode  :character   Median :29.00   Median :86.5
 Mean   :2.50                      Mean   :29.50   Mean   :86.0
 3rd Qu.:3.25                      3rd Qu.:31.25   3rd Qu.:89.0
 Max.   :4.00                      Max.   :35.00   Max.   :92.0
> cat("\nStructure of the Data Frame:\n")

Structure of the Data Frame:
> print(str(my_dataframe))
'data.frame':   4 obs. of  4 variables:
 $ ID   : num  1 2 3 4
 $ Name : chr  "John" "Alice" "Bob" "Emily"
 $ Age  : num  25 30 28 35
 $ Score: num  85 92 88 79
NULL
>
```

3. Write a R program to extract specific column from a data frame using column name

```
1.R - main - RStudio
File  Edit  Code  View  Plots  Session  Build  Debug  Profile  Tools  Help
readxl                                                    Addins
Untitled1*
Source on Save
1
2  my_dataframe <- data.frame(
3    ID = c(1, 2, 3, 4),
4    Name = c("John", "Alice", "Bob", "Emily"),
5    Age = c(25, 30, 28, 35),
6    Score = c(85, 92, 88, 79)
7  )
8  name_column <- my_dataframe$Name
9  print(name_column)
```

OUTPUT:

```
Console   Terminal ×   Background Jobs ×
R 4.3.2 · ~/1.R/
> my_dataframe <- data.frame(
+   ID = c(1, 2, 3, 4),
+   Name = c("John", "Alice", "Bob", "Emily"),
+   Age = c(25, 30, 28, 35),
+   Score = c(85, 92, 88, 79)
+ )
> name_column <- my_dataframe$Name
> print(name_column)
[1] "John"  "Alice" "Bob"   "Emily"
>
>
>
```

4. Write a R program to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame.

```
1.R - main - RStudio
File  Edit  Code  View  Plots  Session  Build  Debug  Profile  Tools  Help
readxl                                                          Addins
Untitled1*
Source on Save

my_dataframe <- data.frame(
  ID = c(1, 2, 3, 4, 5),
  Name = c("John", "Alice", "Bob", "Emily", "David"),
  Age = c(25, 30, 28, 35, 40),
  Score = c(85, 92, 88, 79, 95)
)
extracted_data <- my_dataframe[c(3, 5), c(1, 3)]
print(extracted_data)
```

OUTPUT:

```
Console   Terminal ×   Background Jobs ×
R 4.3.2 · ~/1.R/
> my_dataframe <- data.frame(
+   ID = c(1, 2, 3, 4, 5),
+   Name = c("John", "Alice", "Bob", "Emily", "David"),
+   Age = c(25, 30, 28, 35, 40),
+   Score = c(85, 92, 88, 79, 95)
+ )
> extracted_data <- my_dataframe[c(3, 5), c(1, 3)]
> print(extracted_data)
  ID Age
3  3  28
5  5  40
```

5. Write a R program to add new row(s) and new column to an existing data frame.

1.R - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

readxl Addins

Untitled1*

Source on Save

```r

existing_dataframe <- data.frame(
  ID = c(1, 2, 3),
  Name = c("John", "Alice", "Bob"),
  Age = c(25, 30, 28)
)
new_row <- c(4, "Emily", 35)
existing_dataframe <- rbind(existing_dataframe, new_row)
cat("Data frame with new row:\n")
print(existing_dataframe)
existing_dataframe$Score <- c(85, 92, 88, 79)
cat("\nData frame with new column:\n")
print(existing_dataframe)

```

OUTPUT:

14:1 (Top Level)

Console Terminal Background Jobs

R 4.3.2 · ~/1.R/

```
  ID  Name Age
1  1  John  25
2  2 Alice  30
3  3   Bob  28
4  4 Emily  35
> existing_dataframe$Score <- c(85, 92, 88, 79)
> cat("\nData frame with new column:\n")

Data frame with new column:
> print(existing_dataframe)
  ID  Name Age Score
1  1  John  25    85
2  2 Alice  30    92
3  3   Bob  28    88
4  4 Emily  35    79
>
```