**Solution Provider Name: Dipali Bharat Jadhav**

**College: V.P.Institute of Management Studies & Research Sangli  
R Version: 4.0.3**

**R Studio Version: 1.1.463 Operation System: Windows 10 OS Exercise No. 3**

* rm(list = ls())

rm() function is used for remove data in R environment.

# Matrix:

Matrix is 2 dimensional data structure in R programming. Matrix is a similar collection of data types for ex. Numeric, character, float, double etc.

**Vector**: Vector is a basic data type structure in R programming. For creating a vector in R programming, R provide c() function.

**Syntax**: vector\_name <- c(“value1”, “value2”,……,“value\_n”) Where, vector\_name is used to define name of the vector.

c() is used to create a function for passing values or arguments. Value may be integer, string, float values etc.

Example: demo <- c (10,20,30,40)

# Que 1 Create 2 matrix of 3\*3 from two vectors Command: Matrix 1

> V1 <- c(10,11,12)

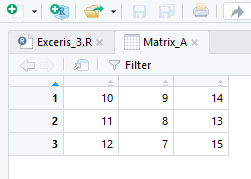
>V2 <- c(9,8,7,14,13,15)

>Matrix\_A <- data.frame(V1,V2)

>Matrix\_A=array(c(V1,V2),dim = c(3,3))

>View(Matrix\_A)

# Output:



**Create .csv file:** write.csv(Matrix\_A,"NewMatrix\_A.csv") NewMatrix\_A <- read.csv("NewMatrix\_A.csv") print(NewMatrix\_A)

# Command: Matrix 1

> A1 <- c(9,8,7)

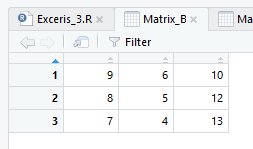
> A2 <- c(6,5,4,10,12,13)

* Matrix\_B <- data.frame(A1,A2)

>Matrix\_B=array(c(A1,A2),dim = c(3,3))

>View(Matrix\_B)

# Output:



**Create .csv file:** write.csv(Matrix\_B,"NewMatrix\_B.csv") NewMatrix\_B <- read.csv("NewMatrix\_B.csv") print(NewMatrix\_B)

# Que 2 Display Addition of 2 matrix

Using plus (+) sign to calculate addition of two matrix

# Command:

* Matrix\_A + Matrix\_B

# Output:

[,1] [,2] [,3]

[1,] 19 15 24

[2,] 19 13 25

[3,] 19 11 28

# Que 3 Display Subtraction of 2 matrix

Using minus (-) sign to calculate Subtraction of two matrix

# Command:

* Matrix\_A - Matrix\_B

# Output:

[,1] [,2] [,3]

|  |  |  |  |
| --- | --- | --- | --- |
| [1,] | 1 | 3 | 4 |
| [2,] | 3 | 3 | 1 |
| [3,] | 5 | 3 | 2 |

# Que 4 Display Multiplication of 2 matrix

Using **%\*%** this operator to calculate Multiplication of two matrix. This operator satisfied their condition that the number of columns in the first matrix is equal to the number of rows in second

* Matrix\_A %\*% Matrix\_B

# Output:

[,1] [,2] [,3]

[1,] 260 161 390

[2,] 254 158 375

[3,] 269 167 399

# Que 5 Display Transpose of Matrix

Transpose of matrix is use to convert rows into columns and columns into rows into matrix.

Use **t ()** function to convert transpose matrix.

**Syntax**: t(x)

Where x is data frame or matrix name

# Matrix 1

* t(Matrix\_A)

# Output:

[,1] [,2] [,3]

|  |  |  |  |
| --- | --- | --- | --- |
| [1,] | 9 | 8 | 7 |
| [2,] | 6 | 5 | 4 |
| [3,] | 10 | 12 | 13 |

# Matrix 2

* t(Matrix\_B)

# Output:

[,1] [,2] [,3]

|  |  |  |  |
| --- | --- | --- | --- |
| [1,] | 10 | 11 | 12 |
| [2,] | 9 | 8 | 7 |
| [3,] | 14 | 13 | 15 |

# Que 6 Display Inverse of Matrix

**Inverse**: Changing in the opposite way in relation to something else. Function **solve()** is used to display inverse of matrix

# Matrix 1

>Inverse\_A <- solve(Matrix\_A)

>print(Inverse\_A)

# Output:

|  |  |  |  |
| --- | --- | --- | --- |
| [,1] | [,2] | [,3] | |
| [1,] -0.5087719 | | 0.6491228 | -0.0877193 |
| [2,] 0.1578947 | | 0.3157895 | -0.4210526 |
| [3,] 0.3333333 | | -0.6666667 | 0.3333333 |
| **Matrix 2 Output:** | |  |  |

* Inverse\_B <- solve(Matrix\_B) print(Inverse\_B)

# Output:

|  |  |  |  |
| --- | --- | --- | --- |
| [,1] | [,2] | [,3] | |
| [1,] 5.666667 | | -12.66667 | 7.333333 |
| [2,] -6.666667 | | 15.66667 | -9.333333 |
| [3,] -1.000000 | | 2.00000 | -1.000000 |

**Que 7 Display even number from Matrix**

Using for loop and if statement it display even number.

**For loop:** A for loop is used to iterate over a [vector in R programming](https://www.datamentor.io/r-programming/vector).

**Syntax:** for (variable in vector) {

statements

}

Where, variable is value during in the loop.

Vector is sequence.

**If Statement**: It is used for test expression which is true or false.

**Syntax**: if(Condition/Expression )

{

statements

}

# Matrix 1: Command:

* for (val in Matrix\_A)

{

+ if(val %% 2 == 0)

{

print(val)

}

+ }

# Output:

[1] 8

[1] 6

[1] 4 [1] 10 [1] 12

# Matrix 2: Command:

* for (val in Matrix\_B) {

+ if(val %% 2 == 0){ print(val)

}

+ }

# Output:

[1] 10

[1] 12

[1] 8 [1] 14

# Que 8 Display sum of matrix element.

Display sum of matrix element using for loop and sum variable. Print() function is display output of a program

# Matrix 1 Command:

* sum=0
* for (val in Matrix\_A) {

+ sum=sum+val

+ }

* print(sum)

# Output:

[1] 99

# Matrix 2 Command:

* sum=0
* for (val in Matrix\_B) {

+ sum=sum+val

+ }

* print(sum)

# Output:

[1] 74