

**Name:- Mithun G**

**USN:- 19BTRCR006**

## **LAB PROGRAM 2**

### **1. Explore assignment operator**

In [2]:

```
a <- 5  
a
```

5

In [3]:

```
b = 5  
b
```

5

In [6]:

```
3 -> c  
c
```

3

In [7]:

```
d <<- 4  
d
```

4

In [9]:

```
4 ->> e  
e
```

4

### **2.Create vectors using c(), seq(), rep(), colon operator.**

In [17]:

```
vec <- c(1,2,3)
vec
```

1 · 2 · 3

In [19]:

```
vec1 <- seq(1,10,2)
vec1
```

1 · 3 · 5 · 7 · 9

In [21]:

```
vec2 <- rep(1,5)
vec2
```

1 · 1 · 1 · 1 · 1

In [27]:

```
vec3 <- c(1:10)
vec3
```

1 · 2 · 3 · 4 · 5 · 6 · 7 · 8 · 9 · 10

### 3. Create different matrices using matrix() operator and explore its rows, columns, and diagonals.

In [47]:

```
mat_r <- matrix(data=c(1:9),nrow=3)
mat_r
```

A matrix:

3 × 3 of

type int

1 4 7

2 5 8

3 6 9

In [48]:

```
mat_r[1,1:3]      # printing the first row elements
```

1 · 4 · 7

In [53]:

```
mat_c <- matrix(data=c(1:9),nrow=3, byrow=T)
mat_c
```

A matrix:

3 × 3 of

type int

```
1 2 3
4 5 6
7 8 9
```

In [50]:

```
mat_c[1:3,1]      # printing the first column elements
```

```
1 4 7
```

In [54]:

```
diag(mat_c)        # printing the diagonal elements
```

```
1 5 9
```

#### 4. Perform different basic operation of matrices on above created matrices.

In [59]:

```
# previously created matrices
print(mat_r)
print(mat_c)
```

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

In [58]:

```
# matirx addition  
mat_r + mat_c
```

A matrix: 3 × 3  
of type int

```
2   6  10  
6   10 14  
10  14 18
```

In [60]:

```
# matirx subtraction  
mat_r - mat_c
```

A matrix: 3  
× 3 of type  
int

```
0   2  4  
-2  0  2  
-4 -2  0
```

In [61]:

```
# matirx hadamard multiplication  
mat_r * mat_c
```

A matrix: 3 × 3  
of type int

```
1   8  21  
8   25 48  
21  48  81
```

In [62]:

```
# matirx true multiplication  
mat_r %**% mat_c
```

A matrix: 3 × 3 of  
type dbl

```
66   78   90  
78   93  108  
90  108  126
```

In [64]:

```
# scalar multiplication
(1/5) * mat_r
```

A matrix: 3 × 3

of type dbl

```
0.2  0.8  1.4
0.4  1.0  1.6
0.6  1.2  1.8
```

## 5. Create single and multidimensional arrays with array() command.

In [65]:

```
# single dimensional array
arr <- array(1:10)
arr
```

```
1· 2· 3· 4· 5· 6· 7· 8· 9· 10
```

In [70]:

```
# multi dimensional array
arr_md = array(data=1:3, dim= c(2,4))
arr_md
```

A matrix: 2 ×

4 of type int

```
1  3  2  1
2  1  3  2
```

## 6. Explore length(), dim(), ncol(), nrow() operators on above matrices and arrays.

In [82]:

```
print(mat_r)  # previously created matrix
print(arr_md) # previously created array
```

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

In [83]:

```
# Length
print(length(mat_r))
print(length(arr_md))
```

```
[1] 9
[1] 8
```

In [84]:

```
# dimensions
print(dim(mat_r))
print(dim(arr_md))
```

```
[1] 3 3
[1] 2 4
```

In [85]:

```
# ncol() and nrow()
print(ncol(mat_r))
print(nrow(arr_md))
```

```
[1] 3
[1] 2
```

## 7. Explore commands for Selecting and extracting elements from above matrices and arrays.

In [86]:

```
print(mat_r)  # previously created matrix
print(arr_md) # previously created array
```

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

In [87]:

```
# selecting element 5 from matrix
mat_r[2,2]
```

```
5
```

In [89]:

```
# selecting element 3 from 1st row from array
arr_md[1,2]
```

```
3
```

In [91]:

```
# extracting 1st row of the matrix
mat_r[1,1:3]
```

1 · 4 · 7

In [98]:

```
# extracting 1st row of the array
arr_md[1,]
```

1 · 3 · 2 · 1

## 8. Explore logical operators from R programming language.

In [117]:

```
vec1 <- c(T,F,T,F)
vec2 <- c(F,F,T,T)
print(vec1)
print(vec2)
```

```
[1] TRUE FALSE TRUE FALSE
[1] FALSE FALSE TRUE TRUE
```

In [118]:

```
# & operation
vec1 & vec2
```

FALSE · FALSE · TRUE · FALSE

In [119]:

```
# | operation
vec1 | vec2
```

TRUE · FALSE · TRUE · TRUE

In [115]:

```
# && operation
vec1 && vec2
```

FALSE

In [120]:

```
# || operation
vec1 || vec2
```

TRUE

In [121]:

```
# not operation
!vec1
```

FALSE · TRUE · FALSE · TRUE

## 9. Remove elements from selected positions from a considered matrix.

In [132]:

```
print(mat_c) # old matrix created
```

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

In [135]:

```
# removing element 2
mat_c[-4] # -4 because 2 is at 4th position on matrix
```

1 · 4 · 7 · 5 · 8 · 3 · 6 · 9

In [141]:

```
# removing first 2 elements
mat_c[-1:-2]
```

7 · 2 · 5 · 8 · 3 · 6 · 9