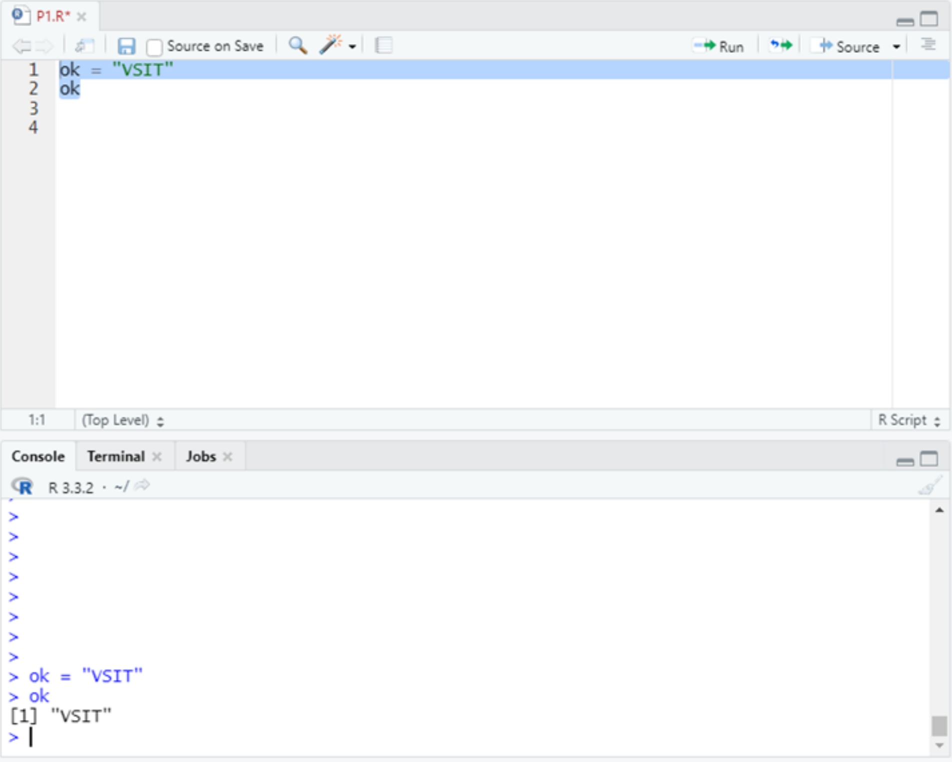
COST Practical's

Practical 1 -

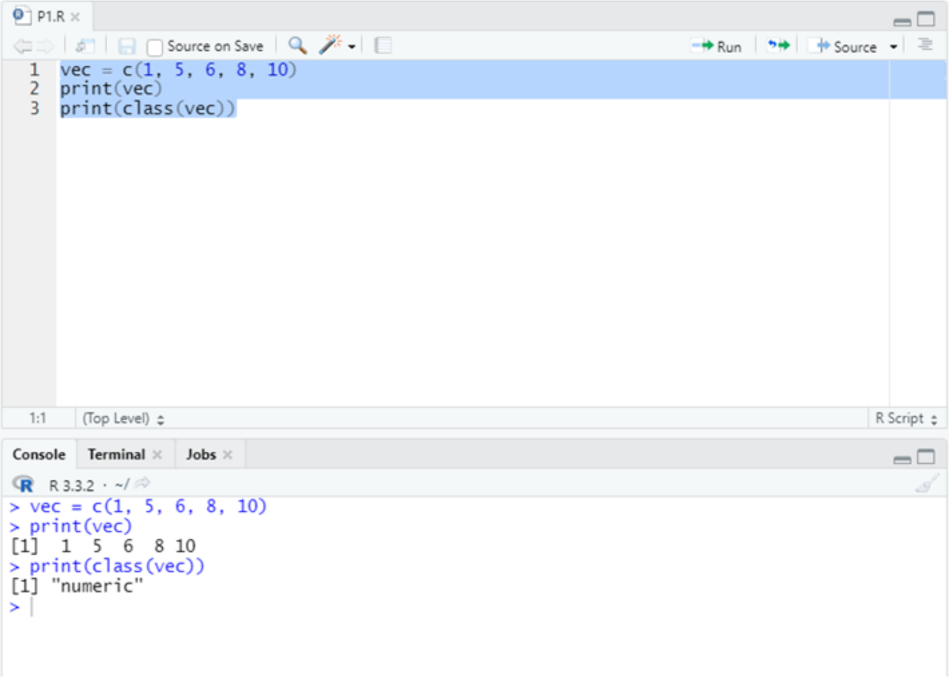
1. Write a code to display "Welcome to VSIT" message along with comment on screen and paste the screenshot in the box given below.



```
ok = "VSIT"
ok
```

```
>
>
>
>
>
>
>
>
> ok = "VSIT"
> ok
[1] "VSIT"
>
```

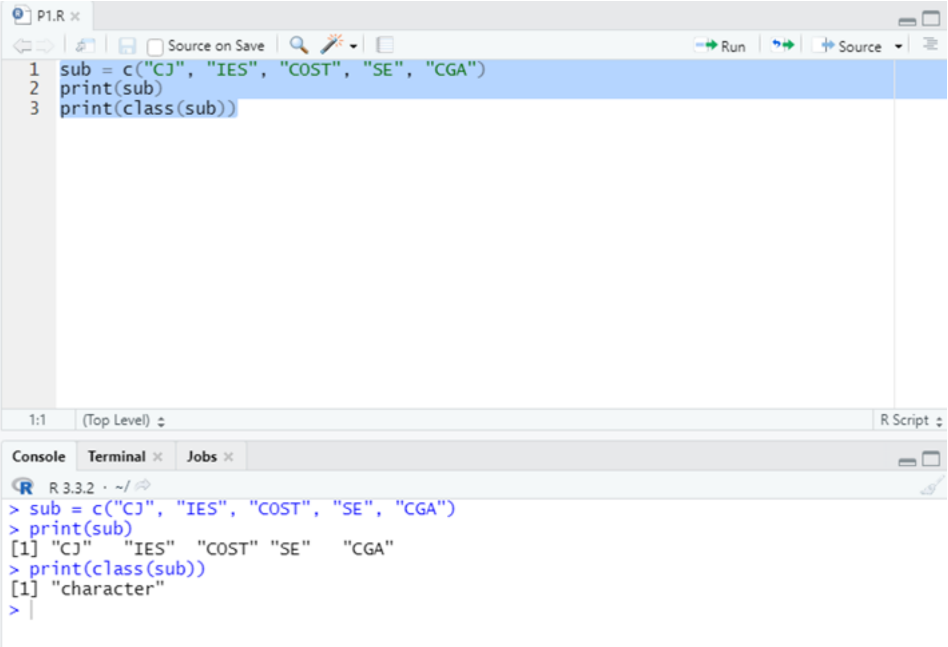
2. Write a code to display vector (1, 5, 6, 8, 10) on screen. Also display vector class. paste the screenshot in the box given below.



```
vec = c(1, 5, 6, 8, 10)
print(vec)
print(class(vec))
```

```
> vec = c(1, 5, 6, 8, 10)
> print(vec)
[1]  1  5  6  8 10
> print(class(vec))
[1] "numeric"
>
```

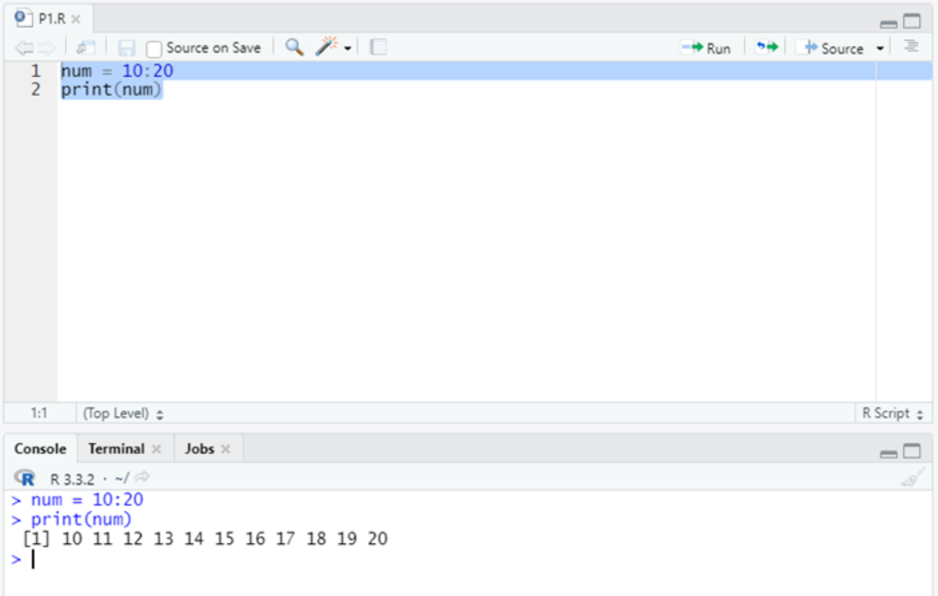
3. Write a code to display vector of all the subjects in SYIT on screen. Also display vector class. paste the screenshot in the box given below.



```
sub = c("CJ", "IES", "COST", "SE", "CGA")
print(sub)
print(class(sub))
```

```
> sub = c("CJ", "IES", "COST", "SE", "CGA")
> print(sub)
[1] "CJ"    "IES"   "COST" "SE"    "CGA"
> print(class(sub))
[1] "character"
>
```

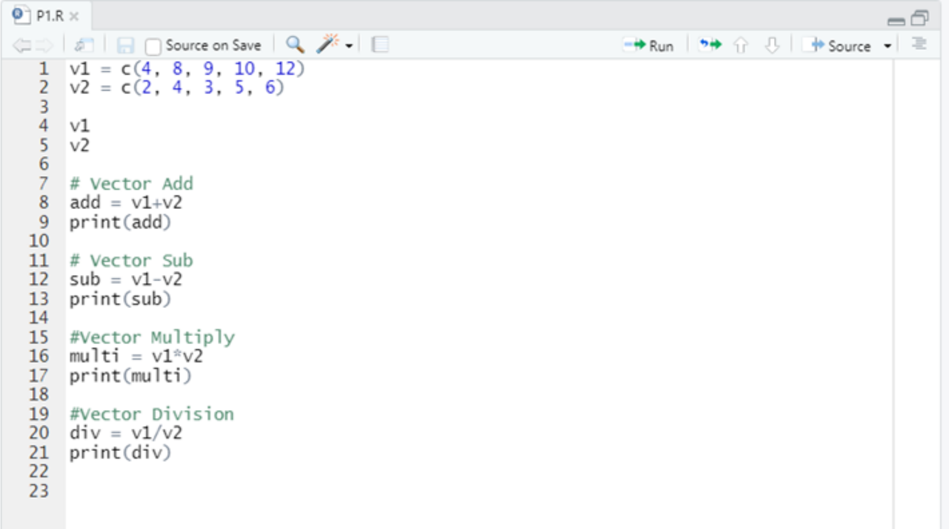
4. Write a code to display vector of all the elements between 10 to 20 on screen. paste the screenshot in the box given below.



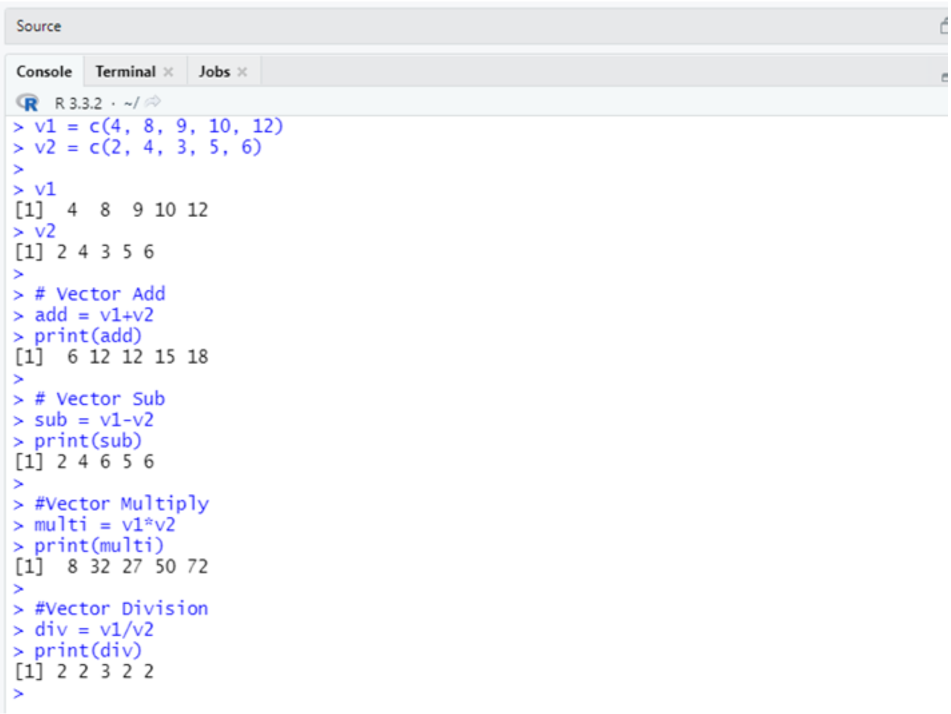
```
num = 10:20
print(num)
```

```
> num = 10:20
> print(num)
 [1] 10 11 12 13 14 15 16 17 18 19 20
> 
```

5. Write a code to add, subtract, multiply and divide two vectors (4, 8, 9, 10, 12) and (2, 4, 3, 5, 6). Paste the screenshot in the box given below.



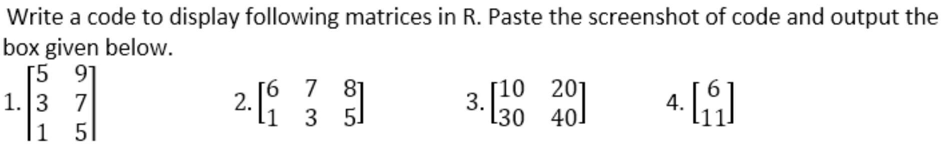
```
v1 = c(4, 8, 9, 10, 12)
v2 = c(2, 4, 3, 5, 6)

v1
v2

# Vector Add
add = v1+v2
print(add)

# Vector Sub
sub = v1-v2
print(sub)

#Vector Multiply
multi = v1*v2
print(multi)

#Vector Division
div = v1/v2
print(div)

```



```
> v1 = c(4, 8, 9, 10, 12)
> v2 = c(2, 4, 3, 5, 6)
>
> v1
[1]  4  8  9 10 12
> v2
[1] 2 4 3 5 6
>
> # Vector Add
> add = v1+v2
> print(add)
[1]  6 12 12 15 18
>
> # Vector Sub
> sub = v1-v2
> print(sub)
[1] 2 4 6 5 6
>
> #Vector Multiply
> multi = v1*v2
> print(multi)
[1]  8 32 27 50 72
>
> #Vector Division
> div = v1/v2
> print(div)
[1] 2 2 3 2 2
>
```

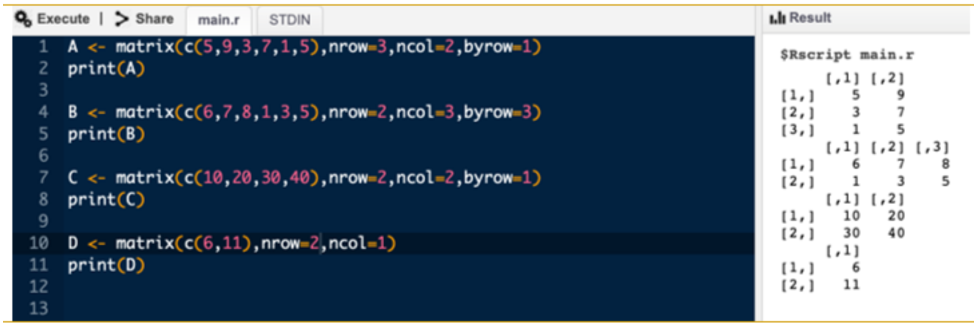
Practical 2 -

1. Write a code to display following matrices in R. Paste the screenshot of code and output the box given below.



Write a code to display following matrices in R. Paste the screenshot of code and output the box given below.

1. $\begin{bmatrix} 5 & 9 \\ 3 & 7 \\ 1 & 5 \end{bmatrix}$ 2. $\begin{bmatrix} 6 & 7 & 8 \\ 1 & 3 & 5 \end{bmatrix}$ 3. $\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$ 4. $\begin{bmatrix} 6 \\ 11 \end{bmatrix}$

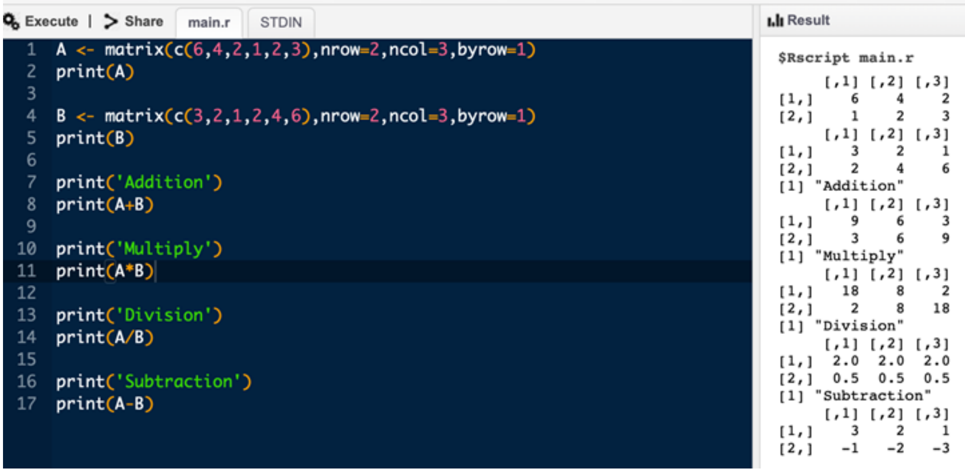


```
A <- matrix(c(5,9,3,7,1,5),nrow=3,ncol=2,byrow=1)
print(A)

B <- matrix(c(6,7,8,1,3,5),nrow=2,ncol=3,byrow=3)
print(B)

C <- matrix(c(10,20,30,40),nrow=2,ncol=2,byrow=1)
print(C)

D <- matrix(c(6,11),nrow=2,ncol=1)
print(D)
```

```
$Rscript main.r
     [,1] [,2]
[1,]    5    9
[2,]    3    7
[3,]    1    5
     [,1] [,2] [,3]
[1,]    6    7    8
[2,]    1    3    5
     [,1] [,2]
[1,]   10   20
[2,]   30   40
     [,1]
[1,]    6
[2,]   11
```

2. Write a code to add, subtract, divide, and multiply the two matrices given below. Paste the screenshot of code and output in the box given below.

$$M1 = \begin{bmatrix} 6 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix} \qquad M2 = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 6 \end{bmatrix}$$

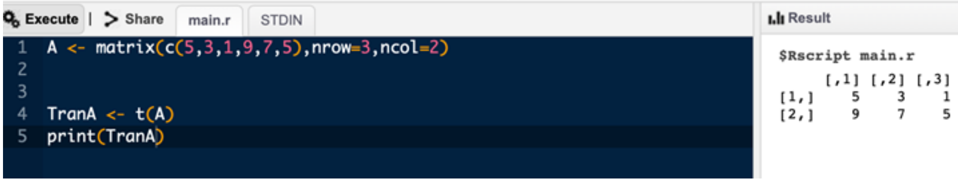


```r
A <- matrix(c(6,4,2,1,2,3),nrow=2,ncol=3,byrow=1)
print(A)

B <- matrix(c(3,2,1,2,4,6),nrow=2,ncol=3,byrow=1)
print(B)

print('Addition')
print(A+B)

print('Multiply')
print(A*B)

print('Division')
print(A/B)

print('Subtraction')
print(A-B)
```

```
$Rscript main.r
     [,1] [,2] [,3]
[1,]    6    4    2
[2,]    1    2    3
     [,1] [,2] [,3]
[1,]    3    2    1
[2,]    2    4    6
[1] "Addition"
     [,1] [,2] [,3]
[1,]    9    6    3
[2,]    3    6    9
[1] "Multiply"
     [,1] [,2] [,3]
[1,]   18    8    2
[2,]    2    8   18
[1] "Division"
     [,1] [,2] [,3]
[1,]  2.0  2.0  2.0
[2,]  0.5  0.5  0.5
[1] "Subtraction"
     [,1] [,2] [,3]
[1,]    3    2    1
[2,]   -1   -2   -3
```

3. What is transpose of a matrix? Write a code to find transpose of a matrix A. Paste the screenshot of code and output in the box given below.

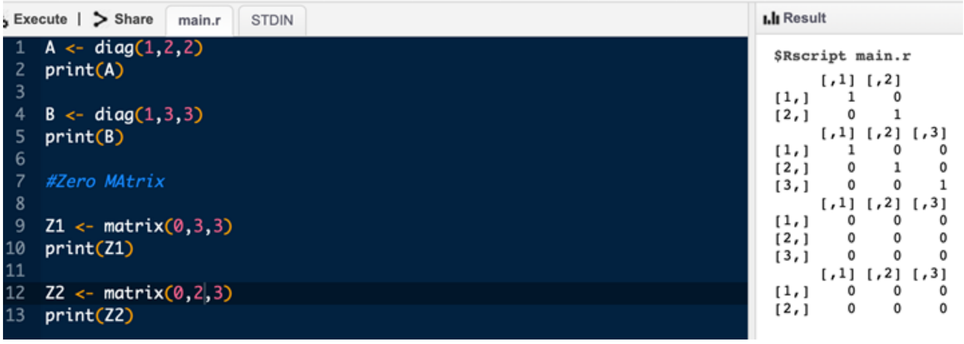
$$A = \begin{bmatrix} 5 & 9 \\ 3 & 7 \\ 1 & 5 \end{bmatrix}$$



```r
A <- matrix(c(5,3,1,9,7,5),nrow=3,ncol=2)


TranA <- t(A)
print(TranA)
```

```
$Rscript main.r
     [,1] [,2] [,3]
[1,]    5    3    1
[2,]    9    7    5
```

4. Write a code to display following **unit matrix** of order **2x2** and **3x2.** And **zero matrix** of order **3x3** and **2x3**. Paste the screenshot in the box given below.

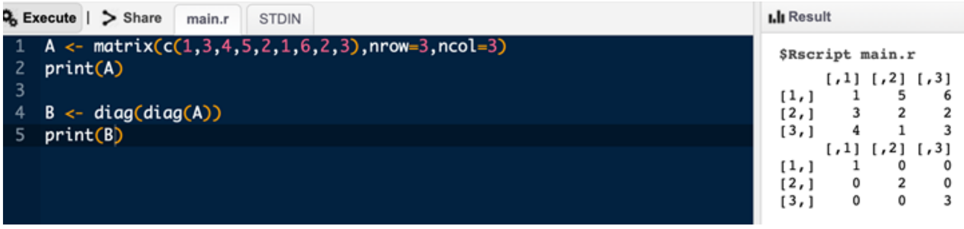


```r
A <- diag(1,2,2)
print(A)

B <- diag(1,3,3)
print(B)

#Zero MAtrix

Z1 <- matrix(0,3,3)
print(Z1)

Z2 <- matrix(0,2,3)
print(Z2)
```

```
$Rscript main.r
     [,1] [,2]
[1,]    1    0
[2,]    0    1
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
```

5. Write a code to display diagonal matrix of the following matrix A. Paste the screenshot in the box given below.

$$\text{A=} \begin{bmatrix} 1 & 5 & 6 \\ 3 & 2 & 2 \\ 4 & 1 & 3 \end{bmatrix}$$

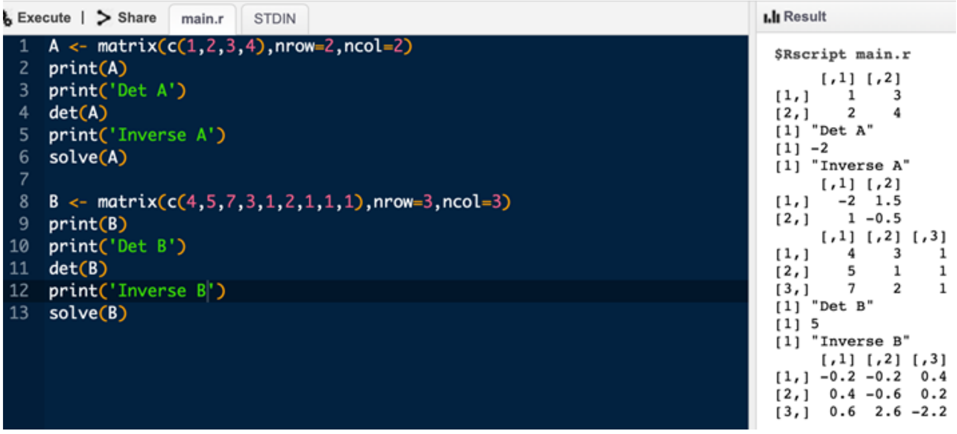


```r
A <- matrix(c(1,3,4,5,2,1,6,2,3),nrow=3,ncol=3)
print(A)

B <- diag(diag(A))
print(B)
```

```
$Rscript main.r
     [,1] [,2] [,3]
[1,]    1    5    6
[2,]    3    2    2
[3,]    4    1    3
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
```

6. Write a code to find a determinant and Inverse of a matrix A and B. Paste the screenshot in the box given below.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B= \begin{bmatrix} 4 & 5 & 7 \\ 3 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

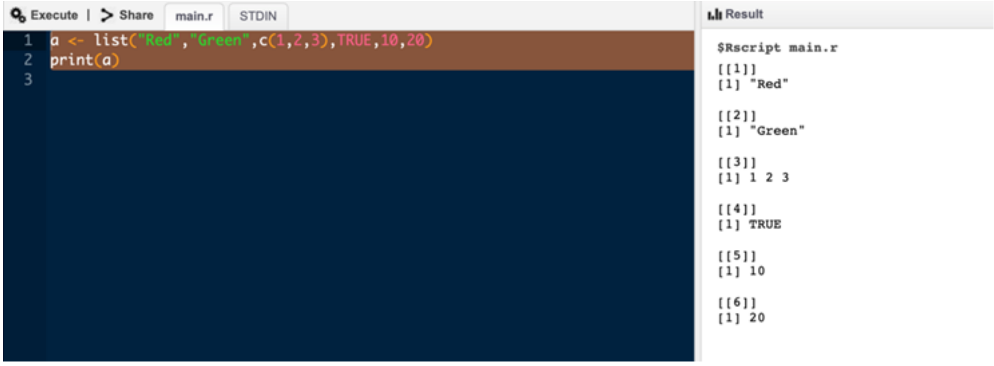


```r
A <- matrix(c(1,2,3,4),nrow=2,ncol=2)
print(A)
print('Det A')
det(A)
print('Inverse A')
solve(A)

B <- matrix(c(4,5,7,3,1,2,1,1,1),nrow=3,ncol=3)
print(B)
print('Det B')
det(B)
print('Inverse B')
solve(B)
```

```
$Rscript main.r
     [,1] [,2]
[1,]    1    3
[2,]    2    4
[1] "Det A"
[1] -2
[1] "Inverse A"
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
     [,1] [,2] [,3]
[1,]    4    3    1
[2,]    5    1    1
[3,]    7    2    1
[1] "Det B"
[1] 5
[1] "Inverse B"
     [,1] [,2] [,3]
[1,] -0.2 -0.2  0.4
[2,]  0.4 -0.6  0.2
[3,]  0.6  2.6 -2.2
```

Practical 3 -

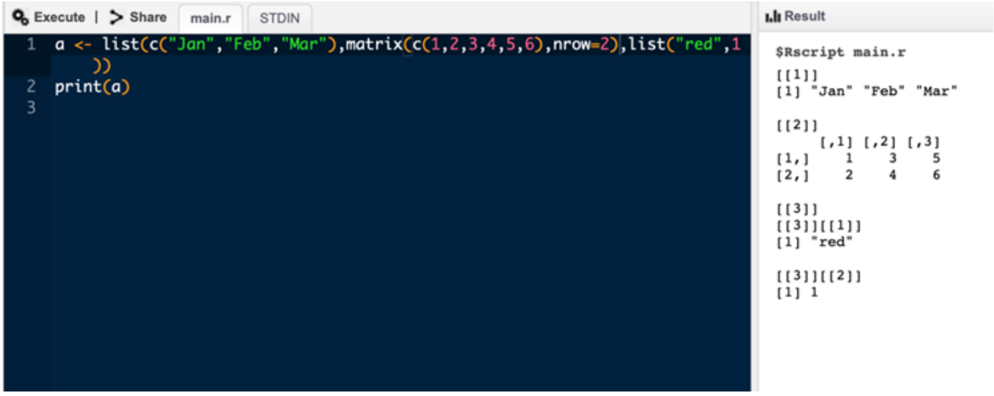
1. Create a list containing strings, numbers, vectors and a logical value.



```r
a <- list("Red","Green",c(1,2,3),TRUE,10,20)
print(a)

```

```
$Rscript main.r
[[1]]
[1] "Red"

[[2]]
[1] "Green"

[[3]]
[1] 1 2 3

[[4]]
[1] TRUE

[[5]]
[1] 10

[[6]]
[1] 20
```

2. Create a list containing a vector, a matrix and a list.



```r
a <- list(c("Jan","Feb","Mar"),matrix(c(1,2,3,4,5,6),nrow=2),list("red",1
    ))
print(a)
```

```
$Rscript main.r
[[1]]
[1] "Jan" "Feb" "Mar"

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

[[3]]
[[3]][[1]]
[1] "red"

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

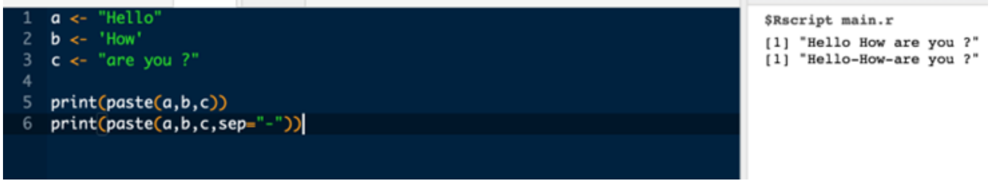
3. Manipulating List Elements - add, delete and update list elements as shown below.

```
b <- list(c("Jan","Feb","Mar"), matrix(c(1,2,3,4,5,6), nrow = 2),
   list("red",1))

# Add element “abc” at the end of the list -
l= list("deepak",10,c(1,2,3),"true")
l1=append(l,"satish kiran")

Output  : > l1
[[1]]
[1] "deepak"

[[2]]
[1] 10

[[3]]
[1] 1 2 3

[[4]]
[1] "true"

[[5]]
[1] "satish kiran"



# Remove the last element -
Input: l= list("deepak",10,c(1,2,3),"true")
l
l1=append(l,"satish kiran")
l1
l2=l1[-4]
l2

Output :
> l2
[[1]]
[1] "deepak"

[[2]]
[1] 10

[[3]]
[1] 1 2 3

[[4]]
[1] "satish kiran"



# Print the 2nd  Element -
Input: l= list("deepak",10,c(1,2,3),"true")
l
l1=append(l,"satish kiran")
l1
l2=l1[-4]
l2
l3=l2[2]
l3

Output: > l3=l2[2]
> l3
[[1]]
[1] 10
```

```
# Update the 3rd Element.(With “yes” ) -
INPUT : l= list("deepak",10,c(1,2,3),"true")
l
l1=append(l,"satish kiran")
l1
l2=l1[-4]
l2
l3=l2[2]
l3
l4=l3[3]="yes"
l4

OUTPUT; > l4=l3[3]="yes"
> l4
[1] "yes"
```

4. Using following characters create String -

a <- "Hello"
b <- 'How'
c <- "are you? "
print(paste(a,b,c))
print(paste(a,b,c,sep=“-"))



```
a <- "Hello"
b <- 'How'
c <- "are you ?"

print(paste(a,b,c))
print(paste(a,b,c,sep="-"))
```

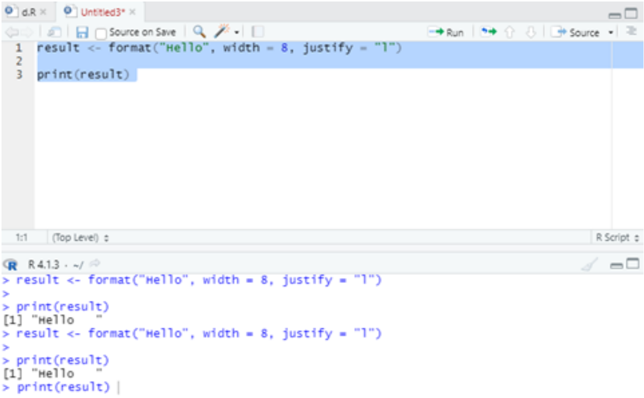
```
$Rscript main.r
[1] "Hello How are you ?"
[1] "Hello-How-are you ?"
```

5. Formatting strings.

# Left justify strings.

result <- format("Hello", width = 8, justify = "l")

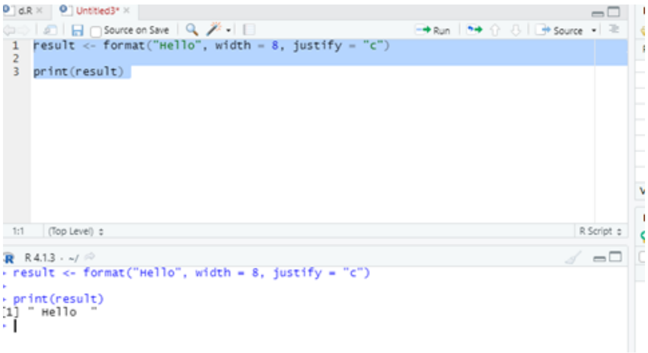
print(result)



```
result <- format("Hello", width = 8, justify = "l")

print(result)
```

```
> result <- format("Hello", width = 8, justify = "l")
>
> print(result)
[1] "Hello   "
> result <- format("Hello", width = 8, justify = "l")
>
> print(result)
[1] "Hello   "
> print(result)
```

# Justfy string with center.
result <- format("Hello", width = 8, justify = "c")
print(result)



```
result <- format("Hello", width = 8, justify = "c")

print(result)
```

```
> result <- format("Hello", width = 8, justify = "c")
>
> print(result)
[1] " Hello  "
>
```

Practical 4 -

```
# Mean and Median
# Write a code to find mean and median of vales – 15, 11, 14, 13, 18, 16. Make use of functions length(), sum() and sort(). Paste the

A = c(15,11,14,13,18,16)

mean(A)
median(A)
length(A)
sum(A)
sort(A)
```

```
# Range
# Write a code to display maximum value, minimum value, and range of the data set- 44, 62, 29, 9, 11. Paste the screenshot of code and

B = c(44,62,29,9,11)

max(B)
min(B)
range(62-9)
```

```
# Mean Deviation and Standard Deviation
# Write a code to find mean deviation, standard deviation and variance of the data set 4,5,8,2,3,6.

c = c(4,5,8,2,3,6)

mad(c) # Mean Absolute Deviation
sd(c) # Standard Deviation
var(c) # Variance
```

```
# Quartiles and Quartile Deviation
# Write a code to find Q1, Q2, Q3 and interquartile range of the data set 1, 3, 5, 7, 9. Paste the screenshot of code and output in th

data = c(1,3,5,7,9)

quantile(data,prob = c(0.25,0.50,0.75))

IQR(data)
```

```
# Mode
# Write a code to find mode of the data set 1, 3, 3, 5, 7, 9. Paste the screenshot of code and output in the box given below.

library(modeest)
a = c(1,3,3,5,7,9)
m = mfv(a)
print(m)
```

```
# Descriptive Statistics
# Write a code to find Mean, median, mode, Q1, Q2, Q3, interquartile range, mean deviation, SD and variance of the data set- 10, 20, 3

A = c(10,20,30,40,50,60,70,80,90,100)
```

```
mean(A)
median(A)
library(modeest)

m = mfv(A)
print(m)
IQR(A)

data = c(10,20,30,40,50,60,70,80,90,100)

quantile(data,prob=c(0.25,0.50,0.75))
mad(c)
sd(c)
var(c)
```

```
# Import the data from Excel/.CSV file and perform the mean and Median.
# Dataset link - CardioGoodFitness.csv

# Path - C:\Users\admin\Downloads\CardioGoodFitness.csv

myData = read.csv("C:/Users/admin/Downloads/CardioGoodFitness.csv",stringsAsFactors=F)

print(head(myData))
mean=mean(myData$Age)
print(mean)
median=median(myData$Age)
print(median
```

Practical 5A -

```
# Import the data from Excel/.CSV file and find the mean and Median of math score, reading score & writing score
# Dataset - StudentsPerformance.csv

mydata = read.csv("C:\\Users\\admin\\Downloads\\StudentsPerformance.csv")

mean_ms = mean(mydata$math.score)
print(mean_ms)
mean_rs = mean(mydata$reading.score)
print(mean_rs)

mean_wr = mean(mydata$writing.score)
print(mean_wr)
median_ms = median(mydata$math.score)
print(median_ms)

median_rs = median(mydata$reading.score)
print(median_rs)

median_wr = median(mydata$writing.score)
print(median_wr)
```

```
# Using the same dataset calculate S.D & Variance of math score, reading score & writing score.

mydata = read.csv("C:\\Users\\admin\\Downloads\\StudentsPerformance.csv")

ms = (mydata$math.score)
sd(ms)
var(ms)
rs = (mydata$reading.score)
sd(rs)
var(rs)
wr = (mydata$writing.score)
sd(wr)
var(wr)
```

```
# Define correlation and covariance of variable.
# Calculate covariance & correlation between the  math score & reading score , reading score & writing score, math score & writing sco

myData = read.csv("C:\\Users\\admin\\Downloads\\StudentsPerformance.csv")

A=cor(myData$math.score,myData$reading.score)
print(A)
B=cor(myData$reading.score,myData$writing.score)
print(B)
```

```
C=cor(myData$math.score,myData$writing.score)
print(C)
E=cov(myData$math.score,myData$reading.score)
print(E)
G=cov(myData$reading.score,myData$writing.score)
print(G)
H=cov(myData$math.score,myData$writing.score)
print(H)
```

```
# Define Skewness and Kurtosis.
# Calculate Skewness and Kurtosis of math score & reading score , reading score & writing score, math score & writing score.

library(moments)

mydata = read.csv("C:\\Users\\admin\\Downloads\\StudentsPerformance.csv")

skewness(mydata$math.score)
skewness(mydata$reading.score)
skewness(mydata$writing.score)
kurtosis(mydata$math.score)
kurtosis(mydata$reading.score)
kurtosis(mydata$writing.score)
```

Practical 5B -

```
# Import the data from Excel/.CSV file and find the mean and Median of Age and Income variables.
# Dataset link - CardioGoodFitness.csv

mydata = read.csv("C:\\Users\\admin\\Downloads\\CardioGoodFitness.csv")

print(head(mydata))
mean_sol = mean(mydata$Age)
print("Mean")
print(mean_sol)
median_sol = median(head(mydata$Age))
print("Median")
print(median_sol)
```

```
# Using the same dataset calculate S.D & Variance of Age, Income and Education.

mydata = read.csv("C:\\Users\\admin\\Downloads\\CardioGoodFitness.csv")

print(head(mydata))
sd_Age = sd(mydata$Age)
print(sd_Age)
sd_Income = sd(mydata$Income)
print(sd_Income)
sd_Education = sd(mydata$Education)
print(sd_Education)
var_Age = var(mydata$Age)
print(var_Age)
var_Income = var(mydata$Income)
print(var_Income)
var_Edu = var(mydata$Education)
print(var_Edu)
```

```
# Define correlation and covariance of variable.
# Calculate covariance & correlation between the  Age and Income , Age and Fitness variables.

mydata = read.csv("C:\\Users\\admin\\Downloads\\CardioGoodFitness.csv")

print(head(mydata))
cov(mydata$Age, mydata$Income)
cov(mydata$Age, mydata$Fitness)
cor(mydata$Age, mydata$Income)
cor(mydata$Age, mydata$Fitness)
```

```
# Define Skewness and Kurtosis.
# Calculate Skewness and Kurtosis of Income and Education.

Installing Package
Install.package(“moments”)
```

```
Library(moments)

mydata = read.csv("C:\\Users\\admin\\Downloads\\CardioGoodFitness.csv")

skewness(mydata$Income)
skewness(mydata$Education)
kurtosis(mydata$Income)
kurtosis(mydata$Education)
```