NITIN RANJAN, 18BCE0272, DATA VISUALISATION

LAB ASSESSMENT 1

1. Create the following sequences using the commands rep and seq.

   a) 1 2 3 4 5 6 7 8 9

   b) "m" "w" "m" "w" "m" "w" "m" "w" "m" "w"

   c) 1 2 3 4 1 2 3 4 1 2 3 4

   d) 4 4 4 3 3 3 2 2 2 1 1 1

   e) 1 2 2 3 3 3 4 4 4 4 5 5 5 5

   f)1 1 3 3 5 5 7 7 9 9 11 11

Answer:

```
#reg 18BCE0272
#name Nitin Ranjan
print("Nitin Ranjan, 18BCE0272")
print(1:9)
print(rep(c("M", "W"), each = 2))
print(rep(1:4,4))
print(rep(4:1,each = 3))
print(rep(c(1,2,3,4,5),times=c(1,2,3,4,5)))
print(rep(seq(1, 11, by = 2), each = 2))
```

Untitled1* ×

Source on Save

```
1  #reg 18BCE0272
2  #name Nitin Ranjan
3  print("Nitin Ranjan, 18BCE0272")
4  print(1:9)
5  print(rep(c("M", "W"), each = 2))
6  print(rep(1:4,4))
7  print(rep(4:1,each = 3))
8  print(rep(c(1,2,3,4,5),times=c(1,2,3,4,5)))
9  print(rep(seq(1, 11, by = 2), each = 2))
10
```

OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 1 2 3 4 5 6 7 8 9
[1] "M" "M" "W" "W"
 [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
 [1] 4 4 4 3 3 3 2 2 2 1 1 1
 [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
 [1]  1  1  3  3  5  5  7  7  9  9 11 11
>
```

2.

a. Try the commands sqrt(16), 16^0.5. Compute 43 . 2.

b. Try the commands log10(1000), log(1000), exp(log(1000)).

c. Try the command log2(64). Make sure you understand different logarithmic functions.

d. Try the command ?log.

e. Try the commands pi, round(pi), round(pi, digits=4), and trunc(pi).

f. The sine and cosine functions are implemented in sin and cos. Calculate sin(π), cos(π), sin(π/2), cos(π/2)

18BCE0272,
NITIN PRAMOD RANJAN

ANSWER:

```
#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

print('question 2 - a')

print(sqrt(16))

print(16^0.5)

print(43*2)

print('question 2-b')

print(log10(1000))

print(log(1000))

print(exp(log(1000)))

print('question 2-c')

print(log2(64))

#so, basically, logx(y) is log y to the base x

print(' question 2 - d')

print(?log)

#the help menu shows about logarithms and exponentials

print(' question 2 - e')

print(pi)

print(round(pi))

print(round(pi, digits = 4))

print(trunc(pi))

print('question 2 - f')

print(sin(pi))

print(cos(pi))

print(sin(pi/2))

print(cos(pi/2))
```

Untitled1* ×

Source on Save

```
1  #reg 18BCE0272
2  #name Nitin Ranjan
3  print("Nitin Ranjan, 18BCE0272")
4  print('question 2 - a')
5  print(sqrt(16))
6  print(16^0.5)
7  print(43*2)
8  print('question 2-b')
9  print(log10(1000))
10 print(log(1000))
11 print(exp(log(1000)))
12 print('question 2-c')
13 print(log2(64))
14 #so, basically, logx(y) is log y to the base x
15 print(' question 2 - d')
16 print(?log)
17 #the help menu shows about logarithms and exponentials
18 print(' question 2 - e')
19 print(pi)
20 print(round(pi))
21 print(round(pi, digits = 4))
22 print(trunc(pi))
23 print('question 2 - f')
24 print(sin(pi))
25 print(cos(pi))
26 print(sin(pi/2))
27 print(cos(pi/2))
```

OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] "question 2 - a"
[1] 4
[1] 4
[1] 86
[1] "question 2-b"
[1] 3
[1] 6.907755
[1] 1000
[1] "question 2-c"
[1] 6
[1] " question 2 - d"
[1] " question 2 - e"
[1] 3.141593
[1] 3
[1] 3.1416
[1] 3
[1] "question 2 - f"
[1] 1.224606e-16
[1] -1
[1] 1
[1] 6.123032e-17
>
```

R: Logarithms and Exponentials — Find in Topic — Refresh Help

log {base} — R Documentati

Logarithms and Exponentials

Description

`log` computes logarithms, by default natural logarithms, `log10` computes common (i.e., base 10) logarithms, and `log2` computes binary (i.e., base 2) logarithms. The general form `log(x, base)` computes logarithms with base `base`.

`log1p(x)` computes *log(1+x)* accurately also for |*x*| << *1*.

`exp` computes the exponential function.

`expm1(x)` computes *exp(x) - 1* accurately also for |*x*| << *1*.

Usage

3. Try the following and record the outputs and describe the effect of the commands

a. > 2 + 3

b. > x = 2 + 3

> x

c. > y = c(2, 3)

> y

> sum(y)

d. > v = c(5: 40)

> v

e.> length(v)

f. > v[10]

g.> v[-10]

h.> z = c(3:10)

> z + 5

i. > 2 * z

j. > w = c(6.9, 2.7, 0, -11.3, 5.5, -7.8, 4.1, 3.2)

> w + z

k.> w * z

l. > w / z

m. > w^2

18BCE0272,
NITIN PRAMOD RANJAN

Answer:

```
#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

print(2+3)

x = 2+3

print(x)

y = c(2,3)

print(y)

print(sum(y))

v = c(5:40)

print(v)

print(length(v))

print(v[10])

print(v[-10])

z = c(3:10)

print(z+5)

print(2*z)

w = c(6.9, 2.7, 0, -11.3, 5.5, -7.8, 4.1, 3.2)

print(w+z)

print(w*z)

print(w/z)

print(w^2)
```

Untitled1* ×

Source on Save

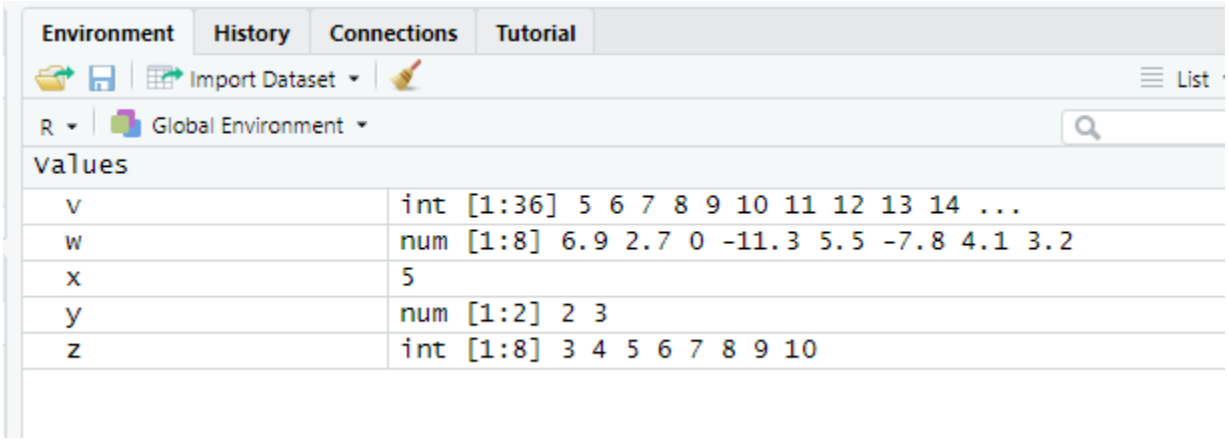
```
1  #reg 18BCE0272
2  #name Nitin Ranjan
3  print("Nitin Ranjan, 18BCE0272")
4  print(2+3)
5  x = 2+3
6  print(x)
7  y = c(2,3)
8  print(y)
9  print(sum(y))
10 v = c(5:40)
11 print(v)
12 print(length(v))
13 print(v[10])
14 print(v[-10])
15 z = c(3:10)
16 print(z+5)
17 print(2*z)
18 w = c(6.9, 2.7, 0, -11.3, 5.5, -7.8, 4.1, 3.2)
19 print(w+z)
20 print(w*z)
21 print(w/z)
22 print(w^2)
23
```

OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 5
[1] 5
[1] 2 3
[1] 5
 [1]  5  6  7  8  9 10 11 12 13 14 15
[12] 16 17 18 19 20 21 22 23 24 25 26
[23] 27 28 29 30 31 32 33 34 35 36 37
[34] 38 39 40
[1] 36
[1] 14
 [1]  5  6  7  8  9 10 11 12 13 15 16
[12] 17 18 19 20 21 22 23 24 25 26 27
[23] 28 29 30 31 32 33 34 35 36 37 38
[34] 39 40
[1]  8  9 10 11 12 13 14 15
[1]  6  8 10 12 14 16 18 20
[1]  9.9  6.7  5.0 -5.3 12.5  0.2
[7] 13.1 13.2
[1]  20.7  10.8   0.0 -67.8  38.5
[6] -62.4  36.9  32.0
[1]  2.3000000  0.6750000  0.0000000
[4] -1.8833333  0.7857143 -0.9750000
[7]  0.4555556  0.3200000
[1]  47.61   7.29   0.00 127.69
[5]  30.25  60.84  16.81  10.24
```

OBSERVATION:

The Only thing different about all these methods is the variable type in which they are stored.



Programs based on R Basics

Try the following commands and record the output along with the effects of the commands

1. > 4+6
2. Object Assignment:

   >x<-6

   >y<-4

   >z<-x+y

   >z

3. >ls()

4. > sqrt(16)

5. >rm(x,y)

6. > z<-c(5,9,1,0)

18BCE0272,
NITIN PRAMOD RANJAN

Answer

```
#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

print(4+6)

x <- 6

y <- 4

z <- x+y

print(z)

ls()

print(sqrt(16))

print(rm(x,y))

z <- c(5,9,1,0)
```

OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 10
[1] 10
[1] 4
NULL
```

The ls() generates the last string object

The rm command removes x and y from the local memory. Hence, the NULL.

2.

i. > x<-c(5,9)

   a. y<-c(1,0)

   b. z<-c(x,y)

ii. >x<-1:10

iii. >seq(1,9,by=2)

iv. >seq(8,20,length=6)

v. >x<-seq(1,10)

vi. >rep(0,100)

vii. > rep(1:3,6)

Viii > rep(1:3,c(6,6,6))

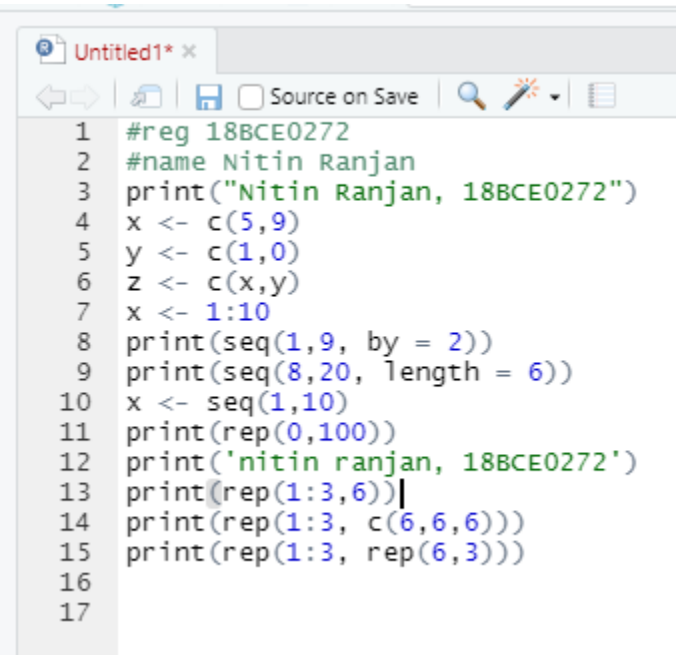
Answer

```
#reg 18BCE0272
#name Nitin Ranjan
print("Nitin Ranjan, 18BCE0272")
x <- c(5,9)
y <- c(1,0)
z <- c(x,y)
x <- 1:10
print(seq(1,9, by = 2))
print(seq(8,20, length = 6))
x <- seq(1,10)
print(rep(0,100))
print('nitin ranjan, 18BCE0272')
print(rep(1:3,6))
```

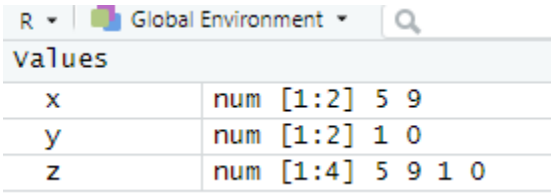
18BCE0272,
NITIN PRAMOD RANJAN

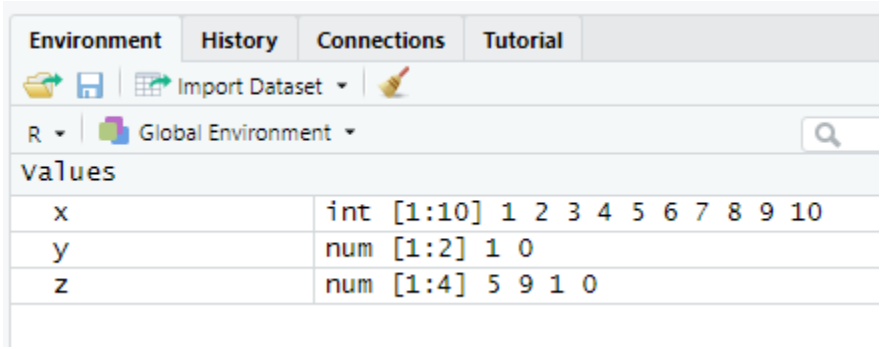
print(rep(1:3, c(6,6,6)))

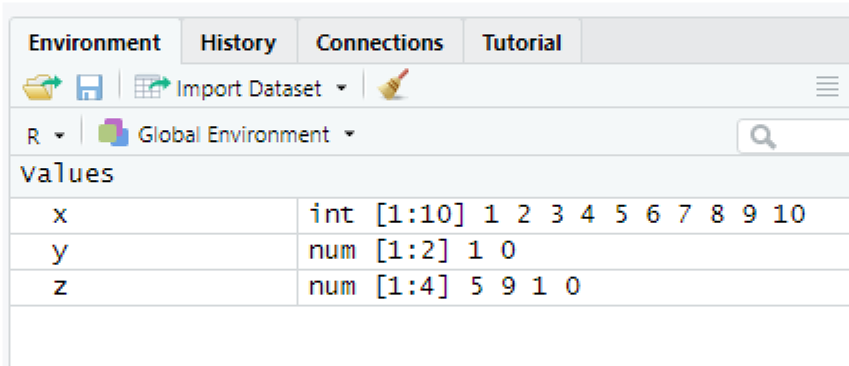
print(rep(1:3, rep(6,3)))



The variable table through this question

1. 

2. 



3.

OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 1 3 5 7 9
[1]  8.0 10.4 12.8 15.2 17.6 20.0
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [48] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [95] 0 0 0 0 0 0
[1] "nitin ranjan, 18BCE0272"
 [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
>
```

3.

>x<-c(6,8,9)

y<-c(1,2,4)

(i) >x+y

(ii) > x*y

(iii) >x<-c(6,8,9)

(iv) > x+2

ANSWER:

#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

x <- c(6,8,9)

y <- c(1,2,4)

print(x+y)

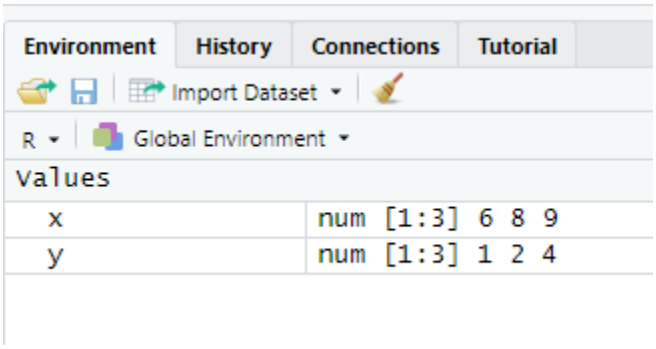
print(x*y)

18BCE0272,
NITIN PRAMOD RANJAN

```
x <- c(6,8,9)

print(x+2)
```

VARIABLE HISTORY:



OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1]  7 10 13
[1]  6 16 36
[1]  8 10 11
```

4. Define x<-c(4,2,6) y<-c(1,0,-1)

Decide what the result will be of the following:

[1] length(x)

[2] sum(x)

[3] sum(x^2)

[4] x+y

[5] x*y

[6] x-2

[7] x^2

ANSWER

```
#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

x <- c(4,2,6)

y <- c(1,0,-1)
```

```
print("length of x is: ")

print(length(x))

print('sum of x is: ')

print(sum(x))

print('x-2 is ')

print(x-2)

print('x raised to power 2 is: ')

print(x^2)
```

OUTPUT

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] "length of x is: "
[1] 3
[1] "sum of x is: "
[1] 12
[1] "x-2 is "
[1] 2 0 4
[1] "x raised to power 2 is: "
[1] 16  4 36
```

5. Decide what the following sequences are and use R to check your answers:

(i) 7:11

(ii) seq(2,9)

(iii) seq(4,10,by=2)

(iv) seq(3,30,length=10)

(v) seq(6,-4,by=-2)

ANSWER:

```
#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

print(7:11) #prints a sequence from 7 to 11
```

print(seq(2,9)) #prints a sequence from 2 to 9

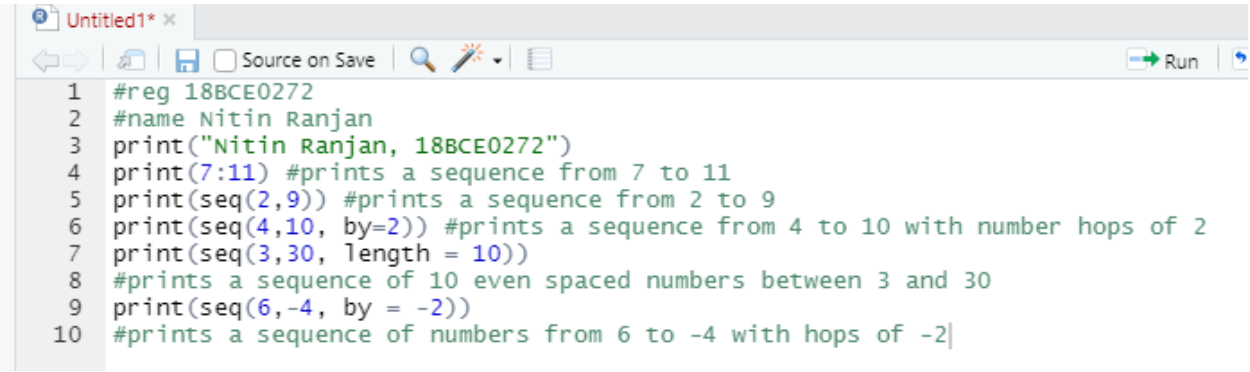
print(seq(4,10, by=2)) #prints a sequence from 4 to 10 with number hops of 2

print(seq(3,30, length = 10))

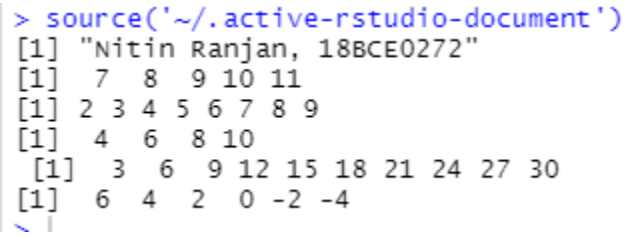
#prints a sequence of 10 even spaced numbers between 3 and 30

print(seq(6,-4, by = -2))

#prints a sequence of numbers from 6 to -4 with hops of -2



OUTPUT:



6. Determine what the result will be of the following R expressions, and then use R to check you are right:

[1]rep(2,4)

[2] rep(c(1,2),4)

[3]rep(c(1,2),c(4,4))

[4] rep(1:4,4)

[5]rep(1:4,rep(3,4))

ANSWER:

#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

print(rep(2,4)) #repeat 2 four times

print(c(1,2),4) #repeat the vector 4 times

print(rep(c(1,2), c(4,4)))

#repeat vector (1,2) with repeatitions equal to the corresponding elements in vector(4,4)

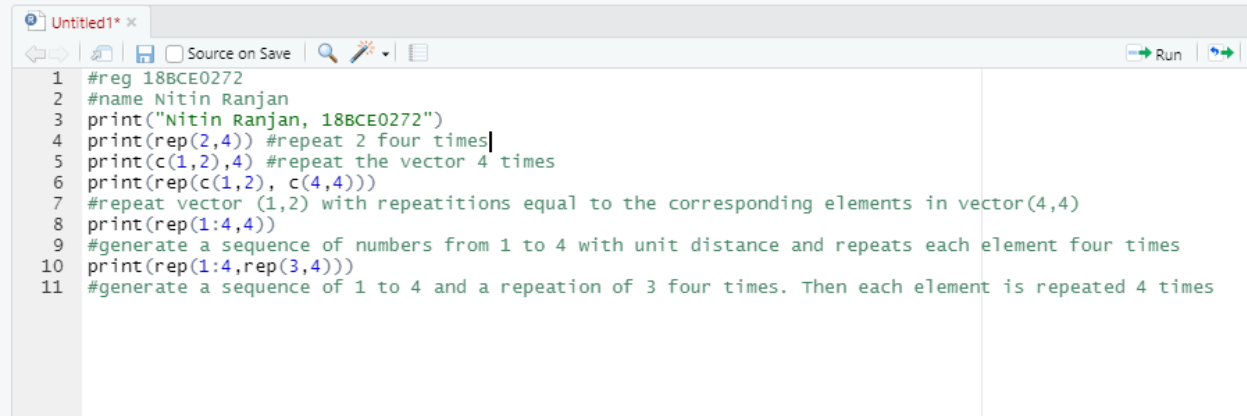
print(rep(1:4,4))

#generate a sequence of numbers from 1 to 4 with unit distance and repeats each element four times

print(rep(1:4,rep(3,4)))

#generate a sequence of 1 to 4 and a repeation of 3 four times. Then each element is repeated 4 times

18BCE0272,
NITIN PRAMOD RANJAN



OUTPUT:

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 2 2 2 2
[1] 1 2
[1] 1 1 1 1 2 2 2 2
 [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
 [1] 1 1 1 2 2 2 3 3 3 4 4 4
```

6. Consider

> x<-

c(7.5,8.2,3.1,5.6,8.2,9.3,6.5,7.0,9.3,1.2,14.5,6.2)

Find out the following:

> mean(x)

> var(x)

> summary(x)

> x[1:6]

> x[7:12]

> summary(x[1:6])

ANSWER

#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

x<-c(7.5,8.2,3.1,5.6,8.2,9.3,6.5,7.0,9.3,1.2,14.5,6.2)

print(mean(x))

print(var(x))

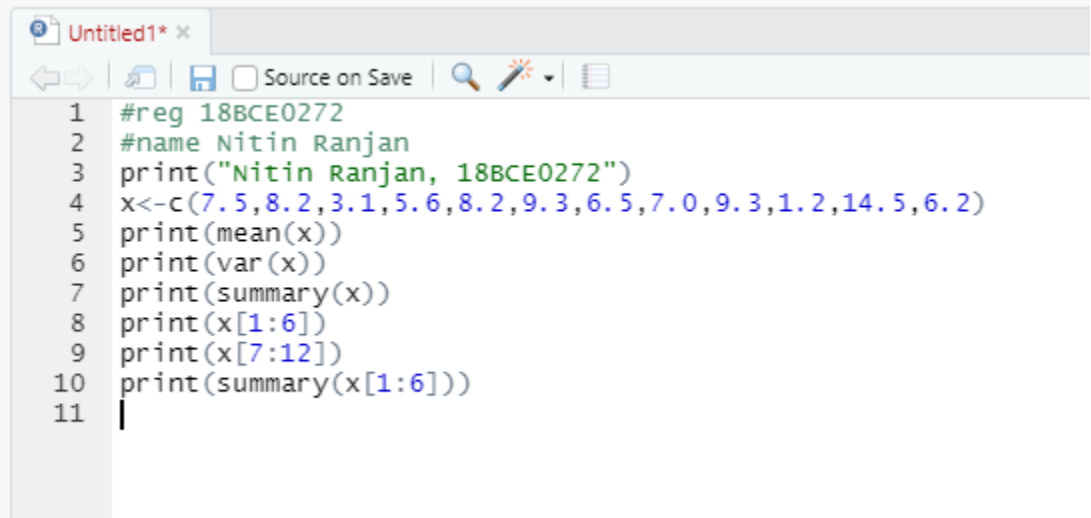
18BCE0272,
NITIN PRAMOD RANJAN

print(summary(x))

print(x[1:6])

print(x[7:12])

print(summary(x[1:6]))



OUTPUT

```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 7.216667
[1] 11.00879
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.200   6.050   7.250   7.217   8.475  14.500
[1] 7.5 8.2 3.1 5.6 8.2 9.3
[1]  6.5  7.0  9.3  1.2 14.5  6.2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.100   6.075   7.850   6.983   8.200   9.300
>
```

7. If x<- c(5,9,2,3,4,6,7,0,8,12,2,9) decide what each of the following is and use R to check your answers:

1. x[2]
2. x[2:4]
3. x[c(2,3,6)] 5
4. x[c(1:5,10:12)]
5. x[-(10:12)]

ANSWER

#reg 18BCE0272

#name Nitin Ranjan

print("Nitin Ranjan, 18BCE0272")

x<- c(5,9,2,3,4,6,7,0,8,12,2,9)

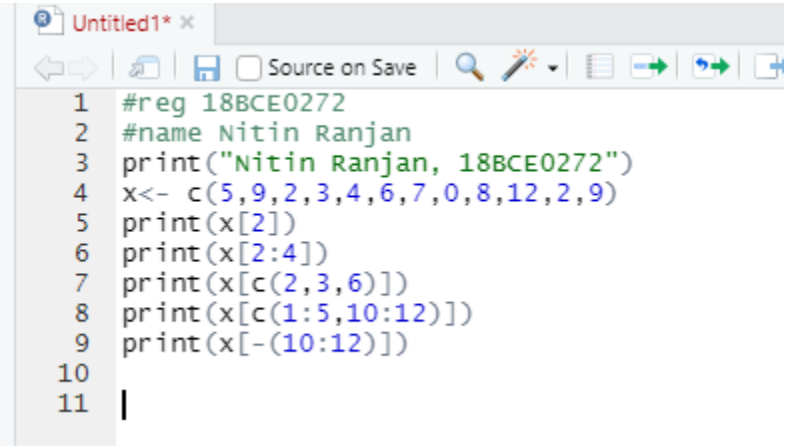
print(x[2])

print(x[2:4])

print(x[c(2,3,6)])

print(x[c(1:5,10:12)])

print(x[-(10:12)])



```
> source('~/.active-rstudio-document')
[1] "Nitin Ranjan, 18BCE0272"
[1] 9
[1] 9 2 3
[1] 9 2 6
[1]  5  9  2  3  4 12  2  9
[1] 5 9 2 3 4 6 7 0 8
> 
```

Matrices

Try the following commands

Try the following commands

1. x<-c(5,7,9)
   y<-c(6,3,4)
   z<-
   cbind(x,y) z
2. dim(z)
3. rbind(z,z)
4. z<-matrix(c(5,7,9,6,3,4),nrow=3)
5. z<-matrix(c(5,7,9,6,3,4),ncol=3)
6. z<-matrix(c(5,7,9,6,3,4),nr=3,byrow=T)
7. z<-matrix(c(5,7,9,6,3,4),nr=3,byrow=F)
8. y<-matrix(c(1,3,0,9,5,-1),nrow=3,byrow=T)
9. y+z
10. y*z
11. x<-matrix(c(3,4,-2,6),nrow=2,byrow=T)
12. y%*%x
13. t(z)
14. solve(x) ⁼ inverse of the matrix
15. Extract sub-components of matrices
    a. z[1,1]
    b. z[c(2,3),2]
    c. z[,2]
    d. z[1:2,]

OUTPUT:

```
> #18BCE0272
> x <- c(5,7,9)
> y <- c(6,3,4)
> z <- cbind(x,y)
> z
     x y
[1,] 5 6
[2,] 7 3
[3,] 9 4
> dim(z)
[1] 3 2
> rbind(z,z)
     x y
[1,] 5 6
[2,] 7 3
[3,] 9 4
[4,] 5 6
[5,] 7 3
[6,] 9 4
```

```
> print('question 4')
[1] "question 4"
> #18BCE0272
> z <- matrix(c(5,7,9,6,3,4), nrow = 3)
> z
     [,1] [,2]
[1,]    5    6
[2,]    7    3
[3,]    9    4
> z <- matrix(c(5,7,9,6,3,4), ncol =  3)
> z
     [,1] [,2] [,3]
[1,]    5    9    3
[2,]    7    6    4
> z <- matrix(c(5,7,9,6,3,4), nr=3, byrow = T)
> z
     [,1] [,2]
[1,]    5    7
[2,]    9    6
[3,]    3    4
> z <- matrix(c(5,7,9,6,3,4), nr=3, byrow = F)
> z
     [,1] [,2]
[1,]    5    6
[2,]    7    3
[3,]    9    4
```

```
> print('question 8')
[1] "question 8"
> z <- matrix(c(1,3,0,9,5,-1), nrow=3, byrow = T)
> z
     [,1] [,2]
[1,]    1    3
[2,]    0    9
[3,]    5   -1
> print('question 9')
[1] "question 9"
> y+z
     [,1] [,2]
[1,]    7    9
[2,]    3   12
[3,]    9    3
> print('question 10')
[1] "question 10"
> y*z
     [,1] [,2]
[1,]    6   18
[2,]    0   27
[3,]   20   -4
> #18BCE0272
> print('question 11')
[1] "question 11"
> x <- matrix(c(3,4,-2,6), nrow =2, byrow = T)
> x
     [,1] [,2]
[1,]    3    4
[2,]   -2    6
> #18BCE0272
> print('question 12')
[1] "question 12"
> y%*%x
Error in y %*% x : non-conformable arguments
> y% * %x
Error in y % * % x : could not find function "% * %"
```

```
> #18BCE0272
> print('question 13')
[1] "question 13"
> t(z)
     [,1] [,2] [,3]
[1,]    1    0    5
[2,]    3    9   -1
> #18BCE0272
> print('question 14')
[1] "question 14"
> solve(x)
           [,1]       [,2]
[1,] 0.23076923 -0.1538462
[2,] 0.07692308  0.1153846
> #that is one way to find inverse of x
```

```
> print('question 15')
[1] "question 15"
> z[1,1]
[1] 1
> z[c(2,3),2]
[1]  9 -1
> z[,2]
[1]  3  9 -1
> z[1:2,]
     [,1] [,2]
[1,]    1    3
[2,]    0    9
```

2.

Create in **R** the matrices

$$x = \begin{bmatrix} 3 & 2 \\ -1 & 1 \end{bmatrix}$$

and

$$y = \begin{bmatrix} 1 & 4 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Calculate the following and check your answers in R:

(a) `2*x`
(b) `x*x`
(c) `x%*%x`

ANSWER

```
> x = c(3,2)
> y = c(-1,1)
> z = rbind(x,y)
> z
  [,1] [,2]
x    3    2
y   -1    1
> #18BCE0272
> m = c(1,4,0)
> n = c(0,1,-1)
> y = rbind(m,n)
> y
  [,1] [,2] [,3]
m    1    4    0
n    0    1   -1
> z
  [,1] [,2]
x    3    2
y   -1    1
> #thus, we created both the matrices
> #now, (a)
> 2*z
  [,1] [,2]
x    6    4
y   -2    2
> #now, (b)
> z*z
  [,1] [,2]
x    9    4
y    1    1
> #now, (c)
> z%*%z
  [,1] [,2]
x    7    8
y   -4   -1
> #18BCE0272
```

2. Calculate the following:
    a. x%*%y
    b. t(y)
    c. solve(x)
3. With x and y as above, calculate the effect of the following subscript operations and check your answers in R.
    a. x[1,]
    b. x[2,]
    c. x[,2]
    d. y[1,2]
    e. y[,2:3]

CODE

```
> #quetion 2
> #2 -a
> z%*%y
  [,1] [,2] [,3]
x    3   14   -2
y   -1   -3   -1
> #2 - b       #18BCE0272, Nitin Ranjan
> t(y)
     m  n
[1,] 1  0
[2,] 4  1
[3,] 0 -1
> #2 -c
> solve(z)
        x    y
[1,] 0.2 -0.4
[2,] 0.2  0.6
> #question 3 18BCE0272, Nitin Ranjan
> z[1,]
[1] 3 2
> z[2,]
[1] -1  1
> z[,2]
x y
2 1
> y[1,2]
m
4
> y[,2:3]
  [,1] [,2]
m    4    0
n    1   -1
```

4. Create three vectors x,y,z with integers and each vector has 3 elements. Combine the three vectors to become a 3×3 matrix A where each column represents a vector. Change the row names to a,b,c.

5. Please check the result from Exercise 4, using is.matrix(A). It should return TRUE, if answer is correct. Hint: Note that is.matrix() will return FALSE on a non-matrix

   type of input.

CODE

```
> x = c(1,2,3)
> y = c(3,4,5)
> z = c(1,5,7)     #18BCE0272  Question 4
> A = matrix(c(x,y,z), nrow=3, ncol=3, byrow = F, dimnames = list( rownames))
> A
  [,1] [,2] [,3]
a    1    3    1
b    2    4    5
c    3    5    7
```

```
> #ANSWER 5   18BCE0272, NITIN RANJAN
> is.matrix(A)
[1] TRUE
>
```

6. Create a vector with 12 integers. Convert the vector to a 4*3 matrix B using matrix(). Please change the column names to x, y, z and row names to a, b, c, d.

7. Obtain the transpose matrix of B named tB .

8. Extract a sub-matrix from B named subB . It should be a 3×3 matrix which includes the last three rows of matrix B and their corresponding columns.

9. Compute A*A

10. A+subB , A-subB

```
>
> #answer 6       18bce0272, NITIN RANJAN
> m <- c(1,2,3,4,5,6,7,8,9,10,11,12)
> m
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
> B <- matrix(m, nrow = 4, ncol = 3)
> B
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
> rownames = c('a','b','c','d')
> colnames = c('x', 'y', 'z')
> B <- matrix(m, nrow = 4, ncol = 3, dimnames = list(rownames, colnames))
> B
  x y  z
a 1 5  9
b 2 6 10
c 3 7 11
d 4 8 12
> #answer 7    #18BCE0272
> tB <- t(B)
> tB
  a  b  c  d
x 1  2  3  4
y 5  6  7  8
z 9 10 11 12
```

```
> #answer 8   18BCE0272
> B[2:4,]
  x y  z
b 2 6 10
c 3 7 11
d 4 8 12
> subB = B[2:4,]
> subB
  x y  z
b 2 6 10
c 3 7 11
d 4 8 12
> A*A
  [,1] [,2] [,3]
a    1    9    1
b    4   16   25
c    9   25   49
> #answer 10   18BCE0272, nitin ranjan
> A+subB
  [,1] [,2] [,3]
a    3    9   11
b    5   11   16
c    7   13   19
> A - subB
  [,1] [,2] [,3]
a   -1   -3   -9
b   -1   -3   -6
c   -1   -3   -5
```