WEST GODAVARI INSTITUTE OF SCIENCE AND ENGINEERING

(Approved by AICTE & Affiliated to JNTUK Kakinada)
Prakasaraopalem, Avapadu, Tadepalligudem, W.G.Dist, A.P., INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

R PROGRAMMING
LABORATORY MANUAL

II B.Tech, II Semester CSE(R20)

(2021 – 2025 Batch**)**

Academic Year: 2023 - 2024

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
II Year – II Semester

L	T	P	C
0	1	2	2

R PROGRAMMING LAB

COURSE OBJECTIVES:

- To learn statistical programming, computation, graphics, and modeling,
- To learn Writing functions and use R in an efficient way,
- To learn about basic types ofstatistical models

COURSE OUTCOMES:

At the end of this course, students will be able to:

- Access online resources for R and import new function packages into the R workspace
- Import, review, manipulate and summarize data-sets in R
- Explore data-setsto create testable hypotheses and identify appropriate statistical tests
- Perform appropriate statistical tests using R
- Create and edit visualizations with R

1) Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.
2) Write a R program to get the details of the objects in memory.
3) Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.
4) Write a R program to create a simple bar plot of five subjects marks.
5) Write a R program to get the unique elements of a given string and unique numbers of vector.
6) Write a R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector. Print the content of the matrix.
7) Write a R program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.
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. 9) Write a R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50.
10) Write a R program to create an array using four given columns, three given rows, and two given tables and display the content of the array.
11) Write a R program to create an empty data frame.
12) Write a R program to create a data frame from four given vectors.
13) Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.
14) Write a R program to save the information of a data frame in a file and display the information of the file.
15) Write a R program to create a matrix from a list of given vectors.

16) Write a R program to concatenate two given matrices ofsame column but different rows.
17) Write a R program to find row and column index of maximum and minimum value in a given matrix.
18) Write a R program to append value to a given empty vector.
19) Write a R program to multiply two vectors of integers type and length 3.
20) Write a R program to find Sum, Mean and Product of a Vector, ignore element like NA or NaN.
21) Write a R program to list containing a vector, a matrix and a list and give names to the elements in the list.
22) 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.
23) Write a R program to create a list containing a vector, a matrix and a list and remove the second element.
24) Write a R program to select second element of a given nested list.
25) Write a R program to merge two given lists into one list.
26) Write a R program to create a list named s containing sequence of 15 capital letters, starting from ‘E’.
27) Write a R program to assign new names "a", "b" and "c" to the elements of a given list.
28) Write a R program to find the levels of factor of a given vector.
29) Write a R program to create an ordered factor from data consisting of the names of months.
30) Write a R program to concatenate two given factor in a single factor.

1.Write a R program to take input from the user (name and age) and display the values.Also print the version of R installation.

Source Code:

```
name = readline(prompt="Input your name: ")

age =  readline(prompt="Input your age: ")

print(paste("My name is",name, "and I am",age ,"years old."))

print(R.version.string)
```

Sample Output:

```
Input your name:
Input your age:
"My name is  and I am  years old."
"R version 3.4.4 (2018-03-15)"
```

2. Write a R program to get the details of the objects in memory.

Source Code:

```
name = "Python";

n1 =  10;

n2 =  0.5

nums = c(10, 20, 30, 40, 50, 60)

print(ls())

print("Details of the objects in memory:")

print(ls.str())
```

Sample Output:

```
"n1"   "n2"   "name" "nums"

"Details of the objects in memory:"

n1 :  num 10

n2 :  num 0.5

name :  chr "Python"

nums :  num [1:6] 10 20 30 40 50 60
```

3. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

Source Code:

```
print("Sequence of numbers from 20 to 50:")

print(seq(20,50))

print("Mean of numbers from 20 to 60:")

print(mean(20:60))

print("Sum of numbers from 51 to 91:")

print(sum(51:91))
```

Sample Output:

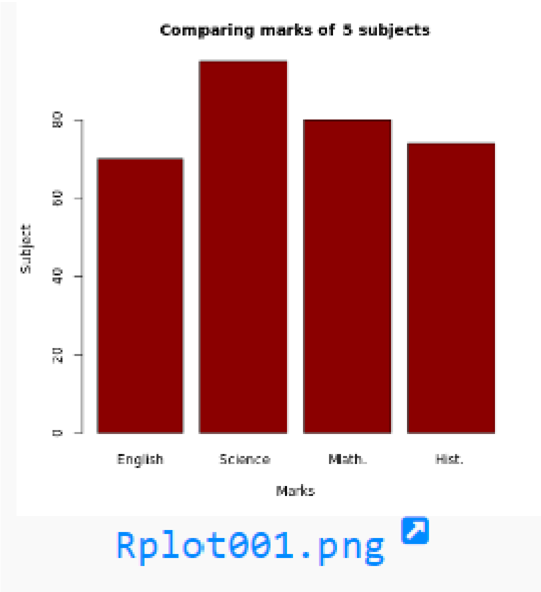
```
  "Sequence of numbers from 20 to 50:"

 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

"Mean of numbers from 20 to 60:"

40

"Sum of numbers from 51 to 91:"

2911
```

4. Write a R program to create a simple bar plot of five subjects marks.

Source Code:

```
marks = c(70, 95, 80, 74)

barplot(marks,

main = "Comparing marks of 5 subjects",

xlab = "Marks",

ylab = "Subject",

names.arg = c("English", "Science", "Math.", "Hist."),

col = "darkred",

horiz = FALSE)
```

Sample Output:



5. Write a R program to get the unique elements of a given string and unique numbers of vector.

Source Code:

```
str1 = "The quick brown fox jumps over the lazy dog."
print("Original vector(string)")
print(str1)
print("Unique elements of the said vector:")
print(unique(tolower(str1)))
nums = c(1, 2, 2, 3, 4, 4, 5, 6)
print("Original vector(number)")
print(nums)
print("Unique elements of the said vector:")
print(unique(nums))
```

Sample Output:

```
"Original vector(string)"
"The quick brown fox jumps over the lazy dog."
"Unique elements of the said vector:"
"the quick brown fox jumps over the lazy dog."
"Original vector(number)"
1 2 2 3 4 4 5 6
"Unique elements of the said vector:"
1 2 3 4 5 6
```

6. Write a R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector. Print the content of the matrix.

Source Code:

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

b<-c(4,5,6)

c<-c(7,8,9)

m<-cbind(a,b,c)

print("Content of the said matrix:")

print(m)
```

Sample Output:

```
"Content of the said matrix:"

a b c

1 4 7

2 5 8

3 6 9
```

7. Write a R program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.

Source Code:

```
m1 = matrix(1:20, nrow=5, ncol=4)

print("5 × 4 matrix:")

print(m1)

cells = c(1,3,5,7,8,9,11,12,14)

rnames = c("Row1", "Row2", "Row3")

cnames = c("Col1", "Col2", "Col3")

m2 = matrix(cells, nrow=3, ncol=3, byrow=TRUE, dimnames=list(rnames, cnames))

print("3 × 3 matrix with labels, filled by rows: ")

print(m2)

print("3 × 3 matrix with labels, filled by columns: ")

m3 = matrix(cells, nrow=3, ncol=3, byrow=FALSE, dimnames=list(rnames, cnames))

print(m3)
```

Sample Output:

```
  "5 × 4 matrix:"

 [,1] [,2] [,3] [,4]

    1    6   11   16

    2    7   12   17

    3    8   13   18

    4    9   14   19

    5   10   15   20

 "3 × 3 matrix with labels, filled by rows: "
```

```
     Col1 Col2 Col3
Row1    1    3    5
Row2    7    8    9
Row3   11   12   14
 "3 × 3 matrix with labels, filled by columns: "
     Col1 Col2 Col3
Row1    1    7   11
Row2    3    8   12
Row3    5    9   14
```

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.

Source Code:

```
num1 = rbind(rep("A",3), rep("B",3), rep("C",3))

print("num1")

print(num1)

num2 = rbind(rep("P",3), rep("Q",3), rep("R",3))

print("num2")

print(num2)

num3 = rbind(rep("X",3), rep("Y",3), rep("Z",3))

print("num3")

print(num3)

a = matrix(t(cbind(num1,num2,num3)),ncol=3, byrow=T)

print("Combine three arrays, taking one row from each one by one:")

print(a)
```

Sample Output:

```
"num1"

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

 "A"  "A"  "A"

 "B"  "B"  "B"

 "C"  "C"  "C"

 "num2"

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

 "P"  "P"  "P"
```

```
"Q"  "Q"  "Q"
 "R"  "R"  "R"
 "num3"
 [,1] [,2] [,3]
 "X"  "X"  "X"
 "Y"  "Y"  "Y"
 "Z"  "Z"  "Z"
 "Combine three arrays, taking one row from each one by one:"
  [,1] [,2] [,3]
  "A"  "A"  "A"
  "P"  "P"  "P"
  "X"  "X"  "X"
  "B"  "B"  "B"
  "Q"  "Q"  "Q"
  "Y"  "Y"  "Y"
  "C"  "C"  "C"
  "R"  "R"  "R"
  "Z"  "Z"  "Z"
```

9. Write a R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50.

Source Code:

```
a <- array(seq(from = 50, length.out = 15, by = 2), c(5, 3))

print("Content of the array:")

print("5×3 array of sequence of even integers greater than 50:")

print(a)
```

Sample Output:

```
 "Content of the array:"

"5×3 array of sequence of even integers greater than 50:"

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

 50   60   70

 52   62   72

 54   64   74

 56   66   76

 58   68   78
```

10. Write a R program to create an array using four given columns, three given rows, and two given tables and display the content of the array.

Source Code:

```
array1 =  array(1:30, dim=c(3,5,2))

print(array1)
```

Sample Output:

```
, , 1

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

   1    4    7   10   13

   2    5    8   11   14

   3    6    9   12   15

, , 2

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

   16   19   22   25   28

   17   20   23   26   29

   18   21   24   27   30
```

11. Write a R program to create an empty data frame.

Source Code:

```
df = data.frame(Ints=integer(),
            Doubles=double(),
            Characters=character(),
            Logicals=logical(),
            Factors=factor(),
            stringsAsFactors=FALSE)
print("Structure of the empty dataframe:")
print(str(df))
```

Sample Output:

```
"Structure of the empty dataframe:"
'data.frame':    0 obs. of  5 variables:
 $ Ints      : int
 $ Doubles   : num
 $ Characters: chr
$ Logicals  : logi
$ Factors   : Factor w/ 0 levels:
NULL
```

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

Source Code:

```
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas')

score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19)

attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)

qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

print("Original data frame:")

print(name)

print(score)

print(attempts)

print(qualify)

df = data.frame(name, score, attempts, qualify)

print(df)
```

Sample Output:

```
 "Original data frame:"

  "Anastasia" "Dima"      "Katherine" "James"     "Emily"     "Michael"

 "Matthew"   "Laura"     "Kevin"     "Jonas"

 12.5  9.0 16.5 12.0  9.0 20.0 14.5 13.5  8.0 19.0

 1 3 2 3 2 3 1 1 2 1

 "yes" "no"  "yes" "no"  "no"  "yes" "yes" "no"  "no"  "yes"

        name  score  attempts qualify

1  Anastasia  12.5        1      yes

2       Dima   9.0        3      no

3  Katherine  16.5        2      yes
```

```
4      James  12.0        3      no
5      Emily   9.0        2      no
6    Michael  20.0        3     yes
7    Matthew  14.5        1     yes
8      Laura  13.5        1      no
9      Kevin   8.0        2      no
10     Jonas  19.0        1     yes
```

13. Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.

Source Code:

```
a = c(10,20,10,10,40,50,20,30)

b = c(10,30,10,20,0,50,30,30)

print("Original data frame:")

ab = data.frame(a,b)

print(ab)

print("Duplicate elements of the said data frame:")

print(duplicated(ab))

print("Unique rows of the said data frame:")

print(unique(ab))
```

Sample Output:

```
 "Original data frame:"

   a  b

1  10 10

2  20 30

3  10 10

4  10 20

5 40  0

6  50 50

7  20 30

8  30 30

 "Duplicate elements of the said data frame:"
```

```
FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE

"Unique rows of the said data frame:"

  a  b

1  10 10

2  20 30

4  10 20

5  40  0

6  50 50

8  30 30
```

14. Write a R program to save the information of a data frame in a file and display the information of the file.

Source Code:

```
exam_data = data.frame(
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'),
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
save(exam_data,file="data.rda")
load("data.rda")
file.info("data.rda")
```

Sample Output:

```
 "Original dataframe:"
        name score attempts qualify
1  Anastasia  12.5        1     yes
2       Dima   9.0        3      no
3  Katherine  16.5        2     yes
4      James  12.0        3      no
5      Emily   9.0        2      no
6    Michael  20.0        3     yes
7    Matthew  14.5        1     yes
```

```
8      Laura  13.5       1      no
9      Kevin   8.0       2      no
10     Jonas  19.0       1     yes
         size isdir mode               mtime               ctime
data.rda  344 FALSE  644 2018-10-25 12:06:09 2018-10-25 12:06:09
         atime  uid  gid   uname  grname
data.rda 2018-10-25 12:06:09 1000 1000 trinket trinket
```

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

Source Code:

```
l = list()

for (i in 1:5) l[[i]] <-  c(i, 1:4)

print("List of vectors:")

print(l)

result = do.call(rbind, l)

print("New Matrix:")

print(result)
```

Sample Output:

```
 "List of vectors:"

[[1]]

 1 1 2 3 4

[[2]]

 2 1 2 3 4

[[3]]

 3 1 2 3 4

[[4]]

 4 1 2 3 4

[[5]]

 5 1 2 3 4

"New Matrix:"

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

   1    1    2    3    4

   2    1    2    3    4
```

```
3   1   2   3   4

4   1   2   3   4

5   1   2   3   4
```

16. Write a R program to concatenate two given matrices of same column but different rows.

Source Code:

```
x = matrix(1:12, ncol=3)

y = matrix(13:24, ncol=3)

print("Matrix-1")

print(x)

print("Matrix-2")

print(y)

result = dim(rbind(x,y))

print("After concatenating two given matrices:")

print(result)
```

Sample Output:

```
 "Matrix-1"

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

  1    5    9

  2    6   10

  3    7   11

  4    8   12

 "Matrix-2"

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

  13   17   21

 14   18   22

  15   19   23
 16   20   24

"After concatenating two given matrices:"

8 3
```

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

Source Code:

```
m = matrix(c(1:16), nrow = 4, byrow = TRUE)

print("Original Matrix:")

print(m)

result = which(m == max(m), arr.ind=TRUE)

print("Row and column of maximum value of the said matrix:")

print(result)

result = which(m == min(m), arr.ind=TRUE)

print("Row and column of minimum value of the said matrix:")

print(result)
```

Sample Output:

```
"Original Matrix:"

 [,1] [,2] [,3] [,4]

1    2    3    4

5    6    7    8

 9   10   11   12

13   14   15   16

"Row and column of maximum value of the said matrix:"

row col

4   4

"Row and column of minimum value of the said matrix:"

 row col

  1   1
```

18. Write a R program to append value to a given empty vector.

Source Code:

```
vector = c()
values = c(0,1,2,3,4,5,6,7,8,9)
for (i in 1:length(values))
  vector[i] <- values[i]
print(vector)
```

Sample Output:

```
 0 1 2 3 4 5 6 7 8 9
```

19. Write a R program to multiply two vectors of integers type and length 3.

Source Code:

```
x = c(10, 20, 30)
 y = c(20, 10, 40)
 print("Original Vectors:")
 print(x)
 print(y)
 print("Product of two Vectors:")
 z = x / y
 print(z)
```

Sample Output:

```
 "Original Vectors:"
 10 20 30
 20 10 40
 "Product of two Vectors:"
 0.50 2.00 0.75
```

20. Write a R program to find Sum, Mean and Product of a Vector, ignore element like NA or NaN.

Source Code:

```
x = c(10, NULL, 20, 30, NA)

print("Sum:")

#ignore NA and NaN values

print(sum(x, na.rm=TRUE))

print("Mean:")

print(mean(x, na.rm=TRUE))

print("Product:")

print(prod(x, na.rm=TRUE))
```

Sample Output:

```
 "Sum:"

 60

 "Mean:"

 20

 "Product:"

 6000
```

21. Write a R program to list containing a vector, a matrix and a list and give names to the elements in the list.

Source Code:

```
list_data <- list(c("Red","Green","Black"), matrix(c(1,3,5,7,9,11), nrow = 2),
list("Python", "PHP", "Java"))
print("List:")
print(list_data)
names(list_data) = c("Color", "Odd numbers", "Language(s)")
print("List with column names:")
print(list_data)
print('1st element:')
print(list_data[1])
print('2nd element:')
print(list_data[2])
```

Sample Output:

```
 "List:"
[[1]]
"Red"   "Green" "Black"
[[2]]
[,1] [,2] [,3]
   1    5    9
   3    7   11
[[3]]
[[3]][[1]]
 "Python"
```

```
[[3]][[2]]
 "PHP"
[[3]][[3]]
 "Java"
 "List with column names:"
$Color
 "Red"   "Green" "Black"
$`Odd numbers`
[,1] [,2] [,3]
   1    5    9
   3    7   11
$`Language(s)`
$`Language(s)`[[1]]
 "Python"
$`Language(s)`[[2]]
 "PHP"
$`Language(s)`[[3]]
 "Java"
 "1st element:"
$Color
 "Red"   "Green" "Black"
 "2nd element:"
$`Odd numbers`
[,1] [,2] [,3]
```

1 5 9

3 7 11

22. 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.

Source Code:

```
list_data <- list(c("Red","Green","Black"), matrix(c(1,3,5,7,9,11), nrow = 2),
list("Python", "PHP", "Java"))
print("List:")
print(list_data)
names(list_data) = c("Color", "Odd numbers", "Language(s)")
print("List with column names:")
print(list_data)
print('1st element:')
print(list_data[1])
print('2nd element:')
print(list_data[2])
```

Sample Output:

```
 "List:"
[[1]]
"Red"   "Green" "Black"
[[2]]
     [,1] [,2] [,3]
    1    5    9
    3    7   11
[[3]]
[[3]][[1]]
```

```
 "Python"

[[3]][[2]]

 "PHP"

[[3]][[3]]

 "Java"

 "List with column names:"

$Color

 "Red"   "Green" "Black"

$`Odd numbers`

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

    1    5    9

    3    7   11

$`Language(s)`

$`Language(s)`[[1]]

 "Python"

$`Language(s)`[[2]]

 "PHP"

$`Language(s)`[[3]]

 "Java"

 "1st element:"

$Color

 "Red"   "Green" "Black"

 "2nd element:"

$`Odd numbers`
```

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

1    5    9

3    7   11
```

23. Write a R program to create a list containing a vector, a matrix and a list and remove the second element.

Source Code:

```
list_data <- list(c("Red","Green","Black"), matrix(c(1,3,5,7,9,11), nrow = 2),

list("Python", "PHP", "Java"))

print("List:")

print(list_data)

print("Remove the second element of the list:")

list_data[2] = NULL

print("New list:")

print(list_data)
```

Sample Output:

```
 "List:"

[[1]]

 "Red"   "Green" "Black"

[[2]]

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

    1    5    9

   3    7   11

[[3]]

[[3]][[1]]

 "Python"

[[3]][[2]]

 "PHP"
```

```
[[3]][[3]]
 "Java"
 "Remove the second element of the list:"
 "New list:"
[[1]]
 "Red"   "Green" "Black"
[[2]]
[[2]][[1]]
 "Python"
[[2]][[2]]
 "PHP"
[[2]][[3]]
 "Java"
```

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

Source Code:

```
x = list(list(0,2), list(3,4), list(5,6))

print("Original nested list:")

print(x)

e = lapply(x, '[[', 2)

print("Second element of the nested list:")

print(e)
```

Sample Output:

```
 "Original nested list:"

[[1]]

[[1]][[1]]

 0

[[1]][[2]]

 2

[[2]]

[[2]][[1]]

 3

[[2]][[2]]

 4

[[3]]

[[3]][[1]]

 5

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

```
 6

 "Second element of the nested list:"

[[1]]

 2

[[2]]

 4

[[3]]

 6
```

25. Write a R program to merge two given lists into one list.

Source Code:

```
n1 = list(1,2,3)

c1 = list("Red", "Green", "Black")

print("Original lists:")

print(n1)

print(c1)

print("Merge the said lists:")

mlist =  c(n1, c1)

print("New merged list:")

print(mlist)
```

Sample Output:

```
 "Original lists:"

[[1]]

 1

[[2]]

 2

[[3]]

 3

[[1]]

 "Red"

[[2]]

 "Green"

[[3]]
```

```
 "Black"

 "Merge the said lists:"

 "New merged list:"

[[1]]

 1

[[2]]

 2

[[3]]

 3

[[4]]

 "Red"

[[5]]

 "Green"

[[6]]

 "Black"
```

26. Write a R program to create a list named s containing sequence of 15 capital letters, starting from ‘E’.

Source Code:

```
l = LETTERS[match("E", LETTERS):(match("E", LETTERS)+15)]

print("Content of the list:")

print("Sequence of 15 capital letters, starting from ‘E’-")

print(l)
```

Sample Output:

```
 "Content of the list:"

 "Sequence of 15 capital letters, starting from ‘E’-"

 "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T"
```

27. Write a R program to assign new names "a", "b" and "c" to the elements of a given list.

Source Code:

Sample list: (g1 = 1:10, g2 = "R Programming", g3 = "HTML")

```
list1 = list(g1 = 1:10, g2 = "R Programming", g3 = "HTML")

print("Original list:")

print(list1)

names(list1) = c("one", "two", "three")

print("Assign new names 'one', 'two' and 'three' to the elements of the said list")

print(list1)
```

Sample Output: "Original list:"

```
$g1

  1  2  3  4  5  6  7  8  9 10

$g2

 "R Programming"

$g3

 "HTML"

 "Assign new names 'one', 'two' and 'three' to the elements of the said list"

$one

  1  2  3  4  5  6  7  8  9 10

$two

 "R Programming"   $three

 "HTML"
```

28. Write a R program to find the levels of factor of a given vector.

ALGORITHM

- STEP 1 : Assign variable v with vector values
- STEP 2 : Use the built-in functions
- STEP 3 : First print original vector values
- STEP 4 : levels(factor(v) with an argument as v to find levels of factors
- STEP 5 : print the result of the function

Source Code:

```
v = c(1, 2, 3, 3, 4, NA, 3, 2, 4, 5, NA, 5)

print("Original vector:")

print(v)

print("Levels of factor of the said vector:")

print(levels(factor(v)))
```

Sample Output:

```
"Original vector:"

 1  2  3  3  4 NA  3  2  4  5 NA  5

"Levels of factor of the said vector:"

"1" "2" "3" "4" "5"
```

29. Write a R program to create an ordered factor from data consisting of the names of months.

Source Code:

```
mons_v = c("March","April","January","November","January",

"September","October","September","November","August","February",

"January","November","November","February","May","August","February",

"July","December","August","August","September","November","September",

"February","April")

print("Original vector:")

print(mons_v)

f = factor(mons_v)

print("Ordered factors of the said vector:")

print(f)

print(table(f))
```

Sample Output:

```
 "Original vector:"

 "March"    "April"    "January"   "November"  "January"   "September"

 "October"   "September" "November"  "August"    "February"  "January"

 "November"  "November"  "February"  "May"       "August"    "February"

 "July"      "December"  "August"    "August"    "September" "November"

 "September" "February"  "April"

 "Ordered factors of the said vector:"

 March     April     January   November  January   September October

 September November  August    February  January   November  November
 February  May       August    February  July      December  August
 August    September November  September February  April
11 Levels: April August December February January July March May ... September
F
  April    August  December  February   January     July     March       May
      2        4        1        4        3        1        1        1
 November   October September
      5        1        4
```

30. Write a R program to concatenate two given factor in a single factor.

ALGORITHM

1. STEP 1: Assign variable fact1,fact2 with factor values.
2. STEP 2: First print original factors values.
3. STEP 3:Call the built-in function factor with level as factor(c(levels(fact1)[fact1], levels(fact2)[fact2]))
4. STEP 4: Assign variable fact with the function result.
5. STEP 5: Print the concatenated factor.

Source Code:

```
f1 <- factor(sample(LETTERS, size=6, replace=TRUE))

f2 <- factor(sample(LETTERS, size=6, replace=TRUE))

print("Original factors:")

print(f1)

print(f2)

f = factor(c(levels(f1)[f1], levels(f2)[f2]))

print("After concatenate factor becomes:")

print(f)
```

Sample Output:

```
 "Original factors:"

 Q Y M J J H

Levels: H J M Q Y

 B J L S F Z

Levels: B F J L S Z

 "After concatenate factor becomes:"

 Q Y M J J H B J L S F Z

Levels: B F H J L M Q S Y Z
```