

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.

Program:

```
name <- readline("Enter your name: ")
age <- as.numeric(readline("Enter your age: "))
# Display the input value
cat("Name: ", name, "\n")
cat("Age: ", age, "\n")
# Print the version of R installation
cat("R Version: ", R.version$version.string, "\n")
```

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

```
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())
```

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

```
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))
```

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

```
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)
```

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

```
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))
```

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

```
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)
```

\

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 2x 2matrix with labels and fill the matrix by columns.

```
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)
```

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

```
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)
```

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

```
a <- array(seq(from = 50, length.out = 15, by = 2), c(5, 3))
print("Content of the array:")
print("5x3 array of sequence of even integers greater than 50:")
print(a)
```

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.

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

```
print(array1)
```

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

```
df = data.frame(Ints=integer(),
```

```
    Doubles=double(),
```

```
    Characters=character(),
```

```
    Logicals=logical(),
```

```
    Factors=factor(),
```

```
    stringsAsFactors=FALSE)
```

```
print("Structure of the empty dataframe:")
```

```
print(str(df))
```

12) Write a R program to create a dataframe from four given vectors.

```
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)
```

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.

```
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))
```

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

```
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")
```

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

```
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)
```

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

```
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)
```

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

```
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)
```

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

```
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)
```

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

```
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)
```

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

```
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))
.
```

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

```
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])
```

22. Write a R program to create a list contain in 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.

```
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("Add a new element at the end of the list:")  
list_data[4] = "New element"  
print("New list:")
```

```
print(list_data)
```

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

```
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)
```

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

```
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)
```

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

```
n1 = list(1,2,3)
c1 = list("Red", "Green", "Black")
print("Original lists:")
print(n1)
print(c1)
print("Merge the said lists:")
m1 = c(n1, c1)
print("New merged list:")
print(m1)
```

26. Write a R program to create a list nameds containing sequence of 15 capital letters, starting from 'E'.

```
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)
```

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

```
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)
```

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

```
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)))
```

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

```
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))
```

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

```
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)
```


