

BASIC

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.**

```
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)
```

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 vector which contains 10 random integer values between -50 and +50.**

```
v = sample(-50:50, 10, replace=TRUE)
print("Content of the vector:")
print("10 random integer values between -50 and +50:")
print(v)
```

5. **Write a R program to get the first 10 Fibonacci numbers.**

```
Fibonacci <- numeric(10)
Fibonacci[1] <- Fibonacci[2] <- 1
for (i in 3:10) Fibonacci[i] <- Fibonacci[i - 2] + Fibonacci[i - 1]
print("First 10 Fibonacci numbers:")
print(Fibonacci)
```

6. **Write a R program to get all prime numbers up to a given number (based on the sieve of Eratosthenes).**

```
prime_numbers <- function(n)
{
```

```

if (n >= 2) {
  x = seq(2, n)
  prime_nums = c()
  for (i in seq(2, n)) {
    if (any(x == i))
    {
      prime_nums = c(prime_nums, i)
      x = x[(x %% i) != 0], i)
    }
  }
  return(prime_nums)
}
else
{
  stop("Input number should be at least 2.")
}
}
prime_numbers(12)

```

7. **Write a R program to print the numbers from 1 to 100 and print "Fizz" for multiples of 3, print "Buzz" for multiples of 5, and print "FizzBuzz" for multiples of both.**

```

for (n in 1:100) {
  if (n %% 3 == 0 & n %% 5 == 0) {print("FizzBuzz")}
  else if (n %% 3 == 0) {print("Fizz")}
  else if (n %% 5 == 0) {print("Buzz")}
  else print(n)
}

```

8. **Write a R program to extract first 10 english letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.**

```

print("First 10 letters in lower case:")
t = head(letters, 10)
print(t)
print("Last 10 letters in upper case:")
t = tail(LETTERS, 10)
print(t)
print("Letters between 22nd to 24th letters in upper case:")
e = tail(LETTERS[22:24])
print(e)

```

9. **Write a R program to find the factors of a given number.**

```

print_factors = function(n) {
  print(paste("The factors of",n,"are:"))
}

```

```
for(i in 1:n) {  
  if((n %% i) == 0) {  
    print(i)  
  }  
}  
  
print_factors(4)  
print_factors(7)  
print_factors(12)
```

10. **Write a R program to find the maximum and the minimum value of a given vector.**

```
nums = c(10, 20, 30, 40, 50, 60)  
print('Original vector:')  
print(nums)  
print(paste("Maximum value of the said vector:",max(nums)))  
print(paste("Minimum value of the said vector:",min(nums)))
```

11. **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))
```

12. **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)
```

13. **Write a R program to create a list of random numbers in normal distribution and count occurrences of each value.**

```
n = floor(rnorm(1000, 50, 100))
print('List of random numbers in normal distribution:')
print(n)
t = table(n)
print("Count occurrences of each value:")
print(t)
```

14. **Write a R program to read the .csv file and display the content.**

```
movie_data = read.csv(file="movies.csv", header=TRUE, sep=",")
print("Content of the .csv file:")
print(movie_data)
```

15. **Write a R program to create three vectors numeric data, character data and logical data. Display the content of the vectors and their type.**

```

a = c(1, 2, 5, 3, 4, 0, -1, -3)
b = c("Red", "Green", "White")
c = c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)
print(a)
print(typeof(a))
print(b)
print(typeof(b))
print(c)
print(typeof(c))

```

- 16. Write a R program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill the matrix by rows and 3 x 3 matrix with labels and fill the matrix by columns.**

```

m1 = matrix(1:20, nrow=5, ncol=4)
print("5 x 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 x 3 matrix with labels, filled by rows: ")
print(m2)
print("3 x 3 matrix with labels, filled by columns: ")
m3 = matrix(cells, nrow=3, ncol=3, byrow=FALSE, dimnames=list(rnames, cnames))
print(m3)

```

- 17. Write a R program to create an array, passing in a vector of values and a vector of dimensions. Also provide names for each dimension.**

```

a = array(
  6:30,
  dim = c(4, 3, 2),
  dimnames = list(
    c("Col1", "Col2", "Col3", "Col4"),
    c("Row1", "Row2", "Row3"),
    c("Part1", "Part2")
  )
)
print(a)

```

18. **Write a R program to create an array with three columns, three rows, and two "tables", taking two vectors as input to the array. Print the array.**

```
v1 = c(1, 3, 5, 7)
v2 = c(2, 4, 6, 8, 10)
arra1 = array(c(v1, v2), dim = c(3,3,2))
print(arra1)
```

19. **Write a R program to create a list of elements using vectors, matrices and a functions. Print the content of the list.**

```
l = list(
  c(1, 2, 2, 5, 7, 12),
  month.abb,
  matrix(c(3, -8, 1, -3), nrow = 2),
  asin
)
print("Content of the list:")
print(l)
```

20. **Write a R program to draw an empty plot and an empty plot specify the axes limits of the graphic.**

```
#print("Empty plot:")
plot.new()
#print("Empty plot specify the axes limits of the graphic:")
plot(1, type="n", xlab="", ylab="", xlim=c(0, 20), ylim=c(0, 20))
```

21. **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)
```

22. **Write a R program to create bell curve of a random normal distribution.**

```
n = floor(rnorm(10000, 500, 100))
t = table(n)
```

```
barplot(t)
```

- 23. Write a R program to compute sum, mean and product of a given vector elements.**

```
nums = c(10, 20, 30)
print('Original vector:')
print(nums)
print(paste("Sum of vector elements:",sum(nums)))
print(paste("Mean of vector elements:",mean(nums)))
print(paste("Product of vector elements:",prod(nums)))
```

- 24. Write a R program to create a list of heterogeneous data, which include character, numeric and logical vectors. Print the lists.**

```
my_list = list(Chr="Python", nums = 1:15, flag=TRUE)
print(my_list)
```

- 25. Write a R program to create a Dataframes which contain details of 5 employees and display the details.**

```
Employees = data.frame(Name=c("Anastasia S","Dima R","Katherine S", "JAMES A","LAURA MARTIN"),
  Gender=c("M","M","F","F","M"),
  Age=c(23,22,25,26,32),
  Designation=c("Clerk","Manager","Exective","CEO","ASSISTANT"),
  SSN=c("123-34-2346","123-44-779","556-24-433","123-98-987","679-77-576")
)
print("Details of the employees:")
print(Employees)
```

- 26. Write a R program to create a Data Frames which contain details of 5 employees and display summary of the data.**

```
Employees = data.frame(Name=c("Anastasia S","Dima R","Katherine S", "JAMES A","LAURA MARTIN"),
  Gender=c("M","M","F","F","M"),
  Age=c(23,22,25,26,32),
  Designation=c("Clerk","Manager","Exective","CEO","ASSISTANT"),
  SSN=c("123-34-2346","123-44-779","556-24-433","123-98-987","679-77-576")
)
print("Summary of the data:")
print(summary(Employees))
```

- 27. Write a R program to create the system's idea of the current date with and without time.**

```
print("System's idea of the current date with and without time:")
print(Sys.Date())
print(Sys.time())
```

ARRAY

- 28. Write a R program to convert a given matrix to a 1 dimensional array.**

```

m=matrix(1:12,3,4)
print("Original matrix:")
print(m)
a = as.vector(m)
print("1 dimensional array:")
print(a)

```

- 29. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors.**

```

print("Two vectors of different lengths:")
v1 = c(1,3,4,5)
v2 = c(10,11,12,13,14,15)
print(v1)
print(v2)
result = array(c(v1,v2),dim = c(3,3,2))
print("New array:")
print(result)

```

- 30. Write a R program to create an 3 dimensional array of 24 elements using the dim() function.**

```

v = sample(1:5,24,replace = TRUE)
dim(v) = c(3,2,4)
print("3-dimension array:")
print(v)

```

- 31. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.**

```

print("Two vectors of different lengths:")
v1 = c(1,3,4,5)
v2 = c(10,11,12,13,14,15)
print(v1)
print(v2)
result = array(c(v1,v2),dim = c(3,3,2))
print("New array:")
print(result)
print("The second row of the second matrix of the array:")
print(result[2,,2])
print("The element in the 3rd row and 3rd column of the 1st matrix:")
print(result[3,3,1])

```

- 32. 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)

```

- 33. 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)

```

- 34. 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)

```

DATAFRAME

- 35. 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))

```

- 36. Write a R program to create a data frame 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)
```

37. Write a R program to get the structure of a given data frame.

```
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)
print("Structure of the said data frame:")
print(str(exam_data))
```

38. Write a R program to get the statistical summary and nature of the data of a given data frame.

```
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)
print("Statistical summary and nature of the data of the said dataframe:")
print(summary(exam_data))
```

39. Write a R program to extract specific column from a data frame using column name.

```
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)
print("Extract Specific columns:")
result <- data.frame(exam_data$name, exam_data$score)
print(result)
```

40. Write a R program to extract first two rows from a given data frame.

```
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)
print("Extract first two rows:")
result = exam_data[1:2,]
print(result)

```

41. Write a R program to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame.

```

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)
print("Extract 3rd and 5th rows with 1st and 3rd columns :")
result = exam_data[c(3,5),c(1,3)]
print(result)

```

42. Write a R program to add a new column in a given data frame.

```

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)
print("New data frame after adding the 'country' column:")
exam_data$country = c("USA", "USA", "USA", "USA", "USA", "USA", "USA", "USA", "USA", "USA")
print(exam_data)

```

43. Write a R program to add new row(s) to an existing data frame.

```

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)
new_exam_data = data.frame(
name = c('Robert', 'Sophia'),
score = c(10.5, 9),

```

```

attempts = c(1, 3),
qualify = c('yes', 'no')
)
exam_data = rbind(exam_data, new_exam_data)
print("After adding new row(s) to an existing data frame:")
print(exam_data)

```

44. Write a R program to drop column(s) by name from a given data frame.

```

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)
exam_data = subset(exam_data, select = -c(name, qualify))
print(exam_data)

```

45. Write a R program to drop row(s) by number from a given data frame.

```

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)
exam_data <- exam_data[-c(2, 4, 6), ]
print(exam_data)

```

46. Write a R program to sort a given data frame by multiple column(s).

```

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)
print("dataframe after sorting 'name' and 'score' columns:")
exam_data = exam_data[with(exam_data, order(name, score)), ]
print(exam_data)

```

47. Write a R program to create inner, outer, left, right join (merge) from given two data frames.

```

df1 = data.frame(numid = c(12, 14, 10, 11))

```

```

df2 = data.frame(numid = c(13, 15, 11, 12))
print("Left outer Join:")
result = merge(df1, df2, by = "numid", all.x = TRUE)
print(result)
print("Right outer Join:")
result = merge(df1, df2, by = "numid", all.y = TRUE)
print(result)
print("Outer Join:")
result = merge(df1, df2, by = "numid", all = TRUE)
print(result)
print("Cross Join:")
result = merge(df1, df2, by = NULL)
print(result)

```

48. Write a R program to replace NA values with 3 in a given data frame.

```

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, NA, 2, NA, 2, NA, 1, NA, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
exam_data[is.na(exam_data)] = 3
print("After removing NA with 3, the said dataframe becomes:")
print(exam_data)

```

49. Write a R program to change a column name of a given data frame.

```

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, NA, 2, NA, 2, NA, 1, NA, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
print("Change column-name 'name' to 'student_name' of the said dataframe:")
colnames(exam_data)[which(names(exam_data) == "name")] = "student_name"
print(exam_data)

```

50. Write a R program to change more than one column name of a given data frame.

```

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, NA, 2, NA, 2, NA, 1, NA, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)

```

```

)
print("Original dataframe:")
print(exam_data)
print("Change more than one column name of the said dataframe:")
colnames(exam_data)[which(names(exam_data) == "name")] = "student_name"
colnames(exam_data)[which(names(exam_data) == "score")] = "avg_score"
print(exam_data)

```

51. Write a R program to select some random rows from a given data frame.

```

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)
print("Select three random rows of the said dataframe:")
print(exam_data[sample(nrow(exam_data), 3),])

```

52. Write a R program to reorder an given data frame by column name.

```

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)
print("Reorder by column name:")
exam_data = exam_data[c("name", "attempts", "score", "qualify")]
print(exam_data)

```

53. Write a R program to compare two data frames to find the row(s) in first data frame that are not present in second data frame.

```

df_90 = data.frame(
  "item" = c("item1", "item2", "item3"),
  "Jan_sale" = c(12, 14, 12),
  "Feb_sale" = c(11, 12, 15),
  "Mar_sale" = c(12, 14, 15)
)
df_91 = data.frame(
  "item" = c("item1", "item2", "item3"),
  "Jan_sale" = c(12, 14, 12),
  "Feb_sale" = c(11, 12, 15),
  "Mar_sale" = c(12, 15, 18)
)

```

```
)
print("Original Dataframes:")
print(df_90)
print(df_91)
print("Row(s) in first data frame that are not present in second data frame:")
print(setdiff(df_90,df_91))
```

54. Write a R program to find elements which are present in two given data frames.

```
a = c("a", "b", "c", "d", "e")
b = c("d", "e", "f", "g")
print("Original Dataframes")
print(a)
print(b)
print("Elements which are present in both dataframe:")
result = intersect(a, b)
print(result)
```

55. Write a R program to find elements come only once that are common to both given data frames.

```
a = c("a", "b", "c", "d", "e")
b = c("d", "e", "f", "g")
print("Original Dataframes")
print(a)
print(b)
print("Find elements come only once that are common to both given dataframes:")
result = union(a, b)
print(result)
```

56. 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")
```

57. Write a R program to count the number of NA values in a data frame column.

```
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, NA, 2, NA, 2, NA, 1, NA, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
print("The number of NA values in attempts column:")
print(sum(is.na(exam_data$attempts)))

```

- 58. 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))

```

- 59. Write a R program to call the (built-in) dataset airquality. Check whether it is a data frame or not? Order the entire data frame by the first and second column.**

```

data = airquality
print("Original data: Daily air quality measurements in New York, May to September 1973.")
print(class(data))
print(head(data,10))
result = data[order(data[,1]),]
print("Order the entire data frame by the first and second column:")
print(result)

```

- 60. Write a R program to call the (built-in) dataset airquality. Remove the variables 'Solar.R' and 'Wind' and display the data frame.**

```

data = airquality
print("Original data: Daily air quality measurements in New York, May to September 1973.")
print(data)
data[,c("Solar.R")] = NULL
data[,c("Wind")] = NULL
print("data.frame after removing 'Solar.R' and 'Wind' variables:")
print(data)

```

MATRIX

- 61. Write a R program to create a blank matrix.**

```

m = matrix(), nrow = 10, ncol = 5)
print("Empty matrix of 10 rows and 5 columns:")

```



```
print(m)
```

- 62. Write a R program to create a matrix taking a given vector of numbers as input. Display the matrix.**

```
M = matrix(c(1:16), nrow = 4, byrow = TRUE)
print("Original Matrix:")
print(M)
```

- 63. Write a R program to create a matrix taking a given vector of numbers as input and define the column and row names. Display the matrix.**

```
row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names, col_names))
print("Original Matrix:")
print(M)
```

- 64. Write a R program to access the element at 3rd column and 2nd row, only the 3rd row and only the 4th column of a given matrix.**

```
row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names, col_names))
print("Original Matrix:")
print(M)
print("Access the element at 3rd column and 2nd row:")
print(M[2,3])
print("Access only the 3rd row:")
print(M[3,])
print("Access only the 4th column:")
print(M[,4])
```

- 65. Write a R program to create two 2x3 matrix and add, subtract, multiply and divide the matrices.**

```
m1 = matrix(c(1, 2, 3, 4, 5, 6), nrow = 2)
print("Matrix-1:")
print(m1)
m2 = matrix(c(0, 1, 2, 3, 0, 2), nrow = 2)
print("Matrix-2:")
print(m2)
result = m1 + m2
print("Result of addition")
print(result)
result = m1 - m2
print("Result of subtraction")
print(result)
result = m1 * m2
print("Result of multiplication")
print(result)
```

```

result = m1 / m2
print("Result of division:")
print(result)

```

66. 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)

```

67. Write a R program to extract the submatrix whose rows have column value > 7 from a given matrix.

```

row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names, col_names))
print("Original Matrix:")
print(M)
result = M[M[,3] > 7,]
print("New submatrix:")
print(result)

```

68. Write a R program to convert a matrix to a 1 dimensional array.

```

row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names, col_names))
print("Original Matrix:")
print(M)
result = as.vector(M)
print("1 dimensional array (column wise):")
print(result)
result = as.vector(t(M))
print("1 dimensional array (row wise):")
print(result)

```

69. Write a R program to create a correlation matrix from a dataframe of same data type.

```

d = data.frame(x1=rnorm(5),
               x2=rnorm(5),
               x3=rnorm(5))
print("Original dataframe:")
print(d)
result = cor(d)
print("Correlation matrix:")
print(result)

```

70. Write a R program to convert a given matrix to a list of column-vectors.

```
x = matrix(1:12, ncol=3)
print("Original matrix:")
print(x)
print("list from the said matrix:")
l = split(x, rep(1:ncol(x), each = nrow(x)))
print(l)
```

71. 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)
```

72. Write a R program to rotate a given matrix 90 degree clockwise rotation.

```
x = matrix(1:9, 3)
print("Original matrix:")
print(x)
rotate = t(apply(x, 2, rev))
print("Rotate the said matrix 90 degree clockwise:")
print(rotate)
```

73. 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)
```

VECTOR

74. Write a R program to create a vector of a specified type and length. Create vector of numeric, complex, logical and character types of length 6.

```

x = vector("numeric", 5)
print("Numeric Type:")
print(x)
c = vector("complex", 5)
print("Complex Type:")
print(c)
l = vector("logical", 5)
print("Logical Type:")
print(l)
chr = vector("character", 5)
print("Character Type:")
print(chr)

```

75. Write a R program to add 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("After adding two Vectors:")
z = x + y
print(z)

```

76. 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)

```

77. 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)

```

78. Write a R program to divide 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(" After dividing two Vectors:")
z = x / y
print(z)
```

79. Write a R program to find Sum, Mean and Product of a Vector.

```
x = c(10, 20, 30)
print("Sum:")
print(sum(x))
print("Mean:")
print(mean(x))
print("Product:")
print(prod(x))
```

80. 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))
```

81. Write a R program to find the minimum and the maximum of a Vector.

```
x = c(10, 20, 30, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Maximum value of the above Vector:")
print(max(x))
print("Minimum value of the above Vector:")
print(min(x))
```

82. Write a R program to sort a Vector in ascending and descending order.

```
x = c(10, 20, 30, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Sort in ascending order:")
print(sort(x))
print("Sort in descending order:")
print(sort(x, decreasing=TRUE))
```

83. Write a R program to test whether a given vector contains a specified element.

```
x = c(10, 20, 30, 25, 9, 26)
```

```

print("Original Vectors:")
print(x)
print("Test whether above vector contains 25:")
print(is.element(25, x))
print("Test whether above vector contains 56:")
print(is.element(56, x))

```

84. Write a R program to count the specific value in a given vector.

```

x = c(10, 20, 30, 20, 20, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Count specific value(20) in above vector:")
print(sum(x==20))

```

85. Write a R program to access the last value in a given vector.

```

x = c(10, 20, 30, 20, 20, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Access the last value of the said vector:")
print(tail(x, n=1))

```

86. Write a R program to find second highest value in a given vector.

```

x = c(10, 20, 30, 20, 20, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Find second highest value in a given vector:")
l = length(x)
print(sort(x, partial = l-1)[l-1])

```

87. Write a R program to find nth highest value in a given vector.

```

x = c(10, 20, 30, 20, 20, 25, 9, 26)
print("Original Vectors:")
print(x)
print("nth highest value in a given vector:")
print("n = 1")
n = 1
print(sort(x, TRUE)[n])
print("n = 2")
n = 2
print(sort(x, TRUE)[n])
print("n = 3")
n = 3
print(sort(x, TRUE)[n])
print("n = 4")
n = 4

```

```
print(sort(x, TRUE)[n])
```

88. Write a R program to find common elements from multiple vectors.

```
x = c(10, 20, 30, 20, 20, 25, 29, 26)
y = c(10, 50, 30, 20, 20, 35, 19, 56)
z = c(10, 40, 30, 20, 20, 25, 49, 26)
print("Original Vectors:")
print("x: ")
print(x)
print("y: ")
print(y)
print("z: ")
print(z)
print("Common elements from above vectors:")
result = intersect(intersect(x,y),z)
print(result)
```

89. Write a R program to convert given dataframe column(s) to a vector.

```
dfc1 = c(1, 2, 3, 4, 5)
dfc2 = c(6, 7, 8, 9, 10)
dfc3 = c(11, 12, 13, 14, 15)
dfc4 = c(16, 17, 18, 19, 20)
v <- data.frame(dfc1=1:5, dfc2=6:10, dfc3=11:15, dfc4=16:20)
print(v)
```

90. Write a R program to extract every nth element of a given vector.

```
v <- 1:100
print("Original vector:")
print(v)
print("After extracting every 5th element of the said vector:")
n <- v[seq(1, length(v), 5)]
print(n)
```

91. Write a R program to list the distinct values in a vector from a given vector.

```
v = c(10, 10, 10, 20, 30, 40, 40, 40, 50)
print("Original vector:")
print(v)
print("Distinct values of the said vector:")
print(unique(v))
```

92. Write a R program to find the elements of a given vector that are not in another given vector.

```
a = c(0, 10, 10, 10, 20, 30, 40, 40, 40, 50, 60)
b = c(10, 10, 20, 30, 40, 40, 50)
print("Original vector-1:")
```

```

print(a)
print("Original vector-2:")
print(b)
print("Elements of a that are not in b:")
result = setdiff(a, b)
print(result)

```

93. Write a R program to reverse the order of given vector.

```

v = c(0, 10, 10, 10, 20, 30, 40, 40, 40, 50, 60)
print("Original vector-1:")
print(v)
rv = rev(v)
print("The said vector in reverse order:")
print(rv)

```

94. Write a R program to concatenate a vector.

```

a = c("Python", "NumPy", "Pandas")
print(a)
x = paste(a, collapse = "")
print("Concatenation of the said string:")
print(x)

```

95. Write a R program to count number of values in a range in a given vector.

```

v = c(0, 10, 20, 30, 40, 50, 60, 70, 80, 90)
print("Original vector:")
print(v)
ctr = sum(v > 10 & v < 50)
print("Number of vector values between 10 and 50:")
print(ctr)

```

96. Write a R program to convert two columns of a data frame to a named vector.

```

df = data.frame(code = c("R", "G", "W", "B"),
                name = c("Red", "Green", "White", "Black")
                )
print("Original vector:")
print(df)
result = setNames(as.character(df$name), df$code)
print(result)

```

97. Write a R program to create a vector and find the length and the dimension of the vector.

```

v = c(1,3,5,7,9)
print("Original vectors:")
print(v)
print("Dimension of the vector:")
print(dim(v))

```



```
print("length of the vector:")
print(length(v))
```

98. Write a R program to combines two given vectors by columns, rows.

```
v1 = c(1,3,5,7,9)
v2 = c(2,4,6,8,10)
print("Original vectors:")
print(v1)
print(v2)
print("Combines the said two vectors by columns:")
result = cbind(v1,v2)
print(result)
print("Combines the said two vectors by rows:")
result = rbind(v1,v2)
print(result)
```

99. Write a R program to test whether the value of the element of a given vector greater than 10 or not. Return TRUE or FALSE.

```
v = c(15,26,9,7,10,0,9,15)
print("Original vector:")
print(v)
print("Test whether the value > 10 or not:")
print(v > 10)
```

100. Write a R program to add 3 to each element in a given vector. Print the original and new vector.

```
v = c(1, 2, NULL, 3, 4, NULL)
print("Original vector:")
print(v)
new_v = (v+3)[(!is.na(v)) & v > 0]
print("New vector:")
print(new_v)
```

101. Write a R program to create a vector using : operator and seq() function.

```
x = 1:15
print("New vector using : operator-")
print(x)
print("New vector using seq() function-")
print("Specify step size:")
y = seq(1, 3, by=0.3)
print(y)
print("Specify length of the vector:")
z = seq(1, 5, length.out = 6)
print(z)
```

LIST

102. Write a R program to create a list containing strings, numbers, vectors and a logical values.

```
list_data = list("Python", "PHP", c(5, 7, 9, 11), TRUE, 125.17, 75.83)
print("Data of the list:")
print(list_data)
```

103. Write a R program to list containing 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)
```

104. 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.

```
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])
```

105. Write a R program to create a list containing a vector, a matrix and a list and add element at the end 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)
```

106. Write a R program to select 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)

```

- 107. 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)

```

- 108. Write a R program to create a list containing a vector, a matrix and a list and update the last element.**

```

list_data <- list(c("Red","Green","Black"), matrix(c(1,3,5,7,9,11), nrow = 2),
list("Python"))
print("List:")
print(list_data)
print("Update the last element of the list:")
list_data[3] = "R programming"
print("New list:")
print(list_data)

```

- 109. 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:")
mlist = c(n1, c1)
print("New merged list:")
print(mlist)

```

- 110. Write a R program to convert a given list to vector.**

```

n1 = list(1,2,3)
c1 = list(4,5,6)
print("Original lists:")
print(n1)
print(c1)

```

```

print("Convert the lists to vectors:")
v1 = unlist(n1)
v2 = unlist(c1)
print(v1)
print(v2)
print("Add two vectors:")
v = v1 + v2
print("New vector:")
print(v)

```

111. Write a R program to create a list of dataframes and access each of those data frames from the list.

```

df1 = data.frame(y1 = c(0, 1, 2), y2 = c(3, 4, 5))
df2 = data.frame(y1 = c(6, 7, 8), y2 = c(9, 10, 11))
new_list = list(df1, df2)
print("New list:")
print(new_list)
print("Data frame-1")
print(new_list[[1]])
print("Data frame-2")
print(new_list[[2]])

```

112. Write a R program to count number of objects in a given 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("Number of objects in the said list:")
length(list_data)

```

113. Write a R program to convert a given dataframe to a list by rows.

```

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)
new_list = split(exam_data, seq(nrow(exam_data)))
print("dataframe rows to a list:")
print(new_list)

```

114. Write a R program to convert a given matrix to a list.

```

m = matrix(1:10,nrow=2, ncol=2)
print("Original matrix:")
print(m)

```

```
l = split(m, rep(1:ncol(m), each = nrow(m)))
print("list from the said matrix:")
print(l)
```

115. Write a R program to assign NULL to a given list element.

```
l = list(1, 2, 3, 4, 5)
print("Original list:")
print(l)
print("Set 2nd and 3rd elements to NULL")
l[2] = list(NULL)
l[3] = list(NULL)
print(l)
```

116. Write a R program to create a list named s 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)
```

117. Write a R program to Add 10 to each element of the first vector in a given list.

```
list1 = list(g1 = 1:10, g2 = "R Programming", g3 = "HTML")
print("Original list:")
print(list1)
print("New list:")
list1$g1 = list1$g1 + 10
print(list1$g1)
```

118. Write a R program to extract all elements except the third element of the first vector of a given list.

```
list1 = list(g1 = 1:10, g2 = "R Programming", g3 = "HTML")
print("Original list:")
print(list1)
print("First vector:")
print(list1$g1)
print("First vector without third element:")
list1$g1 = list1$g1[-3]
print(list1$g1)
```

119. Write a R program to add a new item g4 = "Python" to a given list.

```
list1 = list(g1 = 1:10, g2 = "R Programming", g3 = "HTML")
print("Original list:")
print(list1)
print("Add a new vector to the said list:")
list1$g4 = "Python"
print(list1)
```

120. Write a R program to assign new names "one", "two" and "three" 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)
```

121. Write a R program to get the length of the first two vectors of a given list.

```
list1 = list(g1 = 1:10, g2 = "R Programming", g3 = "HTML")
print("Original list:")
print(list1)
print("Length of the vector 'g1' and 'g2' of the said list")
print(length(list1$g1))
print(length(list1$g2))
```

122. Write a R program to find all elements of a given list that are not in another given list.

```
l1 = list("x", "y", "z")
l2 = list("X", "Y", "Z", "x", "y", "z")
print("Original lists:")
print(l1)
print(l2)
print("All elements of l2 that are not in l1:")
setdiff(l2, l1)
```

FACTORS

123. Write a R program to find the levels of 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)))
```

124. Write a R program to change the first level of a factor with another level of a given factor.

```
v = c("a", "b", "a", "c", "b")
print("Original vector:")
print(v)
f = factor(v)
print("Factor of the said vector:")
print(f)
levels(f)[1] = "e"
print(f)
```

125. 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))
```

126. Write a R program to concatenate two given factor in 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)
```

127. Write a R program to convert a given pH levels of soil to an ordered factor.

```
ph = c(1,3,10,7,5,4,3,7,8,7,5,3,10,10,7)  
print("Original data:")  
print(ph)  
ph_f = factor(ph, levels=c(3,7,10), ordered=TRUE)  
print("pH levels of soil to an ordered factor:")  
print(ph_f)
```

128. Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)

```
L = sample(LETTERS, size=50, replace=TRUE)  
print("Original data:")  
print(L)  
f = factor(L)  
print("Original factors:")  
print(f)  
print("Only five of the levels")  
print(table(L[1:5]))
```

129. Write a R program to create a factor corresponding to height of women data set, which contains height and weights for a sample of women.

```
data = women
print("Women data set of height and weights:")
print(data)
height_f = cut(women$height,3)
print("Factor corresponding to height:")
print(table(height_f))
```