R Programming

|  |
| --- |
| **Final Assignment** |

Exercise – 1

Create a numeric vector and check whether the numbers stored in it are even numbers or odd numbers by using the ifelse () function.

Exercise – 2

Write a program to display the table of 5 using the for loop.

Exercise – 3

Write a program that lists the square of all the numbers between 11 and 20 using the while loop.

Exercise – 4

Write a program that displays the sum of the first 10 natural numbers using the repeat loop.

Exercise – 5

Write a program to display the following output:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

1 2 3 4 5 6 7

Exercise – 6

Write a program to display the following output:

9

9 8

9 8 7

9 8 7 6

9 8 7 6 5

9 8 7 6 5 4

9 8 7 6 5 4 3

9 8 7 6 5 4 3 2

9 8 7 6 5 4 3 2 1

Exercise – 7

Write a program to display the reverse of all the numbers between 301 and 309.

Exercise – 8

Write a program to display all the divisors of a given numbers.

Exercises – 1

Execute the commands shown in the following table and observe the output.

|  |  |
| --- | --- |
| **Command** | **Output** |
| c(T, T) & C(T, F) |  |
| c(T, T) | c (T, F) |  |
| c(T, T) && c(F, T) |  |
| c(T, T) || c(F, T) |  |
| !c(T, F) |  |
| c(1, 2) %in% c(1, 3, 4, 5) |  |

Exercise – 2

Execute the math function shown in the following table and observe the output.

|  |  |
| --- | --- |
| **Math Function** | **Output** |
| sqrt (2) |  |
| sqrt (-2) |  |
| log (0) |  |
| log (1) |  |
| ceiling (10 . 6) |  |
| floor (9.5) |  |
| floor (9.5) |  |
| round (pi, 3) |  |
| signif (pi, 3 |  |
| is. finite(1 / 0) |  |

Exercise – 3

Execute the following code and observe the output.

max(c(1, 2, 3),

c(2, 1, 2),

c(1, 3, 4))

min(c(1, 2, 3),

c(2, 1, 2),

c(1, 3, 4))

Exercise – 4

Write a code that draws a sample of five form a numeric vector without replacement.

Exercise – 5

Write a code that draws a sample of five from a numeric vector with replacement.

Exercise – 6

Generate a random numeric vector of length 50 from standard normal distribution and perform the following tasks:

• Compute the arithmetic sample mean.

• Compute the arithmetic sample median.

• Compute the standard deviation.

• Compute the variance.

• Compute the summary statistics, including four quantile, the median, and the mean.

Exercise – 1

Execute the commands shown in the following table and observe the output.

|  |  |
| --- | --- |
| **Commands** | **Output** |
| sprintf (“%04d” , 10) |  |
| sprintf ( “%f” , pi) |  |
| sprintf (“%.2f” , pi) |  |
| sprintf (“%08.2f” , pi) |  |
| strsplit(c(“hello”, “world”) , splite = “”) |  |
| substr(dates, 1, 3) |  |
| nchar(“Hello”) |  |
| Sys.time()  Sys.Date() |  |

Exercise – 2

Write a function that accepts a string with name type and two numbers as its parameters. When the value of type is add, the function returns the addition of the two numbers . When the value of type is times , it returns the product of the two numbers. Remember that the value of type can be passed either in lowercase or in uppercase.

Exercise – 1

Read a text file named persons.csv and display the number of lines as a character vector.

Exercise – 2

Create an arbitrary data frame and store it in a csv file.

Exercise – 3

Create a normally distributed random numeric vector. Then, create a histogram by using the vector.

Exercise – 4

Create a pie chart by using a numeric vector, which stores the value of different grades.

Additional Activity:

1. Create a vector numbers from 1 to 6 and find out its class
2. Pick 50 random numbers between 1 to 100, with replacement and check the class.
3. Get the last 2 rows in last 2 columns from iris dataset
4. Create following 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 "The number of NA values in attempts column:"

1. Consider the following vector:

addresses <- c("14 Pine Street, Los Angeles", "152 Redwood Street, Seattle", "8 Washington Boulevard, New York")

1. Pull only the numeric part of the addresses vector.
2. Split the addresses vector into two parts: address and city. The result should be a matrix.