192021045L

M.S.KIRUBA

R PROGRAMMING Day1 lab

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(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)

ANS:

Input your age: 33

[1] "My name is and I am 33 years old."

[1] "R version 4.2.2 (2022-10-31 ucrt)"

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

program:

name = "KIRUBA";

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())

ANS:

Input your name: KIRUBA

Input your age: 21

[1] "My name is KIRUBA and I am 21 years old."

[1] "R version 4.2.2 (2022-10-31 ucrt)"

[1] "age" "n1" "n2" "name" "nums"

[1] "Details of the objects in memory:"

age : chr "21"

n1 : num 10

n2 : num 0.5

name : chr "KIRUBA"

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.

PROGRAM:

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))

ANS:

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

[1] "Sequence of numbers from 20 to 50:"

> print(seq(20,50))

[1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

[24] 43 44 45 46 47 48 49 50

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

[1] "Mean of numbers from 20 to 60:"

> print(mean(20:60))

[1] 40

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

[1] "Sum of numbers from 51 to 91:"

> print(sum(51:91))

[1] 2911

4. Write a R program to create a vector which contains 10 random integer values

between -50 and +50.

PROGRAM:

v = sample(-50:50, 10, replace=TRUE)

print("Content of the vector:")

print("10 random integer values between -50 and +50:")

print(v)

ANS:

Browse[2]> v = sample(-50:50, 10, replace=TRUE)

Browse[2]> print("Content of the vector:")

[1] "Content of the vector:"

Browse[2]> print("10 random integer values between -50 and +50:")

[1] "10 random integer values between -50 and +50:"

Browse[2]> print(v)

[1] -22 10 -11 -21 -43 -49 -28 21 -12 8

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

PROGRAM:

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)

ANS:

[1] "First 10 Fibonacci numbers:"

[1] 1 1 2 3 5 8 13 21 34 55

6. Write a R program to get all prime numbers up to a given number (based on

the sieve of Eratosthenes).

PROGRAM:

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 = c(x[(x %% i) != 0], i)

}

}

return(prime\_nums)

}

else

{

stop("Input number should be at least 2.")

}

}

prime\_numbers(12)

ans:

[1] 2 3 5 7 11

7. Write a R program to print the numbers from 1 to 100 and print &quot;Fizz&quot; for

multiples of 3, print &quot;Buzz&quot; for multiples of 5, and print &quot;FizzBuzz&quot; for multiples of

both.

PROGRAM:

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)

}

ANS:

> 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)

+ }

[1] 1

[1] 2

[1] "Fizz"

[1] 4

[1] "Buzz"

[1] "Fizz"

[1] 7

[1] 8

[1] "Fizz"

[1] "Buzz"

[1] 11

[1] "Fizz"

[1] 13

[1] 14

[1] "FizzBuzz"

[1] 16

[1] 17

[1] "Fizz"

[1] 19

[1] "Buzz"

[1] "Fizz"

[1] 22

[1] 23

[1] "Fizz"

[1] "Buzz"

[1] 26

[1] "Fizz"

[1] 28

[1] 29

[1] "FizzBuzz"

[1] 31

[1] 32

[1] "Fizz"

[1] 34

[1] "Buzz"

[1] "Fizz"

[1] 37

[1] 38

[1] "Fizz"

[1] "Buzz"

[1] 41

[1] "Fizz"

[1] 43

[1] 44

[1] "FizzBuzz"

[1] 46

[1] 47

[1] "Fizz"

[1] 49

[1] "Buzz"

[1] "Fizz"

[1] 52

[1] 53

[1] "Fizz"

[1] "Buzz"

[1] 56

[1] "Fizz"

[1] 58

[1] 59

[1] "FizzBuzz"

[1] 61

[1] 62

[1] "Fizz"

[1] 64

[1] "Buzz"

[1] "Fizz"

[1] 67

[1] 68

[1] "Fizz"

[1] "Buzz"

[1] 71

[1] "Fizz"

[1] 73

[1] 74

[1] "FizzBuzz"

[1] 76

[1] 77

[1] "Fizz"

[1] 79

[1] "Buzz"

[1] "Fizz"

[1] 82

[1] 83

[1] "Fizz"

[1] "Buzz"

[1] 86

[1] "Fizz"

[1] 88

[1] 89

[1] "FizzBuzz"

[1] 91

[1] 92

[1] "Fizz"

[1] 94

[1] "Buzz"

[1] "Fizz"

[1] 97

[1] 98

[1] "Fizz"

[1] "Buzz"

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 22 nd  to 24 th  letters in upper case.

PROGRAM:

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)

ANS:

> print("First 10 letters in lower case:")

[1] "First 10 letters in lower case:"

> t = head(letters, 10)

> print(t)

[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"

> print("Last 10 letters in upper case:")

[1] "Last 10 letters in upper case:"

> t = tail(LETTERS, 10)

> print(t)

[1] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

> print("Letters between 22nd to 24th letters in upper case:")

[1] "Letters between 22nd to 24th letters in upper case:"

> e = tail(LETTERS[22:24])

> print(e)

[1] "V" "W" "X"

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

PROGRAM:

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)

ANS:

> 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)

[1] "The factors of 4 are:"

[1] 1

[1] 2

[1] 4

> print\_factors(7)

[1] "The factors of 7 are:"

[1] 1

[1] 7

> print\_factors(12)

[1] "The factors of 12 are:"

[1] 1

[1] 2

[1] 3

[1] 4

[1] 6

[1] 12

>

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

vector.

PROGRAM:

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))

ANS:

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

> print("Original Vectors:")

[1] "Original Vectors:"

> print(x)

[1] 10 20 30 25 9 26

> print("Maximum value of the above Vector:")

[1] "Maximum value of the above Vector:"

> print(max(x))

[1] 30

> print("Minimum value of the above Vector:")

[1] "Minimum value of the above Vector:"

> print(min(x))

[1] 9

11. Write a R program to get the unique elements of a given string and unique

numbers of vector.

PROGRAM:

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))

ANS:

> str1 = "The quick brown fox jumps over the lazy dog."

> print("Original vector(string)")

[1] "Original vector(string)"

> print(str1)

[1] "The quick brown fox jumps over the lazy dog."

> print("Unique elements of the said vector:")

[1] "Unique elements of the said vector:"

> print(unique(tolower(str1)))

[1] "the quick brown fox jumps over the lazy dog."

> nums = c(1, 2, 2, 3, 4, 4, 5, 6)

> print("Original vector(number)")

[1] "Original vector(number)"

> print(nums)

[1] 1 2 2 3 4 4 5 6

> print("Unique elements of the said vector:")

[1] "Unique elements of the said vector:"

> print(unique(nums))

[1] 1 2 3 4 5 6

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.

PROGRAM:

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)

ANS:

> 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:")

[1] "Content of the said matrix:"

> print(m)

a b c

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

13. Write a R program to create a list of random numbers in normal distribution

and count occurrences of each value.

PROGRAM:

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)

ANS:

> n = floor(rnorm(100, 10, 50))

> print('List of random numbers in normal distribution:')

[1] "List of random numbers in normal distribution:"

> print(n)

[1] 40 106 -43 -13 68 -58 1 6 -35 90 -35 43 42 -32 34 -43 34

[18] -6 -55 -30 18 -49 -13 88 46 9 101 29 66 26 52 39 -37 32

[35] -12 -55 50 -12 -6 59 -27 -17 14 -2 84 82 -11 -9 10 -60 39

[52] 0 -51 1 15 -23 3 -16 28 9 87 32 56 20 37 -4 76 35

[69] -1 26 59 -61 13 -34 -10 19 36 94 42 -32 68 -36 17 -40 72

[86] -50 114 -9 2 32 64 17 -61 84 1 -32 -88 -35 21 34

> t = table(n)

> print("Count occurrences of each value:")

[1] "Count occurrences of each value:"

> print(t)

n

-88 -61 -60 -58 -55 -51 -50 -49 -43 -40 -37 -36 -35 -34 -32 -30 -27 -23

1 2 1 1 2 1 1 1 2 1 1 1 3 1 3 1 1 1

-17 -16 -13 -12 -11 -10 -9 -6 -4 -2 -1 0 1 2 3 6 9 10

1 1 2 2 1 1 2 2 1 1 1 1 3 1 1 1 2 1

13 14 15 17 18 19 20 21 26 28 29 32 34 35 36 37 39 40

1 1 1 2 1 1 1 1 2 1 1 3 3 1 1 1 2 1

42 43 46 50 52 56 59 64 66 68 72 76 82 84 87 88 90 94

2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1

101 106 114

1 1 1

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

PROGRAM:

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.

PROGRAM:

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))

ANS:

> 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)

[1] 1 2 5 3 4 0 -1 -3

> print(typeof(a))

[1] "double"

> print(b)

[1] "Red" "Green" "White"

> print(typeof(b))

[1] "character"

> print(c)

[1] TRUE TRUE TRUE FALSE TRUE FALSE

> print(typeof(c))

[1] "logical"

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

PROGRAM:

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)

ANS:

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

> print("5 × 4 matrix:")

[1] "5 × 4 matrix:"

> print(m1)

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

[1,] 1 6 11 16

[2,] 2 7 12 17

[3,] 3 8 13 18

[4,] 4 9 14 19

[5,] 5 10 15 20

> 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: ")

[1] "3 × 3 matrix with labels, filled by rows: "

> print(m2)

Col1 Col2 Col3

Row1 1 3 5

Row2 7 8 9

Row3 11 12 14

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

[1] "3 × 3 matrix with labels, filled by columns: "

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

> print(m3)

Col1 Col2 Col3

Row1 1 7 11

Row2 3 8 12

Row3 5 9 14

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.

PROGRAM:

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)

ANS:

> 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)

, , Part1

Row1 Row2 Row3

Col1 6 10 14

Col2 7 11 15

Col3 8 12 16

Col4 9 13 17

, , Part2

Row1 Row2 Row3

Col1 18 22 26

Col2 19 23 27

Col3 20 24 28

Col4 21 25 29

18. Write a R program to create an array with three columns, three rows, and two

&quot;tables&quot;, taking two  vectors as input to the array.  Print the array.

PROGRAM:

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)

ANS:

> 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)

, , 1

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

[1,] 1 7 6

[2,] 3 2 8

[3,] 5 4 10

, , 2

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

[1,] 1 7 6

[2,] 3 2 8

[3,] 5 4 10

19. Write a R program to create a list of elements using vectors, matrices and a

functions. Print the content of the list.

PROGRAM:

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)

ANS:

> 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:")

[1] "Content of the list:"

> print(l)

[[1]]

[1] 1 2 2 5 7 12

[[2]]

[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov"

[12] "Dec"

[[3]]

[,1] [,2]

[1,] 3 1

[2,] -8 -3

[[4]]

function (x) .Primitive("asin")

20. Write a R program to draw an empty plot and an empty plot specify the axes

limits of the graphic

PROGRAM:

#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))

ANS: