

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(R.version.string)
```

OUTPUT:

Input your name: sudhar

Input your age: 29

```
[1] "R version 4.2.3 (2023-03-15 ucrt)"
```

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

PROGRAM:

```
name = "REA";
> n1 = 20;
> n2 = 0.10
> nums = c(10, 15, 20, 25, 30, 35)
> print(ls())
> print("Details of the objects in memory:")
> print(ls.str())
```

OUTPUT:

```
[1] "age" "c" "chr" "f" "l" "list1" "mons_v"
[8] "n1" "n2" "name" "nums" "x"
```

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

```
age : chr "29"
c : cplx [1:5] 0+0i 0+0i 0+0i ...
chr : chr [1:5] "" "" "" "" ""
f : Factor w/ 11 levels "April","August",...: 7 1 5 9 5 11 10 11 9 2
...
l : logi [1:5] FALSE FALSE FALSE FALSE FALSE
list1 : List of 4
 $ g1: int [1:10] 1 2 3 4 5 6 7 8 9 10
 $ g2: chr "R Programming"
 $ g3: chr "HTML"
 $ g4: chr "Python"
mons_v : chr [1:27] "March" "April" "January" "November"
"January" ...
n1 : num 20
n2 : num 0.1
name : chr "REA"
nums : num [1:6] 10 15 20 25 30 35
x : num [1:3] 10 20 30

>
```

3. Write a R program to create a sequence of numbers from 30 to 50 and find the mean of numbers from 30 to 60 and sum of numbers from 41 to 91.

PROGRAM&*OUTPUTS

```
print("Sequence of numbers from 30 to 50:")
```

```
[1] "Sequence of numbers from 30 to 50:"
```

```
> print(seq(30,50))
```

```
[1] 30 31 32 33 34 35 36 37 38 39 40 41 42
```

```
[14] 43 44 45 46 47 48 49 50
```

```
> print("Mean of numbers from 30 to 60:")
```

```
[1] "Mean of numbers from 30 to 60:"
```

```
> print(mean(30:60))
```

```
[1] 45
```

```
> print("Sum of numbers from 41 to 91:")
```

```
[1] "Sum of numbers from 41 to 91:"
```

```
> print(sum(41:91))
```

```
[1] 3366
```

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

between -50 and +50.

PROGRAM & OUTPUTS:

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

```
> print("Content of the vector:")
```

```
[1] "Content of the vector:"
```

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

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

```
> print(v)
```

```
[1] 23 -42 35 -46 14 -17 -41 -50 31 -22
```

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

Program&outputs

```
> (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 %% i !=0  
,i)} } return(prime_nums)}else{stop("input number should be  
atleast 2")}}prime_numbers(12)
```

```
[1] 2 3 5 7 11
```

6. 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&outputs:

```
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:")
"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)
[1] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
[1] "Letters between 22nd to 24th letters in upper case:"
[1] "V" "W" "X"
```

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

Program&outputs

```
nums = c(10, 20, 30, 40, 50, 60)
print('Original vector:')
[1] "Original vector:"
print(nums) [1] 10 20 30 40 50 60
print(paste("Maximum value of the said vector:",max(nums)))
[1] "Maximum value of the said vector: 60"
print(paste("Minimum value of the said vector:",min(nums)))
[1] "Minimum value of the said vector: 10"
```

8. 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)")
[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
```

9. Write a R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3x3 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:")
[1] "Content of the said matrix:"
print(m)
```

```
a b c
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

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

```
n = floor(rnorm(40, 30, 50))
> print('List of random numbers in normal distribution:')
[1] "List of random numbers in normal distribution:"

> print(n)
[1] 26 124 -11 53 -56 -2 52 10 21 35 13 88 80 23 53 -74 39
[18] 70 133 -81 -64 82 83 83 -18 53 12 -7 50 72 17 21 -37 -5
[35] -39 0 85 80 78 104

> t = table(n)
> print("Count occurrences of each value:")
[1] "Count occurrences of each value:"

> print(t)
n
-81 -74 -64 -56 -39 -37 -18 -11 -7 -5 -2 0 10 12 13 17 21 23
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
26 35 39 50 52 53 70 72 78 80 82 83 85 88 104 124 133
1 1 1 1 1 3 1 1 1 2 1 2 1 1 1 1 1
```

11. 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)
```

```
[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))
```

```
"logical"
```

12. 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 x 2 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
) [1] "5 × 4 matrix:"
[,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
[1] "3 × 3 matrix with labels, filled by rows: "
Col1 Col2 Col3
Row1 1 3 5
Row2 7 8 9
Row3 11 12 14
[1] "3 × 3 matrix with labels, filled by columns: "
Col1 Col2 Col3
Row1 1 7 11
Row2 3 8 12
Row3 5 9 14

```

13. 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)
, , 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 2

14. 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(arr1)
```

```
, , 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
```

15. 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)  
[1] "Content of the list:"  
[[1]]  
[1] 1 2 2 5 7 12  
[[2]] [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"  
[[3]] [,1] [,2]  
[1,] 3 1  
[2,] -8 -3  
[[4]]  
function (x) .Primitive("asin")
```