

19
20

81

Reg No: 192125001

Name: R. Alshwarya

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.

Coding:

```
name<-readline(prompt="Enter your name:")
age<-readline(prompt="Enter your age:")
cat("Your name is:", name, "\n")
cat("Your age is:", age, "\n")
Print R installation version.
cat("R version :", R.version.string,
    "\n")
```

Output:

Enter your name: Pojetha.

Enter your age: 22

Your name is: Pojetha

Your age is: 22.

R Version: R version 4.1.0 (2021-05-18)

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

Coding:

```
x<-1:100
y<-"hello"
z<-matrix(1:9, nrow=3)
object_size<-apply(lsc, function(x) object_size
                    (get(x)))
object_size_mb<-object_size/1024^2
object_data<-data.frame(object=names(object_
                           sizes), size_MB=object_size_mb)
print(object_data).
```

Output:

	object	size_MB
1,	object_data	0.0000543
2,	x	0.0000019
3,	y	0.0000003
4,	z	0.00000215

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.

Coding:

```
my_seq <- seq(20, 50)
mean_20_to_60 <- mean(seq(20, 60))
sum_51_to_91 <- sum(seq(51, 91))
cat("Sequence of numbers from 20 to 50:\n",
    my_seq, "\n")
cat("Mean of numbers from 20 to 60:",
    mean_20_to_60, "\n")
cat("Sum of numbers from 51 to 91:",
    sum_51_to_91, "\n")
```

Output:

Sequence of numbers
 20 21 22 23 24 25 26 27
 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43
 44 45 46 47 48 49 50

Mean of numbers from
 Sum of numbers from

4. Write a R program to create a vector which contains 10 random integer values between -50 and +50.

Coding:

```
set.seed(123)
my_vector <- sample(-50:50, 10, replace=
    TRUE)
print(my_vector)
```

Output:

-20 28 0 -37 16 -9 -1 -8 50 -37.

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

```

fibonacci <- function(n)
{
  fcb <- c(0,1)
  for(i in 3:n)
  {
    fcb[i] <- fcb[i-1] + fcb[i-2]
  }
  return(fcb[1:n])
}

```

Output:

0 1 1 2 3 5 8 13 21 34

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

Code:

```

sieve_of_eratosthenes <- function(n)
{
  is_prime <- rep(TRUE, n)
  is_prime[1:2] <- FALSE
  for(p in 2:sqrt(n))
  {
    if(is_prime[p])
    {
      is_prime[p^2:n:n] <- FALSE
    }
  }
  return(which(is_prime))
}

```

O/p:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71
73 79 83 89 97

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

```

Output:

```

1
2
"Fizz"
4
"Buzz"
"Fizz"
7
8
"Fizz"
"Buzz"
11
"Fizz"
13
14
"Fizz Buzz"
16
17
"Fizz"
19
"Buzz"
"Fizz"

```

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.

code:

```

english_letters<-"abcdefghijklmnopqrstuvwxyz"
first_10_letters<-substr(english_letters, start=1, stop=10)
last_10_letters<-toupper(substr(english_letters, start=17))
mid_letters<-toupper(substr(english_letters, start=22, stop=24))
cat("First 10 letters in lowercase:", first_10_letters, "\n")
cat("Last 10 letters in uppercase:", last_10_letters, "\n")
cat("Letters between 22nd to 24th in uppercase:", mid_letters, "\n")

```

output:

First 10 letters in lowercase: abcdefghij

Last 10 letters in uppercase: ARSTUVWXYZ.

Letters between 22nd to 24th in uppercase: WXY

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


```

num <- 36
factors <- c()
for (i in 1:num)
{
  if (num % i == 0)
  {
    factors <- c(factors, i)
  }
}
cat("The factors of", num, "are:", factors, "\n")

```

Output

The factors of 36 are: 1 2 3 4 6 9 12 18 36.

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

code:

```

vec <- c(3, 5, -2, 7, 10, -4)
max_val <- max(vec)
min_val <- min(vec)
cat("The minimum value of the vector is:", min_val, "\n")
cat("The maximum value of the vector is:", max_val, "\n")

```

Output:

The maximum value of the vector is 10.

The minimum value of the vector is -4.

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

code:

```
string<-"hello world"
```

```
vec<-c(2,4,6,2,8,4,10)
```

```
unique_string<-unique(strsplit(string," ")[[1]])
```

```
unique_vec<-unique(vec)
```

```
cat("Unique elements of the string are:", unique_string, "\n")
```

```
cat("Unique elements of the vector are:", unique_vec, "\n")
```

Output:

```
unique elements of the string are: h e l l o w o r d
```

```
unique elements of the vector are: 2 4 6 8 10
```

12. 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.

code:

```
a<-c(1,2,3)
```

```
b<-c(4,5,6)
```

```
c<-c(7,8,9)
```

```
matrix<-cbind(a,b,c)
```

```
print(matrix)
```

Output:

1	4	7
2	5	8
3	6	9

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

code:

```

set.seed(123)
nums <- rnorm(100, mean = 0, sd = 1)
counts <- table(nums)
print(counts)
output:
-2.30916 887564081 -1.96661715662964 -1.68669331074241
-1.54875280423022 -1.26539635156826 -1.26506123460653
-1.22071771225454 -1.1313693701195 -1.12310858320335

```

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

```

code:
numeric_vector <- c(1, 2, 3, 4, 5)
character_vector <- c("apple", "banana", "cherry")
logical_vector <- c(TRUE, FALSE, TRUE, TRUE, FALSE)
print(numeric_vector)
print(character_vector)
print(logical_vector)
print(typeof(numeric_vector))
print(typeof(character_vector))
print(typeof(logical_vector))

```

output:

```

12345
"apple" "banana" "cherry"
TRUE FALSE TRUE TRUE FALSE
"double"
"character"
"logical"

```

15. 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.

```

matrix_rows <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25),
  nrow = 5, ncol = 4, byrow = TRUE)
print(matrix_rows)
matrix_labels_rows <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow = 3, ncol = 3)
print(matrix_labels_rows)

```



```
matrix_labels_cols <- matrix (c(1,2,3,4), nrow=2, ncol=2, byrow=FALSE,
                                list(c("Row1", "Row2"), c("col1", "col2")))
print(matrix_labels_cols)
```

output:

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
21 22 23 24
```

```
Row1 col1 col2 col3
      1    2    3
Row2    4    5    6
Row3    7    8    9
```

```
col1 col2
Row1  1    3
Row2  2    4
```

16. 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.

code:

```
values <- c(1,2,3,4,5,6)
```

```
dims <- c(2,3)
```

```
dimnames <- list(c("Row1", "Row2"), c("col1", "col2", "col3"))
```

```
my_array <- array(data=values, dim=dims, dimnames=dimnames)
```

```
print(my_array)
```

output:

```
Row1 col1 col2 col3
      1    3    5
Row2    2    4    6
```

17. 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.

code:

```
vec1 <- c(1,2,3,4,5,6,7,8,9)
```

```
vec2 <- c(10,11,12,13,14,15,16,17,18)
```

```
combined_vec <- c(vec1, vec2)
```

```
my_array <- array(combined_vec, dim=c(3,3,2))
```

```
print(my_array)
```


10	13	16
11	14	17
12	15	18.

18. Write a R program to create a list of elements using vectors, matrices and a function. Print the content of the list.

Code:

```
vec <- c(1,2,3,4)
mat <- matrix(1:6, nrow=2)
func <- function(x)
{
  x^2
  3
}
my_list <- list(vec, mat, func)
print(my_list)
```

Output:

1	2	3	4
[1,]	[1]	[, 2]	[, 3]
	1	3	5
[2,]	2	4	6.

```
function(x)
{
  x^2
  3
}
```

DAY 2 – LAB ASSESSMENT

Reg No:

Name:

1. 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.

Code:

```
mat1 <- matrix(1:9, nrow=3, byrow=TRUE)
mat2 <- matrix(10:18, nrow=3, byrow=TRUE)
```

19
20