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.

Ans

```
name <- readline(prompt = "Enter your name: ")
age <- readline(prompt = "Enter your age: ")
cat("Your name is ", name, " and your age is ", age, "\n")
cat("R version ", paste(R.version$major, R.version$minor, R.version$patch, sep = "."), "\n")
```

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

Ans

```
name <- readline(prompt = "Enter your name: ")
age <- readline(prompt = "Enter your age: ")
cat("Your name is ", name, " and your age is ", age, "\n")
cat("R version ", paste(R.version$major, R.version$minor, R.version$patch, sep = "."), "\n")
```

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.

Ans

```
x <- 20:50
y <- 20:60
mean_y <- mean(y)
z <- 51:91
sum_z <- sum(z)
cat("Sequence of numbers from 20 to 50: ", x, "\n")
cat("Mean of numbers from 20 to 60: ", mean_y, "\n")
cat("Sum of numbers from 51 to 91: ", sum_z, "\n")
```

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

and +50.

Ans

```
set.seed(123)
x <- sample(-50:50, 10, replace = TRUE)
cat("Random vector of 10 integers between -50 and +50: ", x, "\n")
```

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

Ans

```
set.seed(123)
random_vector <- sample.int(-50:50, 10, replace = TRUE)
cat("Random vector of 10 integers between -50 and +50: ", random_vector, "\n")
```

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

Ans

```
fibonacci <- function(n) {
  if(n == 1 || n == 2) {
    return(1)
  } else {
    return(fibonacci(n-1) + fibonacci(n-2))
  }
}
fibonacci_numbers <- sapply(1:10, fibonacci)
cat("First 10 Fibonacci numbers: ", fibonacci_numbers, "\n")
```

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

Ans

```
get_primes <- function(n) {
  is_prime <- rep(TRUE, n)
   for(i in 2:sqrt(n)) {
   # If i is prime, mark all its multiples as not prime
   if(is_prime[i]) {
     is_prime[i^2:n:n] <- FALSE
   }
 }
 return(which(is_prime))
}
primes_up_to_50 <- get_primes(50)
cat("All prime numbers up to 50: ", primes_up_to_50, "\n")
```

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.

Ans

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

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.

Ans

```
lowercase_letters <- letters[1:10]
uppercase_letters <- toupper(letters[17:26])
mid_letters <- toupper(letters[22:24])
cat("Lowercase letters:", lowercase_letters, "\n")
cat("Uppercase letters:", uppercase_letters, "\n")
cat("Mid letters:", mid_letters, "\n")
```

Output

```
Lowercase letters: a b c d e f g h i j
Uppercase letters: Q R S T U V W X Y Z
Mid letters: V W X
```

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

Ans

```
num <- as.integer(readline(prompt = "Enter a number: "))
factors <- c()
for (i in 1:num) {
  if (num %% i == 0) {
    factors <- c(factors, i)
  }
}
cat("The factors of", num, "are:", factors, "\n")
```

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

Ans

```
numbers <- c(10, 25, 7, 42, 18, 3)
max_value <- max(numbers)
min_value <- min(numbers)
cat("The maximum value is:", max_value, "\n")
cat("The minimum value is:", min_value, "\n")
```

OUTPUT

The maximum value is: 42

The minimum value is: 3

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

Ans

```
my_string <- "Hello, World!"
my_vector <- c(1, 2, 3, 1, 2, 4)
unique_string <- unique(strsplit(my_string, "")[[1]])
cat("Unique elements of the string:", paste(unique_string, collapse = ", "), "\n")
unique_vector <- unique(my_vector)
cat("Unique numbers of the vector:", paste(unique_vector, collapse = ", "))
```

Output

Unique elements of the string: H, e, l, o, ,, W, r, d, !

Unique numbers of the vector: 1, 2, 3, 4

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.

Ans

```
a <- c(1, 2, 3)
b <- c(4, 5, 6)
c <- c(7, 8, 9)
m <- cbind(a, b, c)
print(m)
```

Output

```
     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.

Ans

```
my_list <- rnorm(100)
my_table <- table(my_list)
print(my_list)
print(my_table)
```

Output

```
[1] -1.1248342  0.6981634 -0.0075391  0.8446679  0.4394216 -0.4106110  0.2373448 -1.4171465 -1.0805799
[10]  0.5057087 -0.5960223 -1.2272003 -0.2534275 -1.0821665  0.7716267  0.1323315 -1.0694225 -1.4150202
```

[19] 0.0832507 -0.1773326 0.4813828 -1.2676584 -0.3401689 -0.3811318 1.0963254 0.1523044 0.8021703

[28] 1.2355503 -0.1728221 -1.1781577 -0.2982304 -0.5586094 -0.3424878 -1.2286771 -0.0092594 1.6740057

[37] 0.5017013 0.3176793 1.4901154 1.4142274 0.4137447 1.1983469 1.4401385 -1.6725823 -0.0509813

[46] -0.3783245 -1.0641565 -1.2047764 0.2838914 0.6756462 -0.8265569 -0.1972279 -0.3853107 0.7293718

[55] -1.1773636 -0.2762853 0.0741253 -0.4586231 -0.0110632 -0.4590451 -0.8025933 -0.5478993 0.9019831

[64] 1.5973442 1.1809289 -0.6445365 -0.8080117 0.1029558 -0.6196456 -0.9469814 -1.9263839 -1.1244204

[73] 0.5216785 0.1836796 -1.1985679 0.3633361 -0.7151543 -0.2170542 -0.0349016 0.1545507 0.4830523

[82] -1.0175206 -0.7337119 -0.2568586 -0.3822954 0.5239054 -1.5578934 0.4852339 0.7339712 -0.6245461

[91] -0.1395845 0.0676763 -0.3808113 -1.5642023 -1

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.

Ans

```
num_vector <- c(3.14, 2.718, 1.618)
cat("Numeric vector:", num_vector, "\n")
cat("Type:", typeof(num_vector), "\n\n")
char_vector <- c("apple", "banana", "cherry")
cat("Character vector:", char_vector, "\n")
cat("Type:", typeof(char_vector), "\n\n")
log_vector <- c(TRUE, FALSE, TRUE)
cat("Logical vector:", log_vector, "\n")
cat("Type:", typeof(log_vector), "\n")
```

Output

Numeric vector: 3.14 2.718 1.618

Type: double

Character vector: apple banana cherry

Type: character

Logical vector: TRUE FALSE TRUE

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

Ans

```
m1 <- matrix(1:20, nrow = 5, byrow = TRUE)
print(m1)
m2 <- matrix(1:9, nrow = 3)
dimnames(m2) <- list(c("Row1", "Row2", "Row3"), c("Col1", "Col2", "Col3"))
print(m2)
m3 <- matrix(1:4, nrow = 2)
m3 <- matrix(1:4, nrow = 2, byrow = FALSE)
dimnames(m3) <- list(c("R1", "R2"), c("C1", "C2"))
print(m3)
```

Output

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
[5,]   17   18   19   20
```

```
    Col1 Col2 Col3
Row1    1    4    7
Row2    2    5    8
Row3    3    6    9
```

```
  C1 C2
R1  1  3
R2  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.

Ans

```
values <- c(1, 2, 3, 4, 5, 6)
dims <- c(2, 3, 1)
arr <- array(values, dim = dims, dimnames = list(c("A", "B"), c("X", "Y", "Z"), c("I")))
print(arr)
```

Output

```
, , I = 1

  X Y Z
A 1 3 5
B 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.

Ans

```
v1 <- 1:9
v2 <- 10:18
arr <- array(c(v1, v2), dim = c(3, 3, 2))
arr
```

Output

```
, , 1

     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

, , 2

     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   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.

Ans

```
v <- c(1, 2, 3)
m <- matrix(1:9, nrow = 3)
f <- function(x) x^2
my_list <- list(v, m, f)
print(my_list)
```

Output

[[1]]

[1] 1 2 3

[[2]]

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

[[3]]

function(x) x^2