ITA0464 R Programming (Lab Day 1)

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.

```
> name = readline(prompt="Input your name: ")
Input your name: pavan sai
> age = readline(prompt="Input your age: ")
Input your age: 19
> print(paste("My name is",name, "and I am",age ,"years old."))
[1] "My name is pavan sai and I am 19 years old."
> print(R.version.string)
[1] "R version 4.3.2 (2023-10-31 ucrt)"
>
```

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

```
> name = "Python";
> n1 =  10;
> n2 =  0.5
> nums = c(10, 20, 30, 40, 50, 60)
> print(ls())
 [1] "age"              "bins_clustering"  "bins_eq_freq"     "bins_eq_width"    "data"             "kmeans_result"    "mad_value"        "max_abs"
 [9] "n1"               "n2"               "name"             "normalized_data"  "nums"             "power"            "sales"            "z_score_normalized"
[17] "z_scores"
> print("Details of the objects in memory:")
[1] "Details of the objects in memory:"
> print(ls.str())
age :  chr "19"
bins_clustering :  Factor w/ 3 levels "Low","Medium",..: 3 3 3 3 3 3 1 1 1 1 ...
bins_eq_freq :  Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 1 1 1 1 1 1 1 1 2 ...
bins_eq_width :  Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 1 1 1 1 1 1 1 1 2 ...
data :  num [1:5] 200 300 400 600 1000
kmeans_result : List of 9
 $ cluster     : int [1:12] 3 3 3 3 3 3 1 1 1 1 ...
 $ centers     : num [1:3, 1] 67.2 209.5 14.8
 $ totss       : num 58568
 $ withinss    : num [1:3] 1082.8 60.5 544.8
 $ tot.withinss: num 1688
 $ betweenss   : num 56880
 $ size        : int [1:3] 4 2 6
 $ iter        : int 2
 $ ifault      : int 0
mad_value :  num 220
max_abs :  num 1000
n1 :  num 10
n2 :  num 0.5
name :  chr "Python"
normalized_data :  num [1:5] 0.2 0.3 0.4 0.6 1
nums :  num [1:6] 10 20 30 40 50 60
power :  num 3
sales :  num [1:12] 5 10 11 13 15 35 50 55 72 92 ...
z_score_normalized :  num [1:5] -0.909 -0.455 0 0.909 2.727
z_scores :  num [1:5] -0.909 -0.455 0 0.909 2.727
>
```

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

```
> 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 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.

```
> 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] -12 -10 -42   8 -35  -6   9  33  -3 -42
> 
```

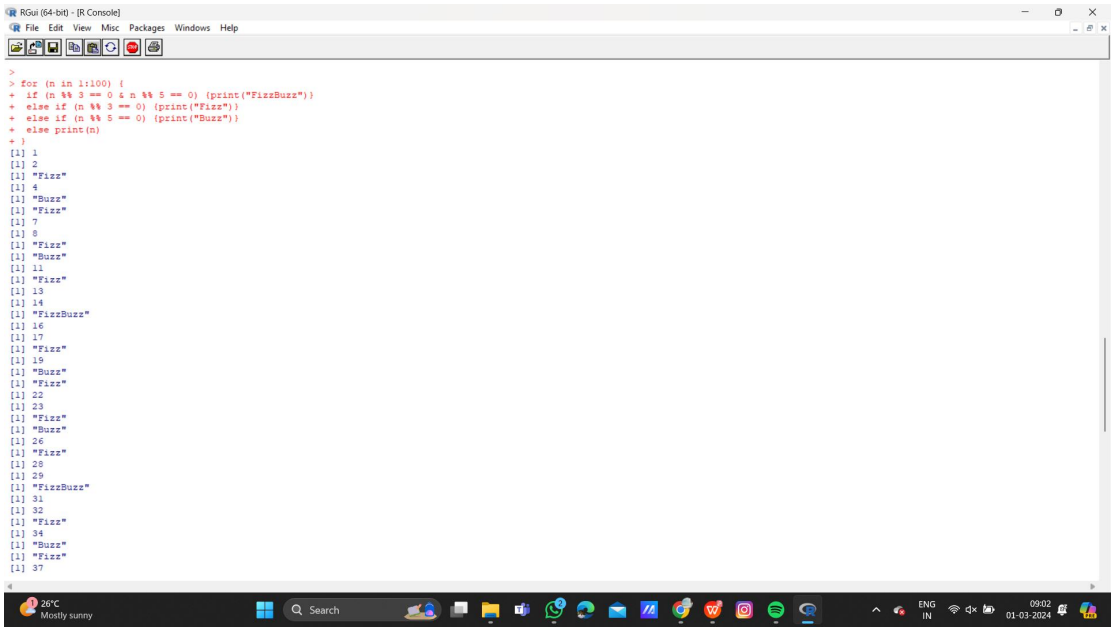
5. Write a R program to get the first 10 Fibonacci numbers.

```
>
> 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:")
[1] "First 10 Fibonacci numbers:"
> print(Fibonacci)
 [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)

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

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.



8. Write a R program to extract first 10 English letters in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.

```
>
> 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

```
> 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

```
>
> 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
```