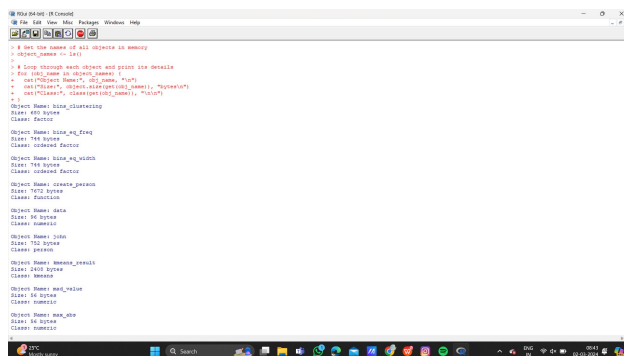


ITA0464 R Programming (Lab Day 1)

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



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
> |
```

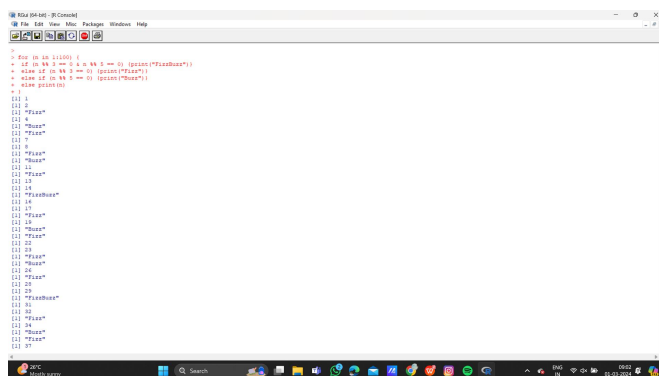
5. Write a R program to get the first 10 Fibonacci numbers.

```
> # Initialize the first two Fibonacci numbers
> fibonacci <- c(0, 1)
>
> # Generate the remaining Fibonacci numbers
> for (i in 3:10) {
+   next_fib <- fibonacci[i-1] + fibonacci[i-2]
+   fibonacci <- c(fibonacci, next_fib)
+ }
>
> # Print the first 10 Fibonacci numbers
> cat("First 10 Fibonacci numbers:", fibonacci, "\n")
First 10 Fibonacci numbers: 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)

```
>
> 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 = 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.



```
> 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)}
+ }
[1] 1
[1] 2
[1] "Fizz"
[1] 4
[1] "Buzz"
[1] "Fizz"
[1] 7
[1] "Fizz"
[1] 10
[1] "Buzz"
[1] 11
[1] "Fizz"
[1] 12
[1] "FizzBuzz"
[1] 13
[1] "Fizz"
[1] 14
[1] "Buzz"
[1] 15
[1] "FizzBuzz"
[1] 16
[1] "Fizz"
[1] 17
[1] "Buzz"
[1] 18
[1] "Fizz"
[1] 19
[1] "Buzz"
[1] 20
[1] "FizzBuzz"
[1] 21
[1] "Fizz"
[1] 22
[1] "Buzz"
[1] 23
[1] "Fizz"
[1] 24
[1] "FizzBuzz"
[1] 25
[1] "Buzz"
[1] 26
[1] "Fizz"
[1] 27
[1] "FizzBuzz"
[1] 28
[1] "Fizz"
[1] 29
[1] "Buzz"
[1] 30
[1] "FizzBuzz"
[1] 31
[1] "Fizz"
[1] 32
[1] "Buzz"
[1] 33
[1] "Fizz"
[1] 34
[1] "Buzz"
[1] 35
[1] "FizzBuzz"
[1] 36
[1] "Fizz"
[1] 37
[1] "Buzz"
[1] 38
[1] "Fizz"
[1] 39
[1] "FizzBuzz"
[1] 40
[1] "Buzz"
[1] 41
[1] "Fizz"
[1] 42
[1] "FizzBuzz"
[1] 43
[1] "Fizz"
[1] 44
[1] "Buzz"
[1] 45
[1] "FizzBuzz"
[1] 46
[1] "Fizz"
[1] 47
[1] "Buzz"
[1] 48
[1] "Fizz"
[1] 49
[1] "Buzz"
[1] 50
[1] "FizzBuzz"
[1] 51
[1] "Fizz"
[1] 52
[1] "Buzz"
[1] 53
[1] "Fizz"
[1] 54
[1] "FizzBuzz"
[1] 55
[1] "Buzz"
[1] 56
[1] "Fizz"
[1] 57
[1] "FizzBuzz"
[1] 58
[1] "Fizz"
[1] 59
[1] "Buzz"
[1] 60
[1] "FizzBuzz"
[1] 61
[1] "Fizz"
[1] 62
[1] "Buzz"
[1] 63
[1] "Fizz"
[1] 64
[1] "Buzz"
[1] 65
[1] "FizzBuzz"
[1] 66
[1] "Fizz"
[1] 67
[1] "Buzz"
[1] 68
[1] "Fizz"
[1] 69
[1] "FizzBuzz"
[1] 70
[1] "Buzz"
[1] 71
[1] "Fizz"
[1] 72
[1] "FizzBuzz"
[1] 73
[1] "Fizz"
[1] 74
[1] "Buzz"
[1] 75
[1] "FizzBuzz"
[1] 76
[1] "Fizz"
[1] 77
[1] "Buzz"
[1] 78
[1] "Fizz"
[1] 79
[1] "Buzz"
[1] 80
[1] "FizzBuzz"
[1] 81
[1] "Fizz"
[1] 82
[1] "Buzz"
[1] 83
[1] "Fizz"
[1] 84
[1] "FizzBuzz"
[1] 85
[1] "Buzz"
[1] 86
[1] "Fizz"
[1] 87
[1] "FizzBuzz"
[1] 88
[1] "Fizz"
[1] 89
[1] "Buzz"
[1] 90
[1] "FizzBuzz"
[1] 91
[1] "Fizz"
[1] 92
[1] "Buzz"
[1] 93
[1] "Fizz"
[1] 94
[1] "FizzBuzz"
[1] 95
[1] "Buzz"
[1] 96
[1] "Fizz"
[1] 97
[1] "FizzBuzz"
[1] 98
[1] "Fizz"
[1] 99
[1] "Buzz"
[1] 100
[1] "FizzBuzz"
```

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
> |
```