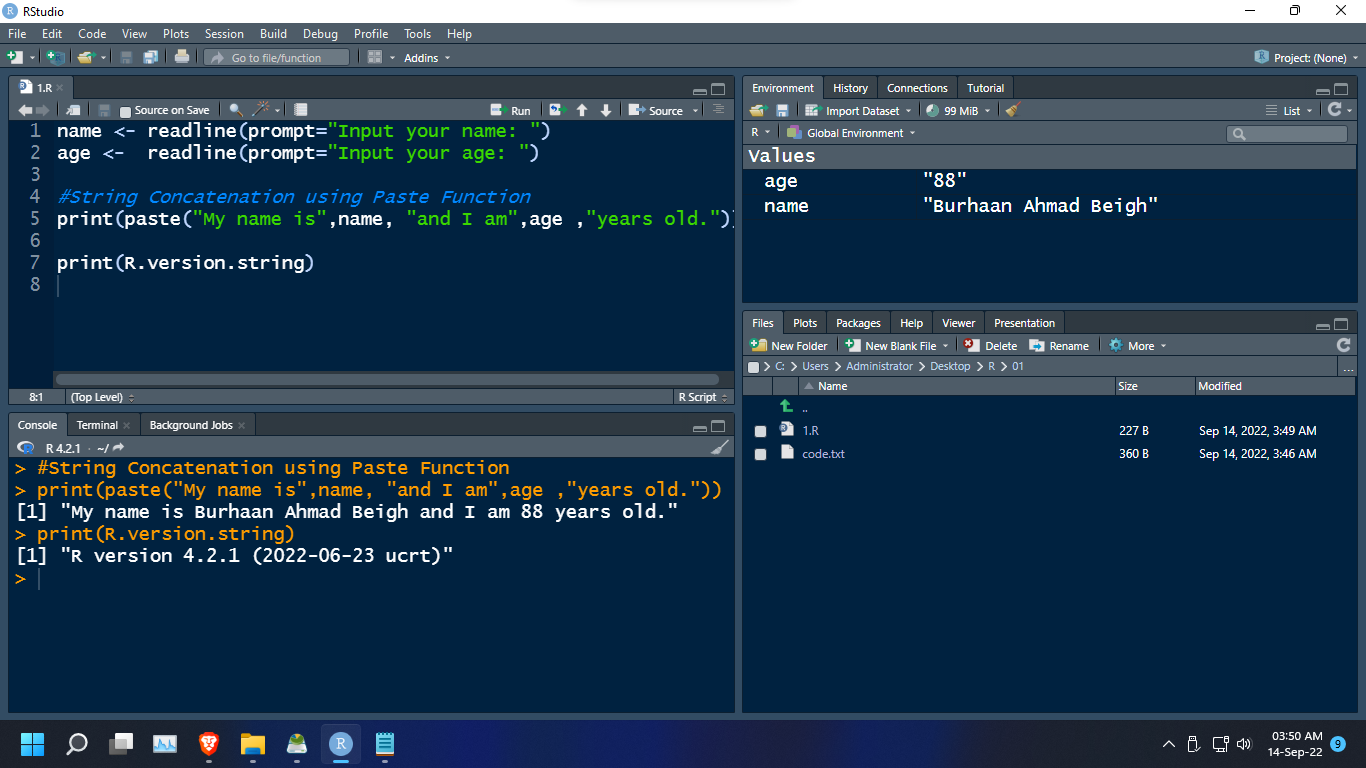
Burhaan Ahmad Beigh

182106

8th Semester – R Programming

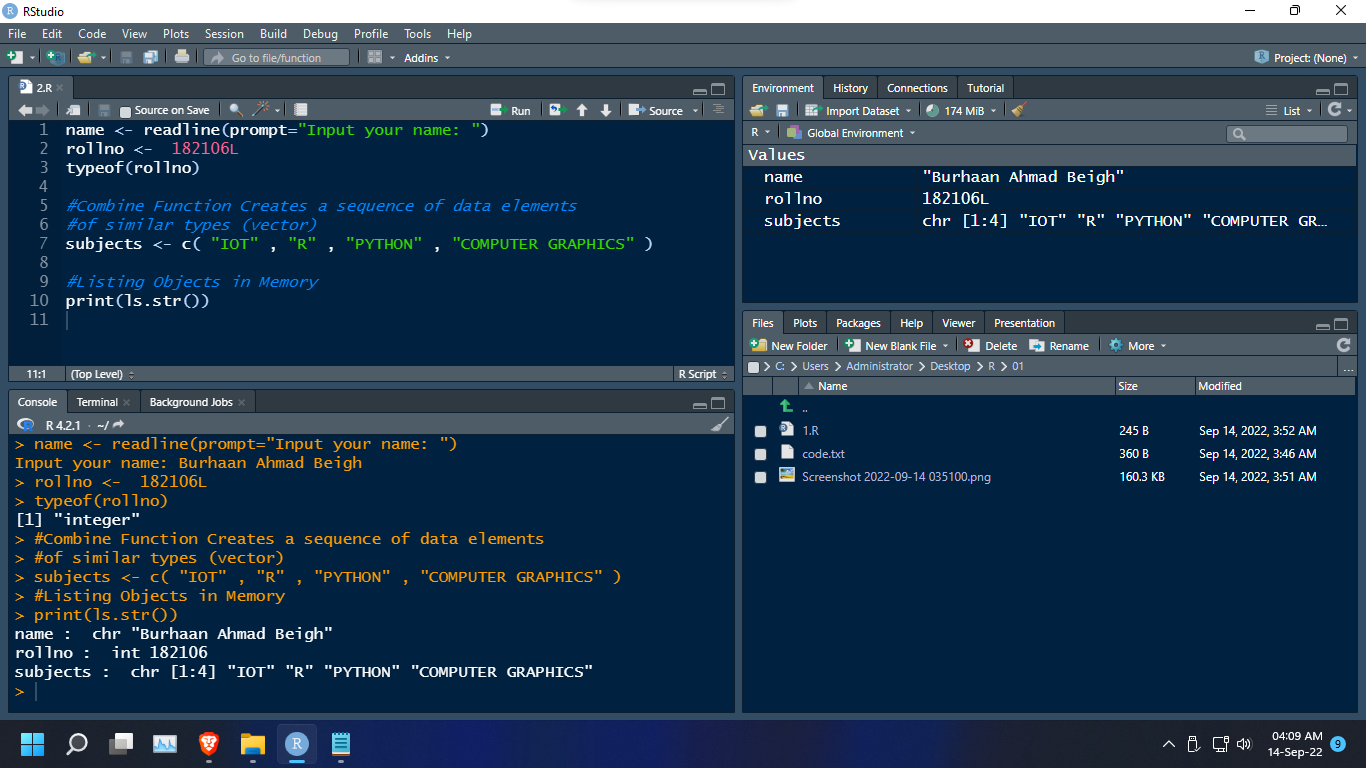
Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.

1. name <- readline(prompt="Input your name: ")
2. age <-  readline(prompt="Input your age: ")
4. *#String Concatenation using Paste Function*
5. **print(paste("My name is",name, "and I am",age ,"years old."))**
7. **print**(R.version.string)



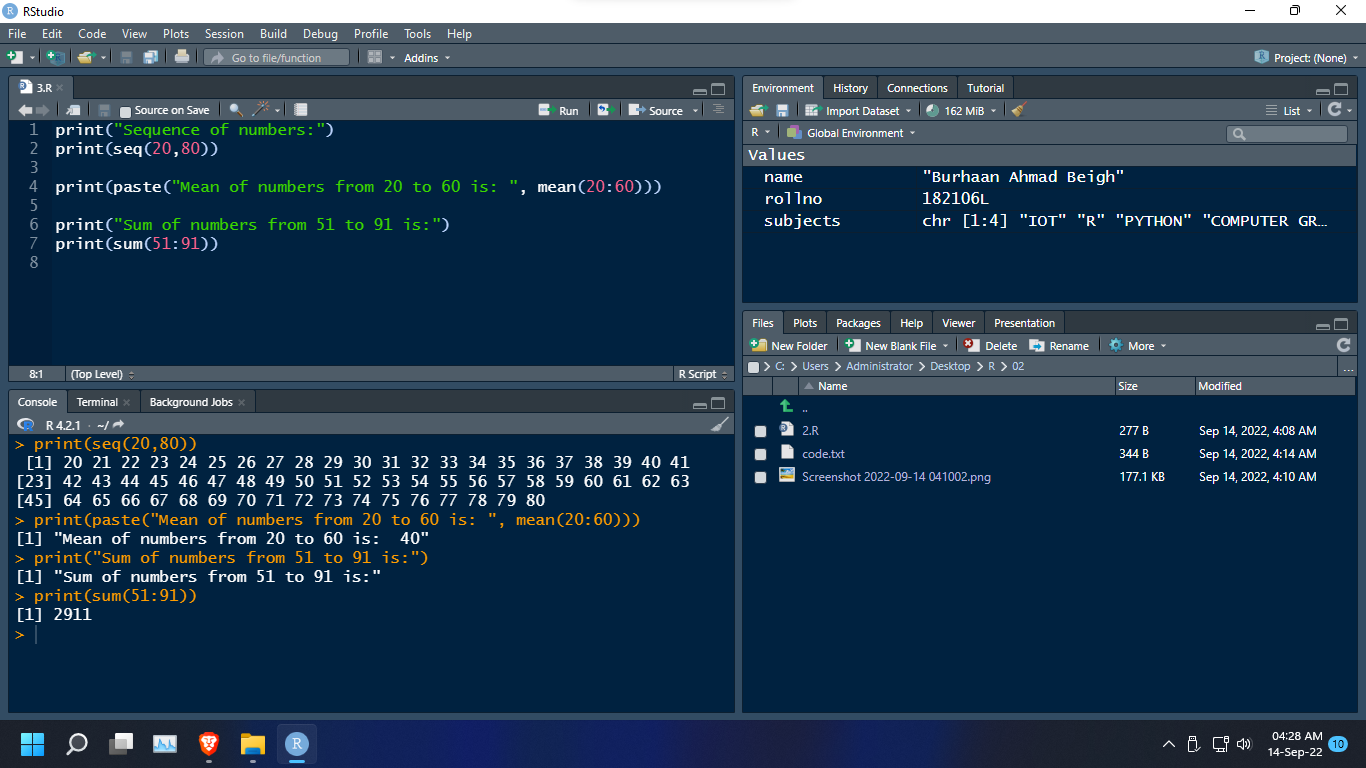
Write a R program to get the details of the objects in memory.

1. name <- readline(prompt="Input your name: ")
2. rollno <-  182106L
3. typeof(rollno)
5. ***#Combine Function Creates a sequence of data elements***
6. *#of similar types (vector)*
7. subjects <- c( "IOT" , "R" , "PYTHON" , "COMPUTER GRAPHICS" )
9. *#Listing Objects in Memory*
10. **print(ls.str())**



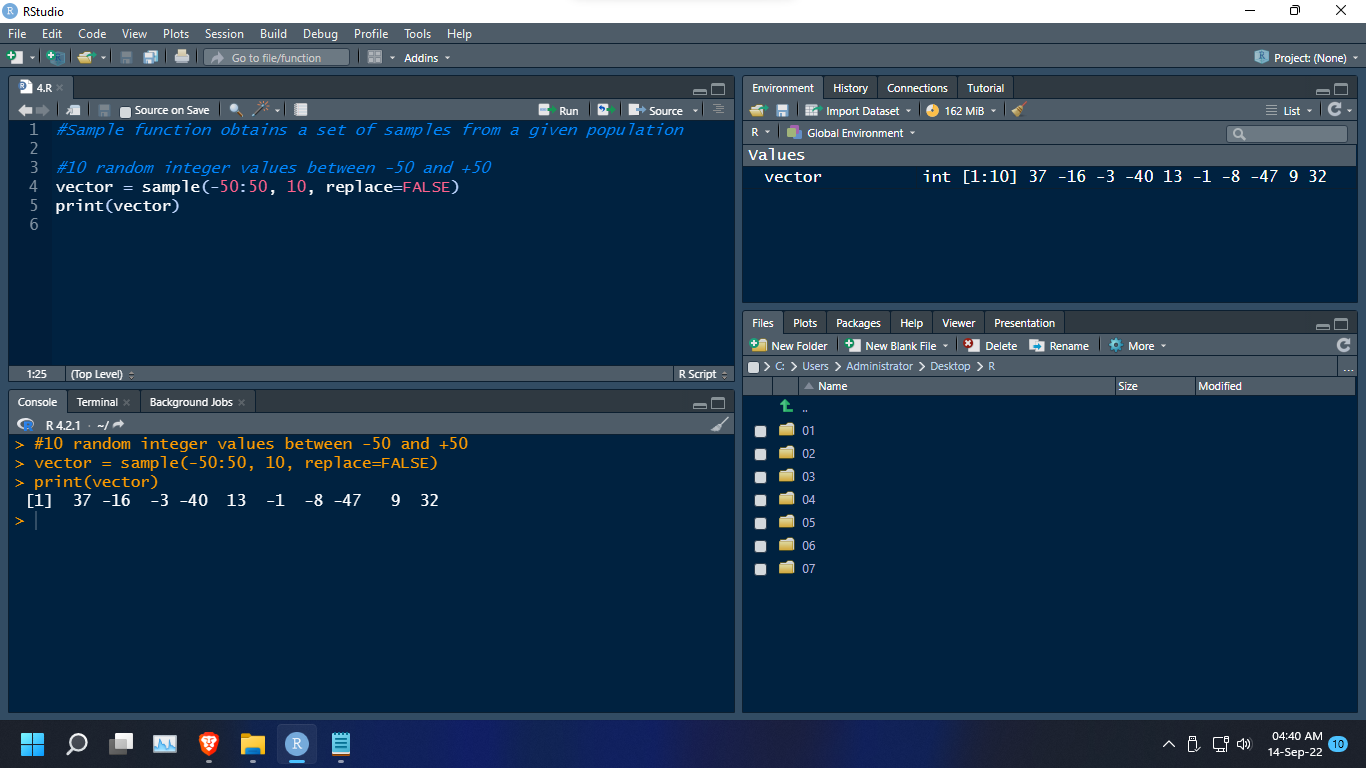
Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

1. **print**("Sequence of numbers:")
2. **print**(seq(20,80))
4. **print**(paste("Mean of numbers from 20 to 60 is: ", mean(20:60)))
6. **print**("Sum of numbers from 51 to 91 is:")
7. **print**(sum(51:91))



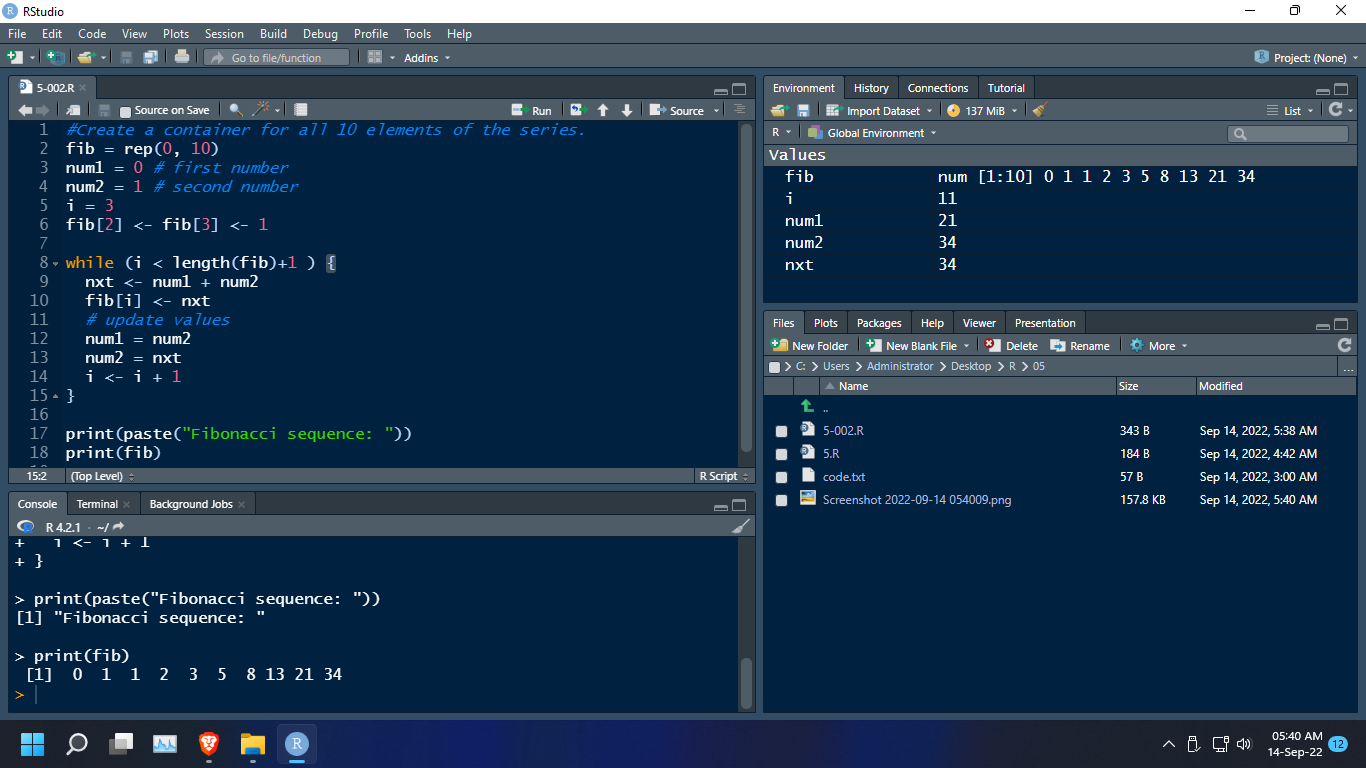
Write a R program to create a vector which contains 10 random integer values between -50 and +50.

1. *#Sample function obtains a set of samples from a given population*
3. *#10 random integer values between -50 and +50*
4. vector = sample(-50:50, 10, replace=FALSE)
5. **print(vector)**



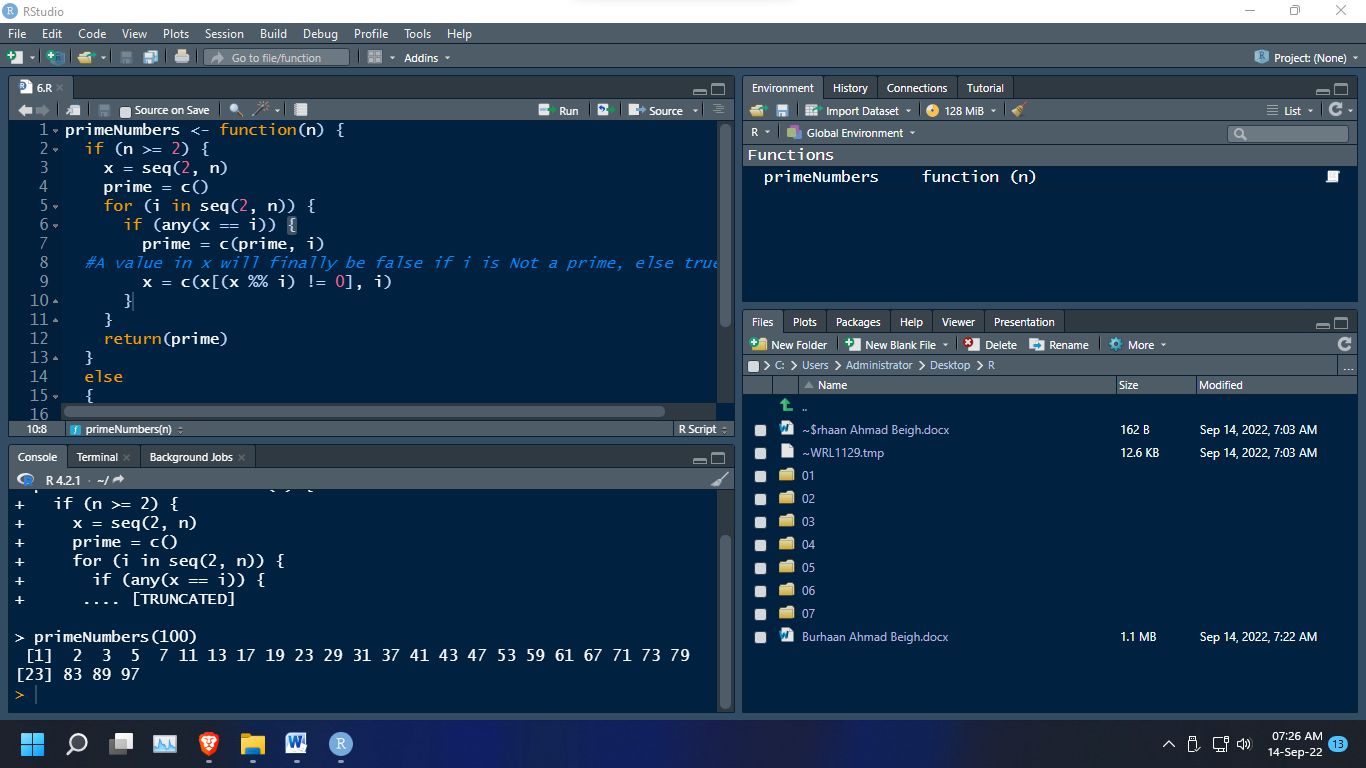
Write a R program to get the first 10 Fibonacci numbers.

1. *#Create a container for all 10 elements of the series.*
2. fib = rep(0, 10)
3. num1 = 0 *# first number*
4. num2 = 1 *# second number*
5. **i = 3**
6. fib[2] <- fib[3] <- 1
8. **while** (i < length(fib)+1 ) {
9. nxt <- num1 + num2
10. **fib[i] <- nxt**
11. *# update values*
12. num1 = num2
13. num2 = nxt
14. i <- i + 1
15. **}**
17. **print**(paste("Fibonacci sequence: "))
18. **print**(fib)



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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | primeNumbers < - function (n) {  **if** (n >= **2**) {  x = seq(**2**, n)  prime = c()  **for** (i **in** seq(**2**, n)) {  **if** (any(x == i)) {  prime = c(prime, i)  #A value in x will finally be false  **if** i **is** Not a prime,  **else** true.  x = c(x[(x % % i) != **0**], i)  }  }  **return** (prime)  } **else** {  stop("Input number should be at least 2.")  }  }  primeNumbers(**100)** |



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 "Fizz Buzz" for multiples of both.

1. **for** (i **in** 1:100) {
2. **if** (i %% 3 == 0 & i %% 5 == 0) {**print**("FizzBuzz")}
3. **else** **if** (i %% 3 == 0) {**print**("Fizz")}
4. **else** **if** (i %% 5 == 0) {**print**("Buzz")}
5. **else print(i)**
6. }

