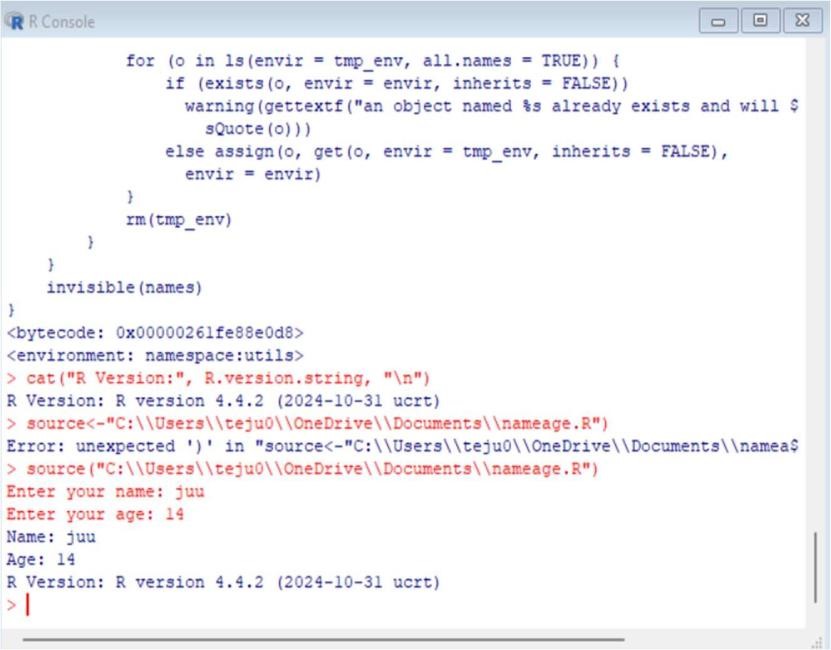
**Unit 1**

**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.**

**Code:**

name <- readline(prompt = "Enter your name: ") age <- readline(prompt = "Enter your age: ") cat("Name:", name, "\n") cat("Age:", age, "\n") cat("R Version:", R.version.string, "\n")

**output:**



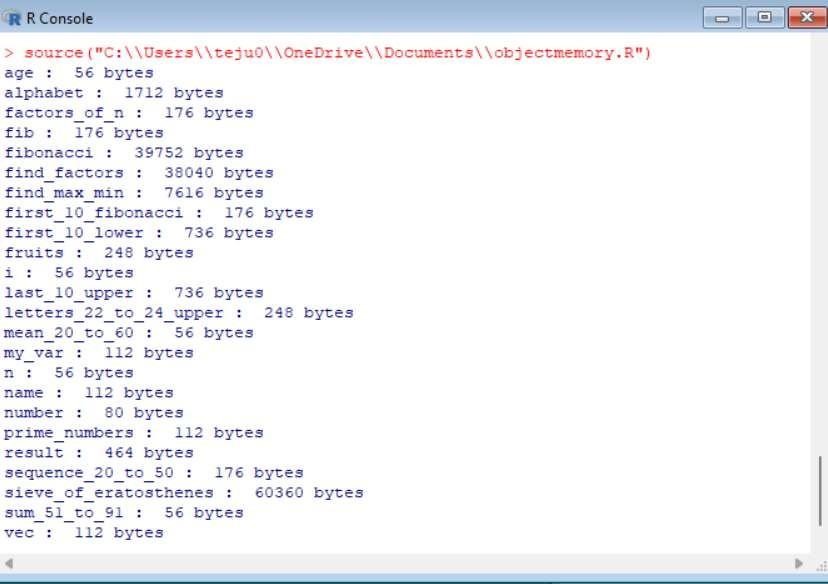
1. **Write a R program to get the details of the objects in memory**

**Code:**

objects <- ls() for (obj in objects) { cat(obj, ": ", object.size(get(obj)), "bytes\n")

}

**Output:**

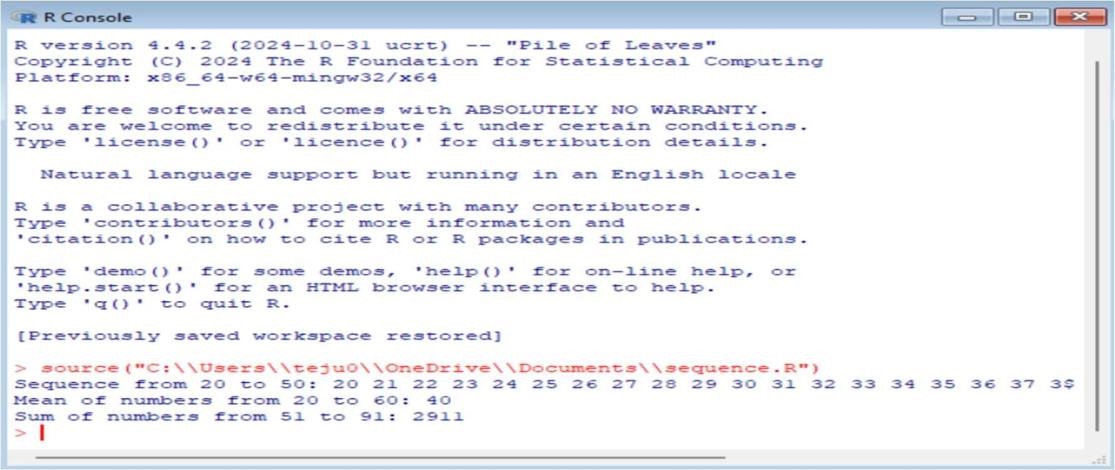


1. **Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91**

**Code:**

a<- seq(20, 50) b<- mean(seq(20, 60)) c<- sum(seq(51, 91)) cat("Sequence from 20 to 50:", a,"\n") cat("Mean of numbers from 20 to 60:", b,"\n") cat("Sum of numbers from 51 to 91:", c, "\n")

**output:**



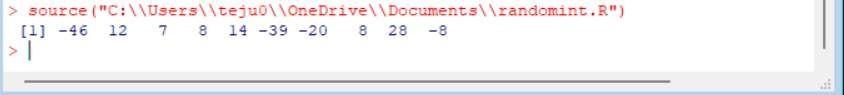
1. **Write a R program to create a vector which contains 10 random integer values between -50 and +50.**

**Code:**

random\_integers <- sample(-50:50, 10, replace =

TRUE) print(random\_integers)

**output:**



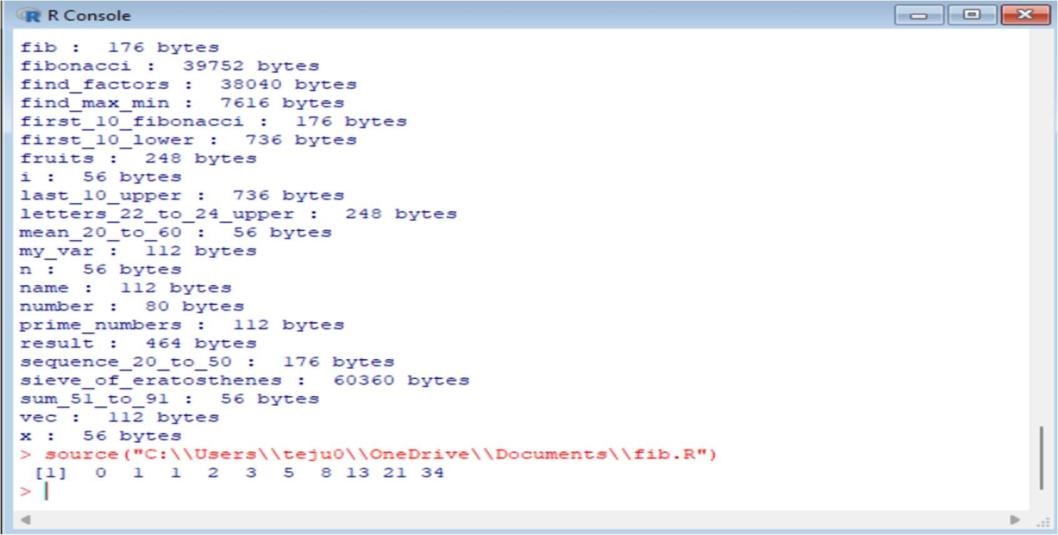
1. **Write a R program to get the first 10 Fibonacci numbers.**

**Code:**

fib <- c(0, 1) for (i in 3:10)

{ fib[i] <- fib[i - 1] + fib[i - 2]

} print(fib)  **output:**



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

**Code:**

a <- func on(n)

{

primes <- rep(TRUE, n+1) primes[1] <- FALSE # 0 and 1 are not prime numbers for (i in 2:sqrt(n))

{

if (primes[i] == TRUE)

{

primes[seq(i\*i, n, i)] <- FALSE

}

}

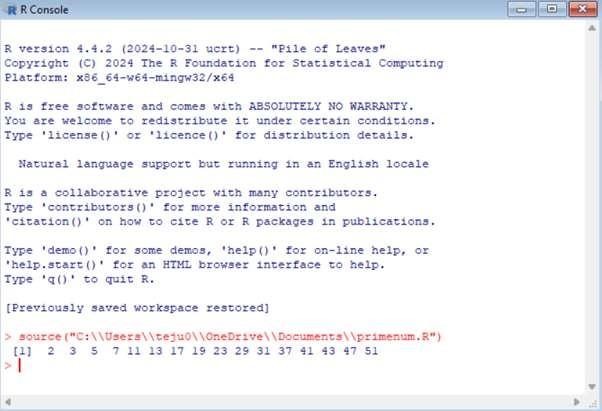
return(which(primes == TRUE))

}

n <- 50 prime\_numbers

<- a(n) print(prime\_numbers)

**output:**



**7.Write a R program to print the numbers from 1 to 10 and print "Fizz" for mul ples of 3, print "Buzz" for multiples of 5, and print "FizzBuzz" for multiples of both.**

**Code:**

for (i in 1:10)

{

if (i %% 3 == 0 && i %% 5 ==

0) { print("FizzBuzz")

} else if (i %% 3 == 0) { print("Fizz")

} else if (i %% 5 ==

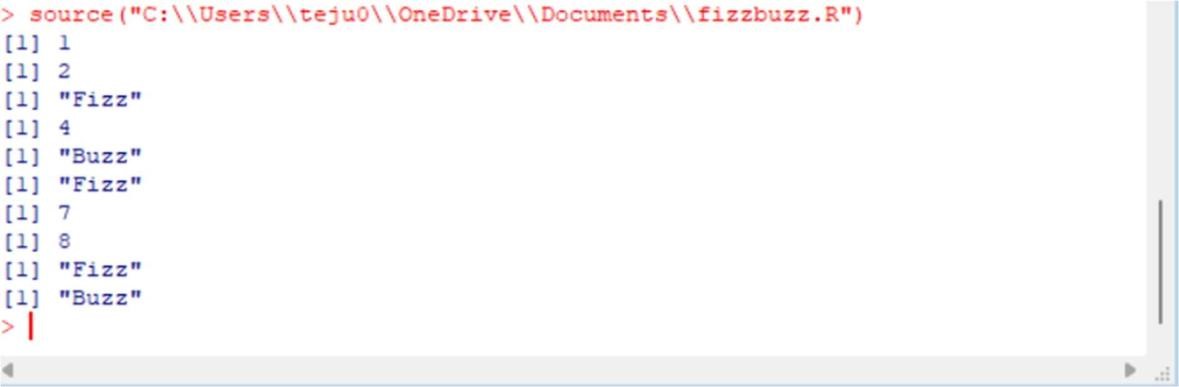
0) { print("Buzz")

} else { print(i)

}

}

**Output:**



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

**Code:**

alphabet <- le ers first\_10\_lower

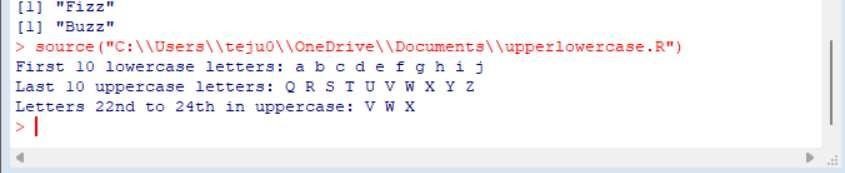
<- alphabet[1:10]

last\_10\_upper <- toupper(alphabet[(length(alphabet)- 9):

length(alphabet)]) le ers\_22\_to\_24\_upper <- toupper(alphabet[22:24]) cat("First 10 lowercase le ers:", first\_10\_lower, "\n") cat("Last 10 uppercase le ers:", last\_10\_upper, "\n")

cat("Le ers 22nd to 24th in uppercase:", le ers\_22\_to\_24\_upper,

"\n") **output:**



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

**Code:**

find\_factors <- func on(n)

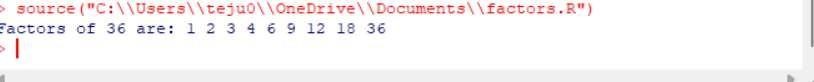
{ factors <- numeric(0) for (i in 1:n) { if (n %% i == 0) { factors <- c(factors, i)

} } return(factors)

}

n <- 36 factors\_of\_n <- find\_factors(n) cat("Factors of", n, "are:", factors\_of\_n, "\n")

**output:**



**10. Write a R program to find the maximum and the minimum value of a given vector** **Code:**  maxmin <- func on(vec)

{

a<- max(vec) b<- min(vec) return(list(max = a, min = b))

} vec <- c(3, 7, 1, 9, 4, 2, 8)

result <- maxmin(vec) cat("Maximum value:", result$max, "\n") cat("Minimum value:", result$min, "\n") **output:**

