

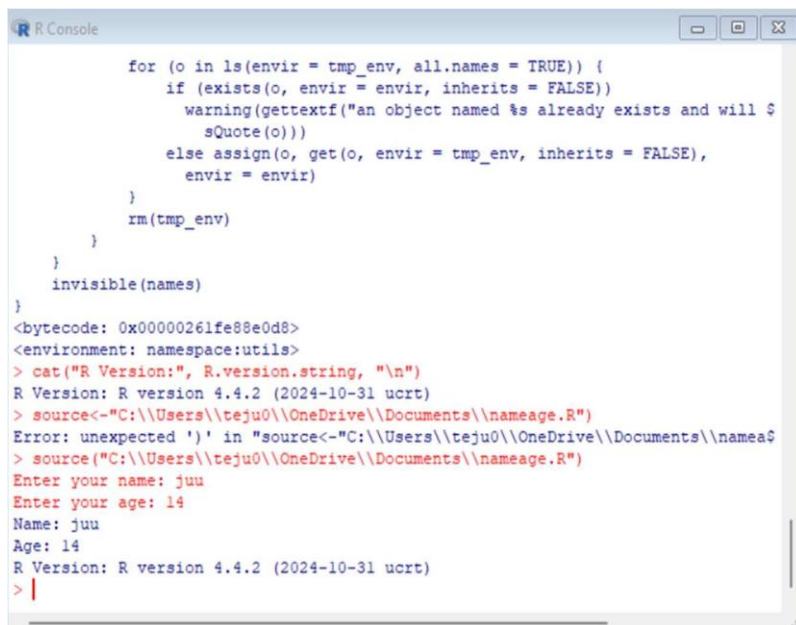
UNIT-1

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:



```
R Console

for (o in ls(envir = tmp_env, all.names = TRUE)) {
  if (exists(o, envir = envir, inherits = FALSE))
    warning(gettextf("an object named %s already exists and will %s",
      sQuote(o)))
  else assign(o, get(o, envir = tmp_env, inherits = FALSE),
    envir = envir)
}
rm(tmp_env)
}
invisible(names)
}
<bytecode: 0x00000261fe88e0d8>
<environment: namespace:utils>
> cat("R Version:", R.version.string, "\n")
R Version: R version 4.4.2 (2024-10-31 ucrt)
> source("C:\\Users\\teju0\\OneDrive\\Documents\\nameage.R")
Error: unexpected ')' in "source<-" "C:\\Users\\teju0\\OneDrive\\Documents\\namea$
> source("C:\\Users\\teju0\\OneDrive\\Documents\\nameage.R")
Enter your name: juu
Enter your age: 14
Name: juu
Age: 14
R Version: R version 4.4.2 (2024-10-31 ucrt)
> |
```

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

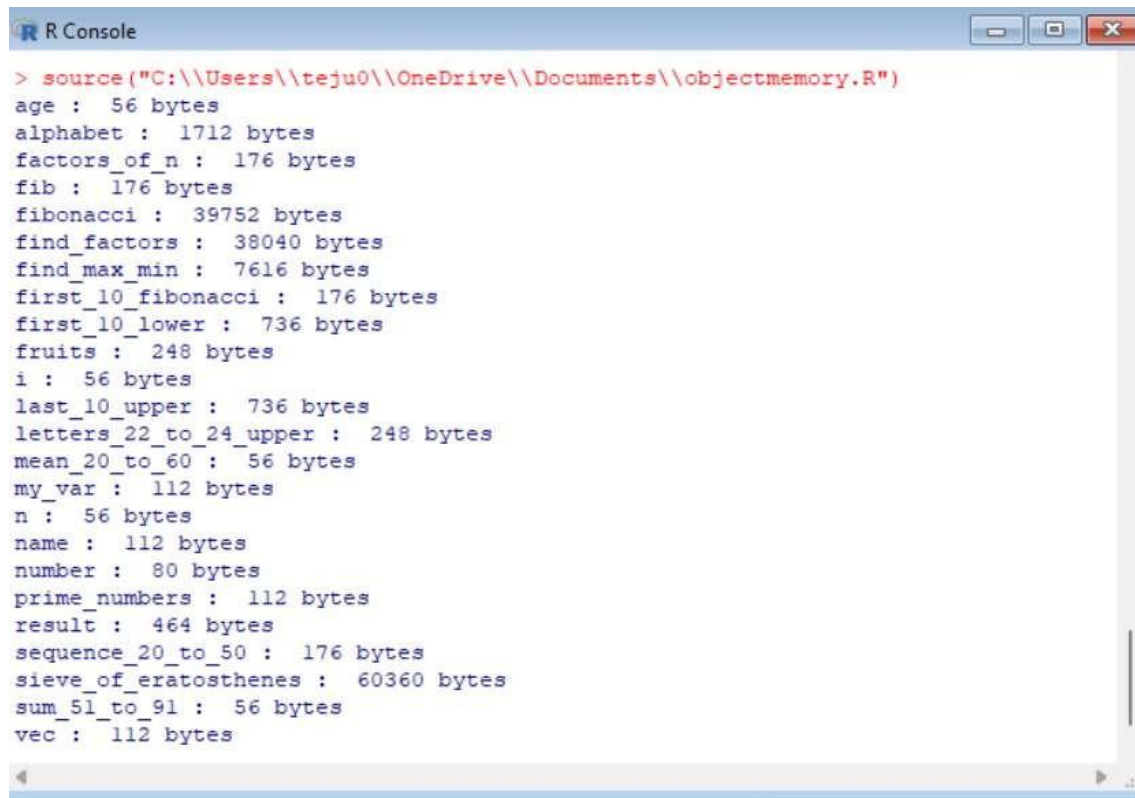
Code:

```
objects <- ls() for
(obj in objects) {

  cat(obj, ": ", object.size(get(obj)), "bytes\n")

}
```

Output:



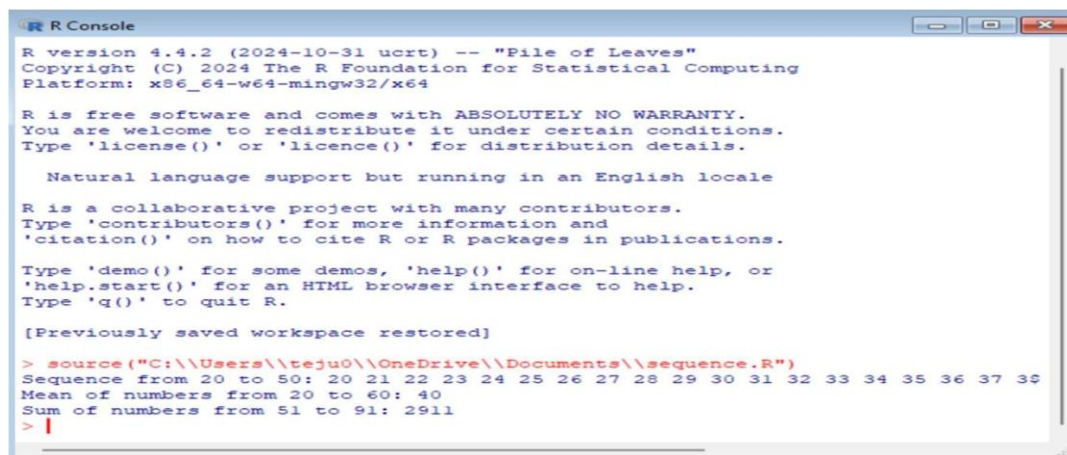
```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\objectmemory.R")
age : 56 bytes
alphabet : 1712 bytes
factors_of_n : 176 bytes
fib : 176 bytes
fibonacci : 39752 bytes
find_factors : 38040 bytes
find_max_min : 7616 bytes
first_10_fibonacci : 176 bytes
first_10_lower : 736 bytes
fruits : 248 bytes
i : 56 bytes
last_10_upper : 736 bytes
letters_22_to_24_upper : 248 bytes
mean_20_to_60 : 56 bytes
my_var : 112 bytes
n : 56 bytes
name : 112 bytes
number : 80 bytes
prime_numbers : 112 bytes
result : 464 bytes
sequence_20_to_50 : 176 bytes
sieve_of_eratosthenes : 60360 bytes
sum_51_to_91 : 56 bytes
vec : 112 bytes
```

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

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:



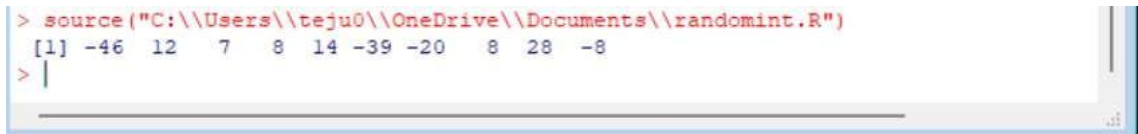
```
R Console  
R version 4.4.2 (2024-10-31 ucrt) -- "File of Leaves"  
Copyright (C) 2024 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
[Previously saved workspace restored]  
  
> source("C:\\Users\\teju0\\OneDrive\\Documents\\sequence.R")  
Sequence from 20 to 50: 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38  
Mean of numbers from 20 to 60: 40  
Sum of numbers from 51 to 91: 2911  
> |
```

4. 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:



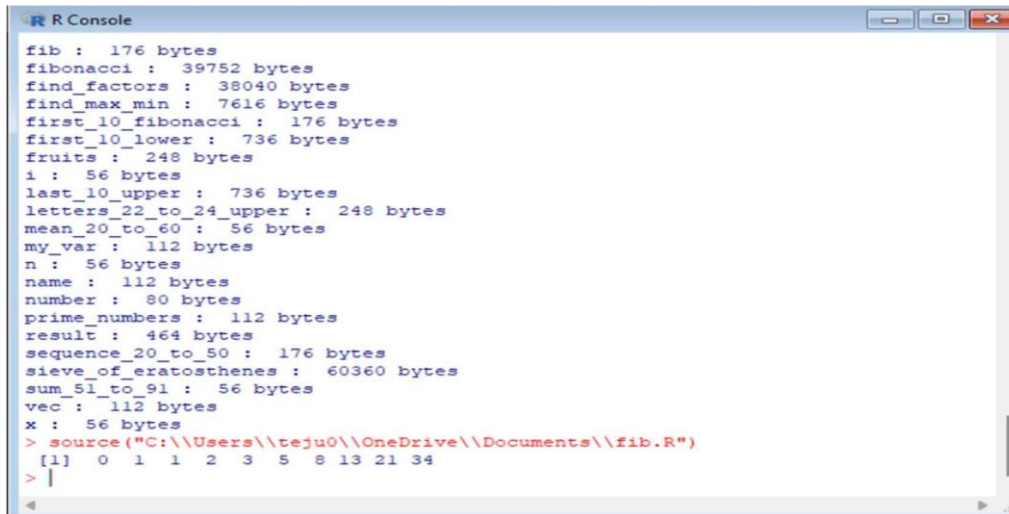
```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\randomint.R")  
[1] -46  12   7   8  14 -39 -20   8  28  -8  
> |
```

5. 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:



```
R Console
fib : 176 bytes
fibonacci : 39752 bytes
find_factors : 38040 bytes
find_max_min : 7616 bytes
first_10_fibonacci : 176 bytes
first_10_lower : 736 bytes
fruits : 248 bytes
i : 56 bytes
last_10_upper : 736 bytes
letters_22_to_24_upper : 248 bytes
mean_20_to_60 : 56 bytes
my_var : 112 bytes
n : 56 bytes
name : 112 bytes
number : 80 bytes
prime_numbers : 112 bytes
result : 464 bytes
sequence_20_to_50 : 176 bytes
sieve_of_eratosthenes : 60360 bytes
sum_51_to_91 : 56 bytes
vec : 112 bytes
x : 56 bytes
> source("C:\\Users\\teju0\\OneDrive\\Documents\\fib.R")
[1] 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)

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)
```

```

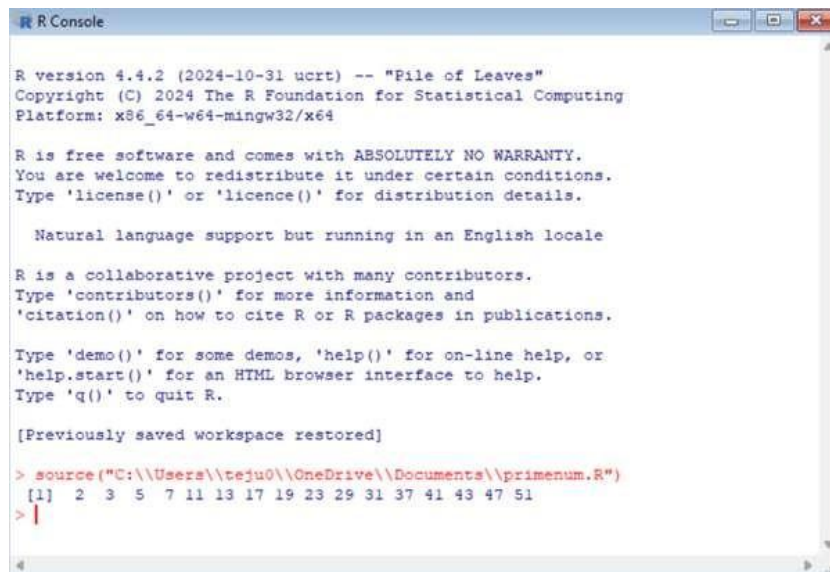
{
primes[seq(i*i, n, i)] <- FALSE
}
}

return(which(primes == TRUE))
}

n <- 50 prime_numbers
<- a(n)
print(prime_numbers)

```

output:



```

R Console

R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> source("C:\\Users\\teju0\\OneDrive\\Documents\\primenum.R")
[1] 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 51
>

```

7. Write a R program to print the numbers from 1 to 10 and print "Fizz" for multiples 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:



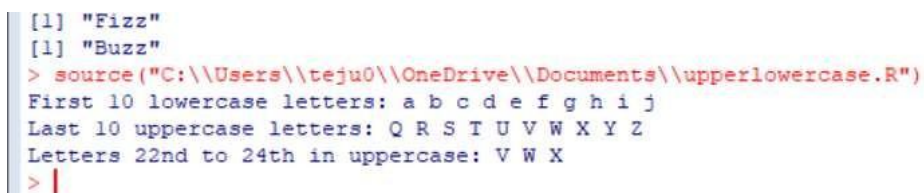
```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\fizzbuzz.R")
[1] 1
[1] 2
[1] "Fizz"
[1] 4
[1] "Buzz"
[1] "Fizz"
[1] 7
[1] 8
[1] "Fizz"
[1] "Buzz"
> |
```

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.

Code:

```
alphabet <- letters
first_10_lower <- alphabet[1:10]
last_10_upper <- toupper(alphabet[(length(alphabet)-
length(alphabet))]
letters_22_to_24_upper <- toupper(alphabet[22:24])
cat("First 10 lowercase letters:", first_10_lower, "\n")
cat("Last 10 uppercase letters:", last_10_upper, "\n")
cat("Letters 22nd to 24th in uppercase:", letters_22_to_24_upper,
"\n")
```

output:

A screenshot of an R console window showing the execution of an R script. The output displays the first 10 lowercase letters, the last 10 uppercase letters, and the letters from the 22nd to the 24th position in uppercase. The console has a light blue background and a white border.

```
[1] "Fizz"
[1] "Buzz"
> source("C:\\Users\\teju0\\OneDrive\\Documents\\upperlowercase.R")
First 10 lowercase letters: a b c d e f g h i j
Last 10 uppercase letters: Q R S T U V W X Y Z
Letters 22nd to 24th in uppercase: V W X
> |
```


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

Code:

```
find_factors <- function(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:

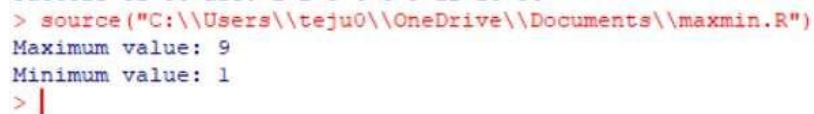
```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\factors.R")
Factors of 36 are: 1 2 3 4 6 9 12 18 36
> |
```

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:

A screenshot of an R console window showing the execution of the R program. The first line is a red prompt character followed by the command to source the file 'maxmin.R' from the path 'C:\\Users\\teju0\\OneDrive\\Documents\\'. The subsequent two lines are the output of the program: 'Maximum value: 9' and 'Minimum value: 1'. The prompt character is followed by a vertical bar, indicating the next command.

```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\maxmin.R")
Maximum value: 9
Minimum value: 1
> |
```