

# Introduction to R Programming Lab

## (BTCCSPCP501)

An Assignment Work Submitted for the 5th Semester  
of Bachelor of Technology

*in*

Computer Science and Engineering (Data Science)

*by*

Suman Mondal

Registration Number: 100227240046



DEPARTMENT OF COMPUTER SCIENCE  
KAZI NAZRUL UNIVERSITY  
ASANSOL - 713340, WEST BENGAL

# Contents

<b>1</b>	<b>Assignment 1</b>	<b>1</b>
1.1	Write a R program to find the sum of two numbers . . . . .	1
1.2	Write a R program to perform all arithmetic operations of two numbers . . . . .	1
1.3	Write a R program to find the perimeter and area of rectangle . . . . .	2
1.4	Write a R program to find the circumference and area of circle . . . . .	3
1.5	Write a R program to change the temperature in Celsius to Fahrenheit . . . . .	3
1.6	Write a R program to calculate total, average of 5 student marks . . . . .	4
1.7	Write a R program to calculate Simple Interest and compound interest . . . . .	4
1.8	Write a R program to swapping of two numbers using third variable . . . . .	5
<b>2</b>	<b>Assignment 2</b>	<b>7</b>
2.1	Write a R program to find maximum between two numbers . . . . .	7
2.2	Write a R program to find minimum between three numbers . . . . .	7
2.3	Write a R program to check whether a number is negative, positive or zero . . . . .	8
2.4	Write a R program to check whether a number is divisible by 5 and 11 or not . . . . .	8
2.5	Write a R program to check whether a number is even or odd . . . . .	9
2.6	Write a R program to check whether a year is leap year or not . . . . .	9
2.7	Write a R program to check whether a character is uppercase or lowercase alphabet . . . . .	10
2.8	Write a R program to input any alphabet and check whether it is vowel or consonant . . . . .	10
<b>3</b>	<b>Assignment 3</b>	<b>12</b>
3.1	Write a R program to print all natural numbers from 1 to n . . . . .	12
3.2	Write a R program to print all odd number between 1 to 100 . . . . .	12
3.3	Write a R program to find sum of all natural numbers between 1 to n . . . . .	13
3.4	Write a R program to find sum of all even numbers between 1 to n . . . . .	14
3.5	Write a R program to count number of digits in a number . . . . .	15
3.6	Write a R program to calculate sum of digits of a number . . . . .	15
3.7	Write a R program to calculate product of digits of a number . . . . .	16
3.8	Write a R program to enter a number and print its reverse . . . . .	17
3.9	Write a R program to check whether a number is palindrome or not . . . . .	17
3.10	Write a R program to find power of a number using for loop . . . . .	18
3.11	Write a R program to find all factors of a number . . . . .	19
3.12	Write a R program to calculate factorial of a number . . . . .	20
3.13	Write a R program to find HCF (GCD) of two numbers . . . . .	20
3.14	Write a R program to find LCM of two numbers . . . . .	21
3.15	Write a R program to check whether a number is Prime number or not . . . . .	22
3.16	Write a R program to check whether a number is Armstrong number or not . . . . .	23
3.17	Write a R program to check whether a number is Perfect number or not . . . . .	24
3.18	Write a R program to print Fibonacci series up to n terms . . . . .	25

# Chapter 1

## Assignment 1

### 1.1 Write a R program to find the sum of two numbers

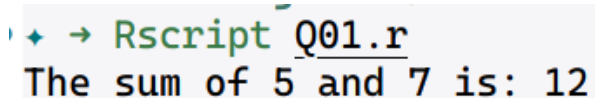
Source Code :

```
# Q01. Write a R program to find the sum of two numbers

num1 <- 5
num2 <- 7

sum_result <- num1 + num2
cat("The sum of", num1, "and", num2, "is:", sum_result, "\n")
```

Program Output :



```
→ Rscript Q01.r
The sum of 5 and 7 is: 12
```

### 1.2 Write a R program to perform all arithmetic operations of two numbers

Source Code :

```
# Q02. Write a R program to perform all arithmetic operations
→ of two numbers

# Define two numbers
num1 <- 10
num2 <- 5

# Addition
sum_result <- num1 + num2
cat("Sum:", sum_result, "\n")
```

```
# Subtraction
diff_result <- num1 - num2
cat("Difference:", diff_result, "\n")

# Multiplication
prod_result <- num1 * num2
cat("Product:", prod_result, "\n")

# Division
div_result <- num1 / num2
cat("Division:", div_result, "\n")
```

Program Output :

```
✦ → Rscript Q02.r
Sum: 15
Difference: 5
Product: 50
Division: 2
```

### 1.3 Write a R program to find the perimeter and area of rectangle

Source Code :

```
# Q03. Write a R program to find the perimeter and area of
→ rectangle

length <- 8
width <- 5

# Calculate the area of the rectangle
area <- length * width

# Calculate the perimeter of the rectangle
perimeter <- 2 * (length + width)

cat("Length of the rectangle: ", length, "\n")
```

```
cat("Width of the rectangle: ", width, "\n")
cat("Area of the rectangle: ", area, "\n")
cat("Perimeter of the rectangle: ", perimeter, "\n")
```

Program Output :

```
✦ → Rscript Q03.r
Length of the rectangle: 8
Width of the rectangle: 5
Area of the rectangle: 40
Perimeter of the rectangle: 26
```

#### 1.4 Write a R program to find the circumference and area of circle

Source Code :

```
# Q04. Write a R program to find the circumference and area of
→ circle
```

```
radius <- 6
area <- pi * radius^2
circumference <- 2 * pi * radius
```

```
# Print the results
cat("Radius of the circle: ", radius, "\n")
cat("Area of the circle: ", area, "\n")
cat("Circumference of the circle: ", circumference, "\n")
```

Program Output :

```
✦ → Rscript Q04.r
Radius of the circle: 6
Area of the circle: 113.0973
Circumference of the circle: 37.69911
```

#### 1.5 Write a R program to change the temperature in Celsius to Fahrenheit

Source Code :

```
# Q05. Write a R program to change the temperature in Celsius
→ to Fahrenheit
```

```
celsius_temp <- 25
fahrenheit_temp <- (celsius_temp * 9/5) + 32
```

```
cat("Temperature in Celsius: ", celsius_temp, "°C\n")
cat("Temperature in Fahrenheit: ", fahrenheit_temp, "°F\n")
```

Program Output :

```
✦ → Rscript Q05.r
Temperature in Celsius: 25 °C
Temperature in Fahrenheit: 77 °F
```

## 1.6 Write a R program to calculate total, average of 5 student marks

Source Code :

```
# Q06. Write a R program to calculate total, average of 5
→ student marks
```

```
marks <- c(85, 92, 78, 90, 88)
total_marks <- sum(marks)
average_marks <- total_marks / length(marks)
```

```
cat("Total Marks: ", total_marks, "\n")
cat("Average Marks: ", average_marks, "\n")
```

Program Output :

```
✦ → Rscript Q06.r
Total Marks: 433
Average Marks: 86.6
```

## 1.7 Write a R program to calculate Simple Interest and compound interest

Source Code :

```
# Q07. Write a R program to calculate Simple Interest and
→ compound interest
```

```
principal <- 1000
rate <- 0.05
time <- 3
```

```
simple_interest <- (principal * rate * time)
compound_interest <- principal * (1 + rate)^time - principal

cat("Principal Amount: Rs.", principal, "\n")
cat("Annual Interest Rate: ", rate * 100, "%\n")
cat("Time Period (years): ", time, "\n")
cat("Simple Interest: Rs.", simple_interest, "\n")
cat("Compound Interest: Rs.", compound_interest, "\n")
```

Program Output :

```
✦ → Rscript Q07.r
Principal Amount: Rs. 1000
Annual Interest Rate:  5 %
Time Period (years):  3
Simple Interest: Rs. 150
Compound Interest: Rs. 157.625
```

## 1.8 Write a R program to swapping of two numbers using third variable

Source Code :

```
# Q08. Write a R program to swapping of two numbers using third
→ variable
```

```
num1 <- 10
num2 <- 20

cat("Original values:\n")
cat("num1:", num1, "\n")
cat("num2:", num2, "\n")

temp <- num1
num1 <- num2
num2 <- temp

cat("Swapped values:\n")
cat("num1:", num1, "\n")
cat("num2:", num2, "\n")
```

Program Output :

```
◆ → Rscript Q08.r
Original values:
num1: 10
num2: 20
Swapped values:
num1: 20
num2: 10
```



## Chapter 2

# Assignment 2

### 2.1 Write a R program to find maximum between two numbers

Source Code :

```
# Q01. Write a R program to find maximum between two numbers
```

```
a <- 200
```

```
b <- 33
```

```
if (b > a) {  
  print("b is greater than a")  
} else {  
  print("a is greater than b")  
}
```

Program Output :

```
→ Rscript Q01.r  
1] "a is greater than b"
```

### 2.2 Write a R program to find minimum between three numbers

Source Code :

```
# Q02. Write a R program to find minimum between three numbers
```

```
a <- 34
```

```
b <- 25
```

```
c <- -7
```

```
if (a < b) {
```

```
    print ("a is smaller than b and c")
} else if (b < c) {
    print ("b is smaller than a and c")
} else {
    print ("c is smaller than a and b")
}
```

Program Output :

```
→ Rscript Q02.r
1] "c is smaller than a and b"
```

### 2.3 Write a R program to check whether a number is negative, positive or zero

Source Code :

```
# Q03. Write a R program to check whether a number is negative,
→ positive or zero
```

```
a <- 6
```

```
if (a > 0) {
    print ("the number is positive")
} else if (a == 0) {
    print ("the number is equal to zero")
} else if (a < 0){
    print ("the number is negative")
}
```

Program Output :

```
→ Rscript Q03.r
1] "the number is positive"
```

### 2.4 Write a R program to check whether a number is divisible by 5 and 11 or not

Source Code :

```
# Q04. Write a R program to check whether a number is divisible
→ by 5 and 11 or not
```

```
number <- 55

is_divisible_by_5_and_11 <- (number %% 5 == 0) && (number %% 11
→ == 0)

if (is_divisible_by_5_and_11) {
  cat(number, "is divisible by both 5 and 11.\n")
} else {
  cat(number, "is not divisible by both 5 and 11.\n")
}
```

Program Output :

```
➤ → Rscript Q04.r
55 is divisible by both 5 and 11.
```

## 2.5 Write a R program to check whether a number is even or odd

Source Code :

```
# Q05. Write a R program to check whether a number is even or
→ odd
```

```
number <- 7

if (number %% 2 == 0) {
  cat(number, "is an even number.\n")
} else {
  cat(number, "is an odd number.\n")
}
```

Program Output :

```
➤ → Rscript Q05.r
7 is an odd number.
```

## 2.6 Write a R program to check whether a year is leap year or not

Source Code :

```
# Q06. Write a R program to check whether a year is leap year
→ or not
```

```
year <- 2024
```

```
if ((year %% 4 == 0 && year %% 100 != 0) || year %% 400 == 0) {  
  cat(year, "is a leap year.\n")  
} else {  
  cat(year, "is not a leap year.\n")  
}
```

Program Output :

```
➦ → Rscript Q06.r  
2024 is a leap year.
```

## 2.7 Write a R program to check whether a character is uppercase or lowercase alphabet

Source Code :

```
# Q07. Write a R program to check whether a character is  
→ uppercase or lowercase alphabet
```

```
char <- "A"
```

```
if (char %in% letters) {  
  cat(char, "is a lowercase alphabet.\n")  
} else if (char %in% LETTERS) {  
  cat(char, "is an uppercase alphabet.\n")  
} else {  
  cat(char, "is not an alphabet.\n")  
}
```

Program Output :

```
➦ → Rscript Q07.r  
A is an uppercase alphabet.
```

## 2.8 Write a R program to input any alphabet and check whether it is vowel or consonant

Source Code :

*# Q08. Write a R program to input any alphabet and check  
→ whether it is vowel or consonant*

```
alphabet <- readline(prompt = "Enter a single alphabet: ")
```

```
alphabet <- tolower(alphabet)
```

```
if (nchar(alphabet) != 1) {  
  cat("Please enter a single alphabet.\n")  
} else if (alphabet %in% c("a", "e", "i", "o", "u")) {  
  cat(alphabet, "is a vowel.\n")  
} else if (alphabet %in% letters) {  
  cat(alphabet, "is a consonant.\n")  
} else {  
  cat("The input is not an alphabet.\n")  
}
```

Program Output :

```
► + → Rscript Q08.r  
Enter a single alphabet:  
Please enter a single alphabet.
```

## Chapter 3

# Assignment 3

### 3.1 Write a R program to print all natural numbers from 1 to n

Source Code :

```
# Q01. Write a R program to print all natural numbers from 1 to  
→ n
```

```
print_natural_numbers <- function(n) {  
  if (n <= 0) {  
    cat("Please provide a positive integer value for n.\n")  
    return(NULL)  
  }  
  
  for (i in 1:n) {  
    cat(i, " ")  
  }  
  cat("\n")  
}
```

```
print_natural_numbers(10)
```

Program Output :

```
knuce-assignment/Fifth Semester/  
• → Rscript Q01.r  
1 2 3 4 5 6 7 8 9 10
```

### 3.2 Write a R program to print all odd number between 1 to 100

Source Code :

*# Q02. Write a R program to print all odd number between 1 to  
→ 100*

```
print_odd_numbers <- function() {
  for (i in 1:100) {
    if (i %% 2 != 0) {
      cat(i, " ")
    }
  }
  cat("\n")
}
```

```
print_odd_numbers()
```

Program Output :

```
knucse-assignment/Fifth Semester/R Lang/Assignme |
nt_3 on □ main [?]
•→ Rscript Q02.r
1 3 5 7 9 11 13 15 17 19 21 23 25 2
7 29 31 33 35 37 39 41 43 45 47 49 5
1 53 55 57 59 61 63 65 67 69 71 73 7
5 77 79 81 83 85 87 89 91 93 95 97 9
9
```

### 3.3 Write a R program to find sum of all natural numbers between 1 to n

Source Code :

*# Q03. Write a R program to find sum of all natural numbers  
→ between 1 to n*

```
sum_of_natural_numbers <- function(n) {
  if (n <= 0) {
    cat("Please provide a positive integer value for n.\n")
    return(NULL)
  }

  sum <- 0
  for (i in 1:n) {
    sum <- sum + i
  }

  return(sum)
}
```

```
}

n <- 10
result <- sum_of_natural_numbers(n)
cat("The sum of natural numbers from 1 to", n, "is:", result,
    "\n")
```

### Program Output :

```
→ Rscript Q03.r
The sum of natural numbers from 1 to 10 is: 55
```

### 3.4 Write a R program to find sum of all even numbers between 1 to n

#### Source Code :

```
# Q04. Write a R program to find sum of all even numbers
→ between 1 to n

sum_of_even_numbers <- function(n) {
  if (n <= 0) {
    cat("Please provide a positive integer value for n.\n")
    return(NULL)
  }

  sum <- 0
  for (i in 2:n) {
    if (i %% 2 == 0) {
      sum <- sum + i
    }
  }

  return(sum)
}

n <- 10
result <- sum_of_even_numbers(n)
cat("The sum of even numbers from 1 to", n, "is:", result, "\n")
```

### Program Output :



→ Rscript Q04.r

The sum of even numbers from 1 to 10 is: 30

### 3.5 Write a R program to count number of digits in a number

Source Code :

*# Q05. Write a R program to count number of digits in a number*

```
count_digits <- function(number) {  
  if (number < 0) {  
    number <- -number  
  }  
  
  num_digits <- 0  
  while (number > 0) {  
    number <- number %/% 10  
    num_digits <- num_digits + 1  
  }  
  
  return(num_digits)  
}  
  
number <- 863784638  
num_digits <- count_digits(number)  
cat("The number of digits in", number, "is:", num_digits, "\n")
```

Program Output :

→ Rscript Q05.r

The number of digits in 863784638 is: 9

### 3.6 Write a R program to calculate sum of digits of a number

Source Code :

*# Q06. Write a R program to calculate sum of digits of a number*

```
sum_of_digits <- function(number) {  
  if (number < 0) {  
    number <- -number  
  }  
}
```

```
sum <- 0
while (number > 0) {
  digit <- number %% 10
  sum <- sum + digit
  number <- number %/% 10
}

return(sum)
}

number <- 863784638
digit_sum <- sum_of_digits(number)
cat("The sum of digits in", number, "is:", digit_sum, "\n")
```

Program Output :

```
→ Rscript Q06.r
The sum of digits in 863784638 is: 53
```

### 3.7 Write a R program to calculate product of digits of a number

Source Code :

```
# Q07. Write a R program to calculate product of digits of a
→ number
```

```
product_of_digits <- function(number) {
  if (number < 0) {
    number <- -number
  }

  product <- 1
  while (number > 0) {
    digit <- number %% 10
    product <- product * digit
    number <- number %/% 10
  }

  return(product)
}
```

```
number <- 863784638
digit_product <- product_of_digits(number)
cat("The product of digits in", number, "is:", digit_product,
    "\n")
```

Program Output :

```
→ Rscript Q07.r
The product of digits in 863784638 is: 4644864
```

### 3.8 Write a R program to enter a number and print its reverse

Source Code :

```
# Q08. Write a R program to enter a number and print its
reverse

reverse_number <- function(number) {
  reversed <- 0
  while (number > 0) {
    digit <- number %% 10
    reversed <- reversed * 10 + digit
    number <- number %/% 10
  }

  return(reversed)
}

# Take user input for the number
number <- 123456789
reversed_number <- reverse_number(number)
cat("The reverse of", number, "is:", reversed_number, "\n")
```

Program Output :

```
→ Rscript Q08.r
The reverse of 123456789 is: 987654321
```

### 3.9 Write a R program to check whether a number is palindrome or not

Source Code :

*# Q09. Write a R program to check whether a number is  
→ palindrome or not*

```
is_palindrome <- function(number) {  
  original <- number  
  reversed <- 0  
  
  while (number > 0) {  
    digit <- number %% 10  
    reversed <- reversed * 10 + digit  
    number <- number %/% 10  
  }  
  
  return(original == reversed)  
}  
  
# Take user input for the number  
number <- 6565656  
if (is_palindrome(number)) {  
  cat(number, "is a palindrome.\n")  
} else {  
  cat(number, "is not a palindrome.\n")  
}
```

Program Output :

```
→ Rscript Q09.r  
6565656 is a palindrome.
```

**3.10 Write a R program to find power of a number using for loop**

Source Code :

*# Q10. Write a R program to find power of a number using for  
→ loop*

```
calculate_power <- function(base, exponent) {  
  result <- 1  
  
  for (i in 1:exponent) {  
    result <- result * base  
  }  
}
```

```
}

return(result)
}

base <- 2
exponent <- 10
power_result <- calculate_power(base, exponent)
cat(base, "raised to the power of", exponent, "is:",
    ↪ power_result, "\n")
```

Program Output :

```
↪ Rscript Q10.r
2 raised to the power of 10 is: 1024
```

### 3.11 Write a R program to find all factors of a number

Source Code :

*# Q11. Write a R program to find all factors of a number*

```
find_factors <- function(number) {
  factors <- c()

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

  return(factors)
}

number <- 24
factor_list <- find_factors(number)
cat("The factors of", number, "are:", factor_list, "\n")
```

Program Output :

```
↪ Rscript Q11.r
The factors of 24 are: 1 2 3 4 6 8 12 24
```

### 3.12 Write a R program to calculate factorial of a number

Source Code :

*# Q12. Write a R program to calculate factorial of a number*

```
calculate_factorial <- function(number) {  
  if (number < 0) {  
    cat("Factorial is not defined for negative numbers.\n")  
    return(NULL)  
  }  
  
  factorial <- 1  
  for (i in 1:number) {  
    factorial <- factorial * i  
  }  
  
  return(factorial)  
}  
  
number <- 5  
factorial_result <- calculate_factorial(number)  
  
if (!is.null(factorial_result)) {  
  cat("The factorial of", number, "is:", factorial_result, "\n")  
}
```

Program Output :

→ **Rscript Q12.r**  
**The factorial of 5 is: 120**

### 3.13 Write a R program to find HCF (GCD) of two numbers

Source Code :

*# Q13. Write a R program to find HCF (GCD) of two numbers*

```
calculate_hcf <- function(a, b) {  
  while (b != 0) {  
    temp <- b  
    b <- a %% b
```

```
    a <- temp
  }
  return(a)
}

number1 <- 48
number2 <- 18

hcf_result <- calculate_hcf(number1, number2)
cat("The HCF of", number1, "and", number2, "is:", hcf_result,
    "\n")
```

Program Output :

```
→ Rscript Q13.r
The HCF of 48 and 18 is: 6
```

### 3.14 Write a R program to find LCM of two numbers

Source Code :

*# Q14. Write a R program to find LCM of two numbers*

```
calculate_gcd <- function(a, b) {
  while (b != 0) {
    temp <- b
    b <- a %% b
    a <- temp
  }
  return(a)
}

calculate_lcm <- function(a, b) {
  gcd <- calculate_gcd(a, b)
  lcm <- (a * b) / gcd
  return(lcm)
}

number1 <- 24
number2 <- 18
```

```
lcm_result <- calculate_lcm(number1, number2)
cat("The LCM of", number1, "and", number2, "is:", lcm_result,
    "\n")
```

Program Output :

```
-----
→ Rscript Q14.r
The LCM of 24 and 18 is: 72
```

### 3.15 Write a R program to check whether a number is Prime number or not

Source Code :

```
# Q15. Write a R program to check whether a number is Prime
→ number or not
```

```
is_prime <- function(number) {
  if (number <= 1) {
    return(FALSE)
  }

  if (number <= 3) {
    return(TRUE)
  }

  if (number %% 2 == 0 || number %% 3 == 0) {
    return(FALSE)
  }

  i <- 5
  while (i * i <= number) {
    if (number %% i == 0 || number %% (i + 2) == 0) {
      return(FALSE)
    }
    i <- i + 6
  }

  return(TRUE)
}
```



```
number <- 17
```

```
if (is_prime(number)) {  
  cat(number, "is a prime number.\n")  
} else {  
  cat(number, "is not a prime number.\n")  
}
```

Program Output :

```
→ Rscript Q15.r  
17 is a prime number.
```

### 3.16 Write a R program to check whether a number is Armstrong number or not

Source Code :

```
# Q16. Write a R program to check whether a number is Armstrong  
→ number or not
```

```
is_armstrong <- function(number) {  
  num_copy <- number  
  num_digits <- nchar(number)  
  armstrong_sum <- 0  
  
  while (num_copy > 0) {  
    digit <- num_copy %% 10  
    armstrong_sum <- armstrong_sum + digit ^ num_digits  
    num_copy <- num_copy %/% 10  
  }  
  
  return(armstrong_sum == number)  
}
```

```
number <- 153
```

```
if (is_armstrong(number)) {  
  cat(number, "is an Armstrong number.\n")  
} else {
```

```
cat(number, "is not an Armstrong number.\n")
}
```

Program Output :

→ **Rscript Q16.r**  
153 is an Armstrong number.

### 3.17 Write a R program to check whether a number is Perfect number or not

Source Code :

*# Q17. Write a R program to check whether a number is Perfect  
→ number or not*

```
is_perfect <- function(number) {
  if (number <= 0) {
    return(FALSE)
  }

  divisors_sum <- 0
  for (i in 1:(number/2)) {
    if (number %% i == 0) {
      divisors_sum <- divisors_sum + i
    }
  }

  return(divisors_sum == number)
}

number <- 28

if (is_perfect(number)) {
  cat(number, "is a Perfect number.\n")
} else {
  cat(number, "is not a Perfect number.\n")
}
```

Program Output :

→ Rscript Q17.r  
28 is a Perfect number.

### 3.18 Write a R program to print Fibonacci series up to n terms

Source Code :

```
# Q18. Write a R program to print Fibonacci series up to n  
→ terms
```

```
print_fibonacci <- function(n) {  
  if (n <= 0) {  
    cat("Please provide a positive integer value for n.\n")  
    return(NULL)  
  }  
  
  fib_series <- c(0, 1)  
  
  if (n == 1) {  
    cat(fib_series[1], "\n")  
  } else if (n == 2) {  
    cat(fib_series, "\n")  
  } else {  
    for (i in 3:n) {  
      next_term <- fib_series[i - 1] + fib_series[i - 2]  
      fib_series <- c(fib_series, next_term)  
    }  
    cat(fib_series, "\n")  
  }  
}  
  
n <- 10  
print_fibonacci(n)
```

Program Output :

→ Rscript Q18.r  
0 1 1 2 3 5 8 13 21 34