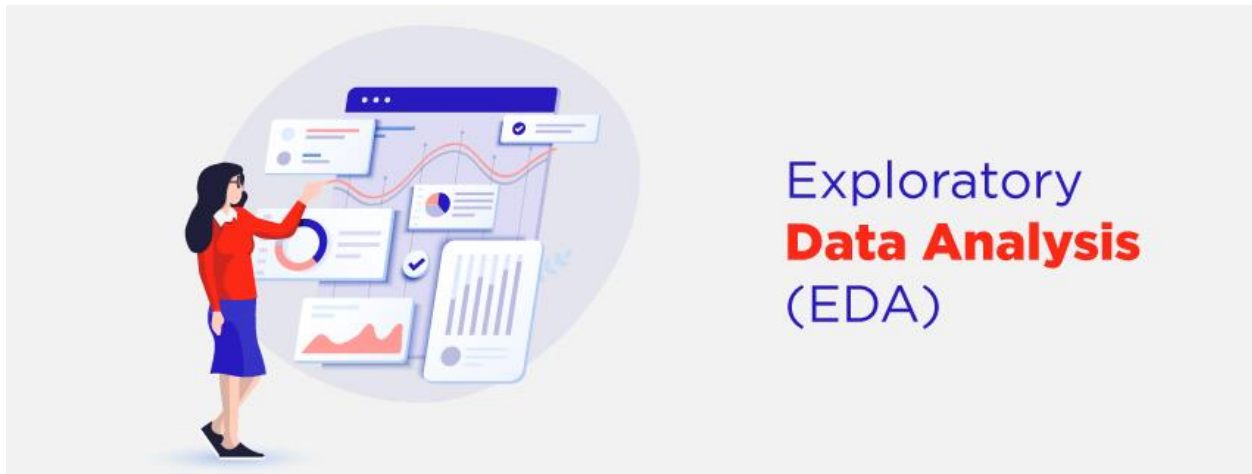


Practice Assignment 2

Lab EDA



Name : Jvn Ganesh

Roll N.O: 21BDS0085

Implementing Branching and Looping

➤ **First I am initializing all the functions and I am calling them in the end.**

1. Fibonacci

Code

```
# 1. Fibonacci Sequence
```

```
print("21BDS0085 Jvn Ganesh")
```

```
fibonacci <- function(n) {
```

```
  fib <- numeric(n)
```

```
  fib[1] <- 0
```

```
  fib[2] <- 1
```

```
  for (i in 3:n) {
```

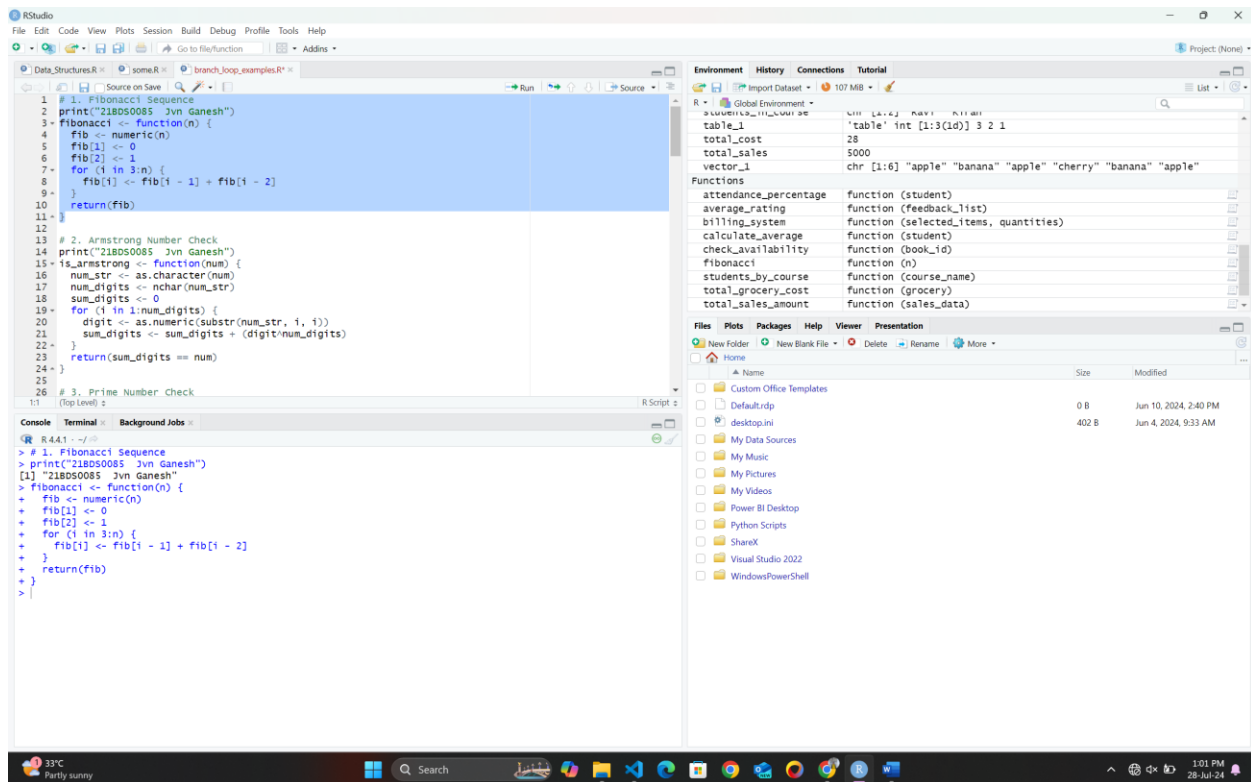
```
    fib[i] <- fib[i - 1] + fib[i - 2]
```

```
  }
```

```
  return(fib)
```

```
}
```

Ouptut



2. Armstrong Number Check

Code

2. Armstrong Number Check

```
print("21BDS0085 Jvn Ganesh")
```

```
is_armstrong <- function(num) {
```

```
  num_str <- as.character(num)
```

```
  num_digits <- nchar(num_str)
```

```
  sum_digits <- 0
```

```
  for (i in 1:num_digits) {
```

```
    digit <- as.numeric(substr(num_str, i, i))
```

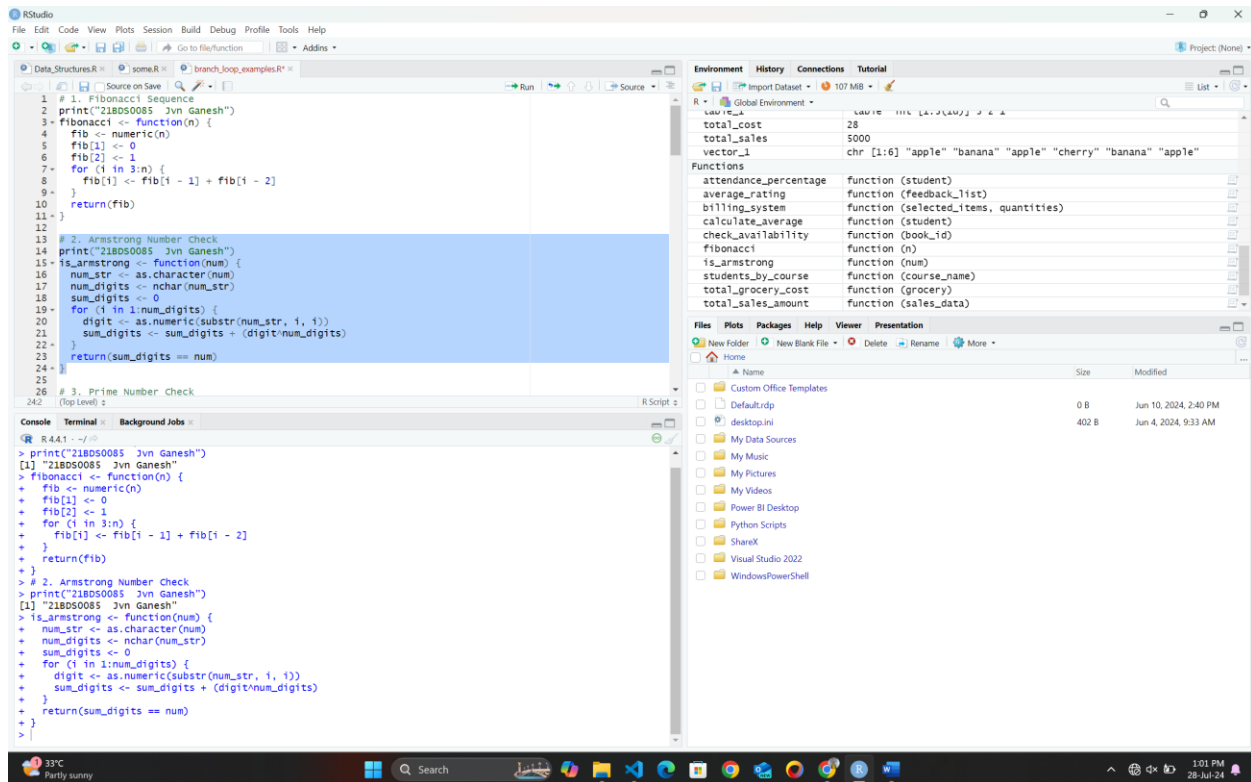
```
    sum_digits <- sum_digits + (digit^num_digits)
```

```
  }
```

```
  return(sum_digits == num)
```

```
}
```

Output



3. Prime Number Check

Code

```
# 3. Prime Number Check
```

```
print("21BDS0085 Jvn Ganesh")
```

```
is_prime <- function(num) {
```

```
  if (num <= 1) return(FALSE)
```

```
  for (i in 2:sqrt(num)) {
```

```
    if (num %% i == 0) return(FALSE)
```

```
  }
```

```
  return(TRUE)
```

```
}
```

Output

The screenshot shows the RStudio interface with a script editor on the left, a console at the bottom, and an environment pane on the right. The script editor contains three functions: `is_armstrong`, `is_prime`, and `factorial`. The console shows the output of these functions when called with the number 218050085. The environment pane shows the global environment with variables like `total_sales` and `vector_1`.

```
# 2. Armstrong Number Check
print("218050085 Jvn Ganesh")
is_armstrong <- function(num) {
  num_str <- as.character(num)
  num_digits <- nchar(num_str)
  sum_digits <- 0
  for (i in 1:num_digits) {
    digit <- as.numeric(substr(num_str, i, i))
    sum_digits <- sum_digits + (digit^num_digits)
  }
  return(sum_digits == num)
}

# 3. Prime Number Check
print("218050085 Jvn Ganesh")
is_prime <- function(num) {
  if (num <= 1) return(FALSE)
  for (i in 2:sqrt(num)) {
    if (num % i == 0) return(FALSE)
  }
  return(TRUE)
}

# 4. Factorial Calculation
print("218050085 Jvn Ganesh")
factorial <- function(n) {
  result <- 1
  for (i in 2:n) {
    result <- result * i
  }
  return(result)
}
```

Console Output:

```
R 4.4.1 ~ -/~/
> # 2. Armstrong Number Check
> print("218050085 Jvn Ganesh")
[1] "218050085 Jvn Ganesh"
> is_armstrong <- function(num) {
+   num_str <- as.character(num)
+   num_digits <- nchar(num_str)
+   sum_digits <- 0
+   for (i in 1:num_digits) {
+     digit <- as.numeric(substr(num_str, i, i))
+     sum_digits <- sum_digits + (digit^num_digits)
+   }
+   return(sum_digits == num)
+ }
> # 3. Prime Number Check
> print("218050085 Jvn Ganesh")
[1] "218050085 Jvn Ganesh"
> is_prime <- function(num) {
+   if (num <= 1) return(FALSE)
+   for (i in 2:sqrt(num)) {
+     if (num % i == 0) return(FALSE)
+   }
+   return(TRUE)
+ }
> # 4. Factorial Calculation
> print("218050085 Jvn Ganesh")
[1] "218050085 Jvn Ganesh"
> factorial <- function(n) {
+   result <- 1
+   for (i in 2:n) {
+     result <- result * i
+   }
+   return(result)
+ }
>
```

4. Factorial Calculation

4. Factorial Calculation

```
print("218050085 Jvn Ganesh")
```

```
factorial <- function(n) {
```

```
  result <- 1
```

```
  for (i in 2:n) {
```

```
    result <- result * i
```

```
  }
```

```
  return(result)
```

```
}
```

Output

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
+   sum_digits <- sum_digits + (digit^num_digits)
+ }
+ return(sum_digits == num)
+ }
> # 3. Prime Number Check
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> is_prime <- function(num) {
+   if (num <= 1) return(FALSE)
+   for (i in 2:sqrt(num)) {
+     if (num %% i == 0) return(FALSE)
+   }
+   return(TRUE)
+ }
> # 4. Factorial Calculation
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> factorial <- function(n) {
+   result <- 1
+   for (i in 2:n) {
+     result <- result * i
+   }
+   return(result)
+ }
> |
```

5. Palindrome Checking

Code

5. Palindrome Check

```
print("21BDS0085 Jvn Ganesh")

is_palindrome <- function(num) {

  num_str <- as.character(num)

  return(num_str == paste(rev(strsplit(num_str, NULL)[[1]]), collapse = ""))

}
```

Output

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> is_prime <- function(num) {
+   if (num <= 1) return(FALSE)
+   for (i in 2:sqrt(num)) {
+     if (num %% i == 0) return(FALSE)
+   }
+   return(TRUE)
+ }
> # 4. Factorial Calculation
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> factorial <- function(n) {
+   result <- 1
+   for (i in 2:n) {
+     result <- result * i
+   }
+   return(result)
+ }
> # 5. Palindrome Check
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> is_palindrome <- function(num) {
+   num_str <- as.character(num)
+   return(num_str == paste(rev(strsplit(num_str, NULL)[[1]]), collapse = ""))
+ }
> |
```

6. Sum if digits

Code

6. Sum of Digits

```
print("21BDS0085 Jvn Ganesh")

sum_of_digits <- function(num) {

  num_str <- as.character(num)

  sum_digits <- 0

  for (i in 1:nchar(num_str)) {

    sum_digits <- sum_digits + as.numeric(substr(num_str, i, i))

  }

  return(sum_digits)

}
```

Output

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
+ result <- 1
+ for (i in 2:n) {
+   result <- result * i
+ }
+ return(result)
+ }
> # 5. Palindrome Check
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> is_palindrome <- function(num) {
+   num_str <- as.character(num)
+   return(num_str == paste(rev(strsplit(num_str, NULL)[[1]]), collapse = ""))
+ }
> # 6. Sum of Digits
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> sum_of_digits <- function(num) {
+   num_str <- as.character(num)
+   sum_digits <- 0
+   for (i in 1:nchar(num_str)) {
+     sum_digits <- sum_digits + as.numeric(substr(num_str, i, i))
+   }
+   return(sum_digits)
+ }
> |
```

7. Reverse a Number

Code

```
# 7. Reverse a Number
```

```
print("21BDS0085 Jvn Ganesh")
```

```
reverse_number <- function(num) {
```

```
  num_str <- as.character(num)
```

```
  reversed_str <- paste(rev(strsplit(num_str, NULL)[[1]]), collapse = "")
```

```
  return(as.numeric(reversed_str))
```

```
}
```

Output


```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
[1] "21BDS0085 Jvn Ganesh"
> is_palindrome <- function(num) {
+   num_str <- as.character(num)
+   return(num_str == paste(rev(strsplit(num_str, NULL)[[1]]), collapse = ""))
+ }
> # 6. Sum of Digits
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> sum_of_digits <- function(num) {
+   num_str <- as.character(num)
+   sum_digits <- 0
+   for (i in 1:nchar(num_str)) {
+     sum_digits <- sum_digits + as.numeric(substr(num_str, i, i))
+   }
+   return(sum_digits)
+ }
> # 7. Reverse a Number
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> reverse_number <- function(num) {
+   num_str <- as.character(num)
+   reversed_str <- paste(rev(strsplit(num_str, NULL)[[1]]), collapse = "")
+   return(as.numeric(reversed_str))
+ }
> |
```

8. Perfect Number Check

Code

```
# 8. Perfect Number Check
```

```
print("21BDS0085 Jvn Ganesh")
```

```
is_perfect <- function(num) {
```

```
  sum_divisors <- 0
```

```
  for (i in 1:(num - 1)) {
```

```
    if (num %% i == 0) sum_divisors <- sum_divisors + i
```

```
  }
```

```
  return(sum_divisors == num)
```

Output

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
+ num_str <- as.character(num)
+ sum_digits <- 0
+ for (i in 1:nchar(num_str)) {
+   sum_digits <- sum_digits + as.numeric(substr(num_str, i, i))
+ }
+ return(sum_digits)
+ }
> # 7. Reverse a Number
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> reverse_number <- function(num) {
+   num_str <- as.character(num)
+   reversed_str <- paste(rev(strsplit(num_str, NULL)[[1]]), collapse = "")
+   return(as.numeric(reversed_str))
+ }
> # 8. Perfect Number Check
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> is_perfect <- function(num) {
+   sum_divisors <- 0
+   for (i in 1:(num - 1)) {
+     if (num %% i == 0) sum_divisors <- sum_divisors + i
+   }
+   return(sum_divisors == num)
+ }
```

9. Sum of First N Natural Numbers

Code

```
# 9. Sum of First N Natural Numbers
```

```
print("21BDS0085 Jvn Ganesh")
```

```
sum_natural_numbers <- function(n) {
```

```
  sum <- 0
```

```
  for (i in 1:n) {
```

```
    sum <- sum + i
```

```
  }
```

```
  return(sum)
```

```
}
```

Output

```
Console | Terminal x | Background Jobs x
R 4.4.1 · ~/
> is_perfect <- function(num) {
+   sum_divisors <- 0
+   for (i in 1:(num - 1)) {
+     if (num %% i == 0) sum_divisors <- sum_divisors + i
+   }
+   return(sum_divisors == num)
+ # 9. Sum of First N Natural Numbers
+ print("21BDS0085 Jvn Ganesh")
+ sum_natural_numbers <- function(n) {
+   sum <- 0
+   for (i in 1:n) {
+     sum <- sum + i
+   }
+   return(sum)
+ }
+ # 9. Sum of First N Natural Numbers
+ print("21BDS0085 Jvn Ganesh")
+ sum_natural_numbers <- function(n) {
+   sum <- 0
+   for (i in 1:n) {
+     sum <- sum + i
+   }
+   return(sum)
+ }
+ }
```

10. GCD of Two Numbers

Code

10. GCD of Two Numbers

```
print("21BDS0085 Jvn Ganesh")
```

```
gcd <- function(a, b){
```

```
  while (b != 0){
```

```
    temp <- b
```

```
    b <- a %% b
```

```
    a <- temp
```

```
  }
```

```
  return(a)
```

```
}
```

Output

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
+   for (i in 1:n) {
+     sum <- sum + i
+   }
+   return(sum)
+ }
+ # 9. Sum of First N Natural Numbers
+ print("21BDS0085 Jvn Ganesh")
+ sum_natural_numbers <- function(n) {
+   sum <- 0
+   for (i in 1:n) {
+     sum <- sum + i
+   }
+   return(sum)
+ }
+ # 10. GCD of Two Numbers
+ print("21BDS0085 Jvn Ganesh")
+ gcd <- function(a, b) {
+   while (b != 0) {
+     temp <- b
+     b <- a %% b
+     a <- temp
+   }
+   return(a)
+ }
+
+
```

➤ Now I will call all the functions I used one by one.

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> num_terms <- 10
> cat("Fibonacci sequence with", num_terms, "terms:\n", fibonacci(num_terms), "\n\n")
Fibonacci sequence with 10 terms:
0 1 1 2 3 5 8 13 21 34
> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> num_terms <- 10
> cat("Fibonacci sequence with", num_terms, "terms:\n", fibonacci(num_terms), "\n\n")
Fibonacci sequence with 10 terms:
0 1 1 2 3 5 8 13 21 34

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 153
> cat(number, "is an Armstrong number:", is_armstrong(number), "\n\n")
153 is an Armstrong number: TRUE

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> num_terms <- 10
> cat("Fibonacci sequence with", num_terms, "terms:\n", fibonacci(num_terms), "\n\n")
Fibonacci sequence with 10 terms:
0 1 1 2 3 5 8 13 21 34

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 153
> cat(number, "is an Armstrong number:", is_armstrong(number), "\n\n")
153 is an Armstrong number: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 29
> cat(number, "is a Prime number:", is_prime(number), "\n\n")
29 is a Prime number: TRUE

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
[1] "21BDS0085 Jvn Ganesh"
> num_terms <- 10
> cat("Fibonacci sequence with", num_terms, "terms:\n", fibonacci(num_terms), "\n\n")
Fibonacci sequence with 10 terms:
0 1 1 2 3 5 8 13 21 34

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 153
> cat(number, "is an Armstrong number:", is_armstrong(number), "\n\n")
153 is an Armstrong number: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 29
> cat(number, "is a Prime number:", is_prime(number), "\n\n")
29 is a Prime number: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 5
> cat("Factorial of", number, "is", factorial(number), "\n\n")
Factorial of 5 is 120

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 153
> cat(number, "is an Armstrong number:", is_armstrong(number), "\n\n")
153 is an Armstrong number: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 29
> cat(number, "is a Prime number:", is_prime(number), "\n\n")
29 is a Prime number: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 5
> cat("Factorial of", number, "is", factorial(number), "\n\n")
Factorial of 5 is 120

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 121
> cat(number, "is a Palindrome:", is_palindrome(number), "\n\n")
121 is a Palindrome: TRUE

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 29
> cat(number, "is a Prime number:", is_prime(number), "\n\n")
29 is a Prime number: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 5
> cat("Factorial of", number, "is", factorial(number), "\n\n")
Factorial of 5 is 120

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 121
> cat(number, "is a Palindrome:", is_palindrome(number), "\n\n")
121 is a Palindrome: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 123
> cat("Sum of digits of", number, "is", sum_of_digits(number), "\n\n")
Sum of digits of 123 is 6

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 5
> cat("Factorial of", number, "is", factorial(number), "\n\n")
Factorial of 5 is 120

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 121
> cat(number, "is a Palindrome:", is_palindrome(number), "\n\n")
121 is a Palindrome: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 123
> cat("Sum of digits of", number, "is", sum_of_digits(number), "\n\n")
Sum of digits of 123 is 6

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 1234
> cat("Reversed number of", number, "is", reverse_number(number), "\n\n")
Reversed number of 1234 is 4321

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 121
> cat(number, "is a Palindrome:", is_palindrome(number), "\n\n")
121 is a Palindrome: TRUE

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 123
> cat("Sum of digits of", number, "is", sum_of_digits(number), "\n\n")
Sum of digits of 123 is 6

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 1234
> cat("Reversed number of", number, "is", reverse_number(number), "\n\n")
Reversed number of 1234 is 4321

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 28
> cat(number, "is a Perfect number:", is_perfect(number), "\n\n")
28 is a Perfect number: TRUE

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
> # 9. Sum of First N Natural Numbers
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> sum_natural_numbers <- function(n) {
+   sum <- 0
+   for (i in 1:n) {
+     sum <- sum + i
+   }
+   return(sum)
+ }
> # 10. GCD of Two Numbers
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> gcd <- function(a, b) {
+   while (b != 0) {
+     temp <- b
+     b <- a %% b
+     a <- temp
+   }
+   return(a)
+ }
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 10
> cat("Sum of first", number, "natural numbers is", sum_natural_numbers(number), "\n\n")
Sum of first 10 natural numbers is 55

> |
```

```
Console Terminal x Background Jobs x
R 4.4.1 · ~/
+   }
+   return(sum)
+ }
> # 10. GCD of Two Numbers
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> gcd <- function(a, b) {
+   while (b != 0) {
+     temp <- b
+     b <- a %% b
+     a <- temp
+   }
+   return(a)
+ }
> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> number <- 10
> cat("Sum of first", number, "natural numbers is", sum_natural_numbers(number), "\n\n")
Sum of first 10 natural numbers is 55

> print("21BDS0085 Jvn Ganesh")
[1] "21BDS0085 Jvn Ganesh"
> a <- 48
> b <- 18
> cat("GCD of", a, "and", b, "is", gcd(a, b), "\n\n")
GCD of 48 and 18 is 6

> |
```


FINAL CODES

EXP-1

Code

```
print("21BDS0085 JVNGANESH")
```

#10

```
print("21BDS0085 JVNGANESH")
```

```
data <- data.frame(
```

```
  ID = 1:5,
```

```
  Name = c("suresh", "ramesh", "harish", "girish", "hari"),
```

```
  Score = c(85, 92, 88, 90, 95)
```

```
)
```

```
print("Original Data Frame:")
```

```
print(data)
```

```
data$Grade <- ifelse(data$Score > 90, "A", "B")
```

```
print("Data Frame with New Column 'Grade':")
```

```
print(data)
```

```
subset_data <- data[data$Score > 90, ]
```

```
print("Subset of Data Frame (Score > 90):")
```

```
print(subset_data)
```

#9

```
print("21BDS0085 JVNGANESH")

nested_list <- list(

  name = "Ganesh",

  age = 30,

  address = list(street = "123 Main St", city = "Anytown", zip = "52156")

)

print("Nested List:")

print(nested_list)
```

```
print("Nested List Element 'address$city':")

print(nested_list$address$city)
```

#8

```
print("21BDS0085 JVNGANESH")

complex_list <- list(

  vector = c(1, 2, 3),

  matrix = matrix(1:4, nrow = 2),

  data_frame = data.frame(A = 1:3, B = c("X", "Y", "Z"))

)

print("Complex List:")

print(complex_list)
```

#7

```
print("21BDS0085 JVNGANESH")  
vector_1 <- c("apple", "banana", "apple", "cherry", "banana", "apple")  
table_1 <- table(vector_1)  
print("Table:")  
print(table_1)
```

#6

```
print("21BDS0085 JVNGANESH")  
factor_1 <- factor(c("low", "medium", "high", "medium", "low"))  
print("Factor:")  
print(factor_1)
```

```
print("Levels:")  
print(levels(factor_1))
```

#5

```
print("21BDS0085 JVNGANESH")  
data_frame_1 <- data.frame(  
  Name = c("Pranay", "gary", "jayanth"),
```

```
Age = c(28, 22, 35),  
Score = c(85, 90, 80)  
)  
print("Data Frame:")  
print(data_frame_1)  
  
print("Column 'Name':")  
print(data_frame_1$Name)
```

#4

```
list_1 <- list(name = "Ganesh", age = 22, scores = c(85, 90, 95))  
print("List:")  
print(list_1)  
  
print("List Element 'name':")  
print(list_1$name)
```

#3

```
array_1 <- array(1:24, dim = c(3, 4, 2))  
print("3D Array:")
```

```
print(array_1)
```

```
#2
```

```
matrix_1 <- matrix(1:9, nrow = 3, ncol = 3)
```

```
print("Matrix 1:")
```

```
print(matrix_1)
```

```
matrix_2 <- t(matrix_1)
```

```
print("Transposed Matrix:")
```

```
print(matrix_2)
```

```
#A1)
```

```
numeric_vector <- c(11, 32, 43, 64, 75)
```

```
print("Numeric Vector:")
```

```
print(numeric_vector)
```

```
char_vector <- c("ganesh", "ramesh", "suresh")
```

```
print("Character Vector:")
```

```
print(char_vector)
```

Code for examples:

```
# 1. Billing System
```

```
print("21BDS0085 Jvn Ganesh")
```

```
items <- data.frame(
  Item = c("Rice", "Wheat", "Dal", "Oil"),
  Price = c(40, 25, 60, 120)
)

billing_system <- function(selected_items, quantities) {
  total_cost <- 0
  for (i in 1:length(selected_items)) {
    item <- selected_items[i]
    quantity <- quantities[i]
    price <- items[items$Item == item, "Price"]
    total_cost <- total_cost + (price * quantity)
  }
  return(total_cost)
}

selected_items <- c("Rice", "Wheat", "Dal")
quantities <- c(2, 1, 1)
total_cost <- billing_system(selected_items, quantities)
print(paste("Total Cost:", total_cost))
```

2. Class Marks System

```
print("21BDS0085 Jvn Ganesh")

students <- data.frame(
  Name = c("Ravi", "Priya", "Kiran"),
  Math = c(90, 80, 85),
  Science = c(95, 75, 88),
  English = c(88, 92, 85)
)
```

```
calculate_average <- function(student) {  
  subjects <- c("Math", "Science", "English")  
  avg_marks <- mean(unlist(students[students$Name == student, subjects]))  
  return(avg_marks)  
}  
  
student_name <- "Ravi"  
  
average_marks <- calculate_average(student_name)  
  
print(paste("Average Marks of", student_name, ":", average_marks))
```

3. Employee Records

```
print("21BDS0085 Jvn Ganesh")  
  
employees <- list(  
  list(Name = "Raj", Age = 30, Department = "HR"),  
  list(Name = "Sita", Age = 25, Department = "Finance"),  
  list(Name = "Amit", Age = 35, Department = "IT")  
)  
  
find_employee_by_department <- function(department) {  
  result <- lapply(employees, function(emp) {  
    if (emp$Department == department) return(emp)  
  })  
  return(Filter(Negate(is.null), result))  
}  
  
department <- "IT"  
  
employees_in_dept <- find_employee_by_department(department)  
  
print(paste("Employees in", department, "Department:"))  
  
print(employees_in_dept)
```

4. Inventory Management System

```
print("21BDS0085 Jvn Ganesh")

inventory <- data.frame(

  Item = c("Laptop", "Mouse", "Keyboard", "Monitor"),

  Quantity = c(10, 50, 30, 20)

)

update_inventory <- function(item, quantity) {

  inventory[inventory$Item == item, "Quantity"] <- quantity

  return(inventory)

}

updated_inventory <- update_inventory("Laptop", 8)

print("Updated Inventory:")

print(updated_inventory)
```

5. Library Management System

```
print("21BDS0085 Jvn Ganesh")

books <- data.frame(

  BookID = 1:4,

  Title = c("Book A", "Book B", "Book C", "Book D"),

  Author = c("Author A", "Author B", "Author C", "Author D"),

  Available = c(TRUE, FALSE, TRUE, TRUE)

)

check_availability <- function(book_id) {

  available <- books[books$BookID == book_id, "Available"]

  return(available)

}

book_id <- 2
```



```
availability <- check_availability(book_id)

print(paste("Is Book ID", book_id, "available?", availability))
```

6. Student Attendance System

```
print("21BDS0085 Jvn Ganesh")

attendance <- matrix(c(1, 0, 1, 1, 1, 1, 0, 0), nrow = 4, byrow = TRUE)

colnames attendance) <- c("Day1", "Day2")

rownames attendance) <- c("Ravi", "Priya", "Kiran", "Mohan")

attendance_percentage <- function(student) {

  total_days <- ncol attendance)

  present_days <- sum attendance student, ])

  percentage <- (present_days / total_days) * 100

  return percentage)

}

student_name <- "Ravi"

attendance_pct <- attendance_percentage(student_name)

print(paste("Attendance Percentage of", student_name, ":", attendance_pct, "%"))
```

7. Customer Feedback System

```
print("21BDS0085 Jvn Ganesh")

feedback <- list(

  list(Customer = "Ravi", Rating = 4, Comment = "Good service"),

  list(Customer = "Priya", Rating = 5, Comment = "Excellent!"),

  list(Customer = "Kiran", Rating = 3, Comment = "Average experience")

)

average_rating <- function(feedback_list) {

  ratings <- sapply(feedback_list, function(fb) fb$Rating)
```

```
    avg_rating <- mean(ratings)
    return(avg_rating)
}

avg_rating <- average_rating(feedback)

print(paste("Average Customer Rating:", avg_rating))
```

8. Sales Records

```
print("21BDS0085 Jvn Ganesh")

sales <- data.frame(
  SalesID = 1:4,
  Product = c("Product A", "Product B", "Product C", "Product D"),
  Quantity = c(10, 5, 8, 6),
  Price = c(100, 200, 150, 300)
)

total_sales_amount <- function(sales_data) {
  total_amount <- sum(sales_data$Quantity * sales_data$Price)
  return(total_amount)
}

total_sales <- total_sales_amount(sales)

print(paste("Total Sales Amount:", total_sales))
```

9. Grocery List System

```
print("21BDS0085 Jvn Ganesh")

grocery_list <- data.frame(
  Item = c("Apple", "Banana", "Carrot", "Dates"),
  Quantity = c(5, 10, 4, 2),
  PricePerUnit = c(2, 1, 0.5, 3)
```

```

)

total_grocery_cost <- function(grocery) {
  total_cost <- sum(grocery$Quantity * grocery$PricePerUnit)
  return(total_cost)
}

total_cost <- total_grocery_cost(grocery_list)

print(paste("Total Grocery Cost:", total_cost))

```

10. University Enrollment System

```

print("21BDS0085 Jvn Ganesh")

enrollment <- data.frame(
  StudentID = 1:4,
  Name = c("Ravi", "Priya", "Kiran", "Mohan"),
  Course = c("Math", "Science", "Math", "History"),
  Status = c("Enrolled", "Enrolled", "Waitlisted", "Enrolled")
)

students_by_course <- function(course_name) {
  students <- enrollment[enrollment$Course == course_name, "Name"]
  return(students)
}

course_name <- "Math"

students_in_course <- students_by_course(course_name)

print(paste("Students enrolled in", course_name, "course:"))

print(students_in_course)

```

Code for exp-2

1. Fibonacci Sequence

```
print("21BDS0085 Jvn Ganesh")
```

```
fibonacci <- function(n) {
```

```
  fib <- numeric(n)
```

```
  fib[1] <- 0
```

```
  fib[2] <- 1
```

```
  for (i in 3:n) {
```

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

```
  }
```

```
  return(fib)
```

```
}
```

```
# 2. Armstrong Number Check
```

```
print("21BDS0085 Jvn Ganesh")
```

```
is_armstrong <- function(num) {
```

```
  num_str <- as.character(num)
```

```
  num_digits <- nchar(num_str)
```

```
  sum_digits <- 0
```

```
  for (i in 1:num_digits) {
```

```
    digit <- as.numeric(substr(num_str, i, i))
```

```
    sum_digits <- sum_digits + (digit^num_digits)
```

```
  }
```

```
  return(sum_digits == num)
```

```
}
```

```
# 3. Prime Number Check
```

```
print("21BDS0085 Jvn Ganesh")
```

```
is_prime <- function(num) {
```

```
if (num <= 1) return(FALSE)
for (i in 2:sqrt(num)) {
  if (num %% i == 0) return(FALSE)
}
return(TRUE)
}
```

4. Factorial Calculation

```
print("21BDS0085 Jvn Ganesh")
factorial <- function(n) {
  result <- 1
  for (i in 2:n) {
    result <- result * i
  }
  return(result)
}
```

5. Palindrome Check

```
print("21BDS0085 Jvn Ganesh")
is_palindrome <- function(num) {
  num_str <- as.character(num)
  return(num_str == paste(rev(strsplit(num_str, NULL)[[1]]), collapse = ""))
}
```

6. Sum of Digits

```
print("21BDS0085 Jvn Ganesh")
sum_of_digits <- function(num) {
```

```
num_str <- as.character(num)
sum_digits <- 0
for (i in 1:nchar(num_str)) {
  sum_digits <- sum_digits + as.numeric(substr(num_str, i, i))
}
return(sum_digits)
}
```

7. Reverse a Number

```
print("21BDS0085 Jvn Ganesh")
reverse_number <- function(num) {
  num_str <- as.character(num)
  reversed_str <- paste(rev(strsplit(num_str, NULL)[[1]]), collapse = "")
  return(as.numeric(reversed_str))
}
```

8. Perfect Number Check

```
print("21BDS0085 Jvn Ganesh")
is_perfect <- function(num) {
  sum_divisors <- 0
  for (i in 1:(num - 1)) {
    if (num %% i == 0) sum_divisors <- sum_divisors + i
  }
  return(sum_divisors == num)
}
```

9. Sum of First N Natural Numbers

```
print("21BDS0085 Jvn Ganesh")

sum_natural_numbers <- function(n) {

  sum <- 0

  for (i in 1:n){

    sum <- sum + i

  }

  return(sum)

}
```

10. GCD of Two Numbers

```
print("21BDS0085 Jvn Ganesh")

gcd <- function(a, b){

  while (b != 0){

    temp <- b

    b <- a %% b

    a <- temp

  }

  return(a)

}
```

Example usage of all functions

```
print("21BDS0085 Jvn Ganesh")

num_terms <- 10

cat("Fibonacci sequence with", num_terms, "terms:\n", fibonacci(num_terms), "\n\n")

print("21BDS0085 Jvn Ganesh")

number <- 153
```

```
cat(number, "is an Armstrong number:", is_armstrong(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
number <- 29
```

```
cat(number, "is a Prime number:", is_prime(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
number <- 5
```

```
cat("Factorial of", number, "is", factorial(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
number <- 121
```

```
cat(number, "is a Palindrome:", is_palindrome(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
number <- 123
```

```
cat("Sum of digits of", number, "is", sum_of_digits(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
number <- 1234
```

```
cat("Reversed number of", number, "is", reverse_number(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
number <- 28
```

```
cat(number, "is a Perfect number:", is_perfect(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```



```
number <- 10
```

```
cat("Sum of first", number, "natural numbers is", sum_natural_numbers(number), "\n\n")
```

```
print("21BDS0085 Jvn Ganesh")
```

```
a <- 48
```

```
b <- 18
```

```
cat("GCD of", a, "and", b, "is", gcd(a, b), "\n\n")
```