Practice Assignment 1

Lab EDA

(Both Exp 1 and Exp 2 are here)

Name : Jvn Ganesh

Roll N.O: 21BDS0085

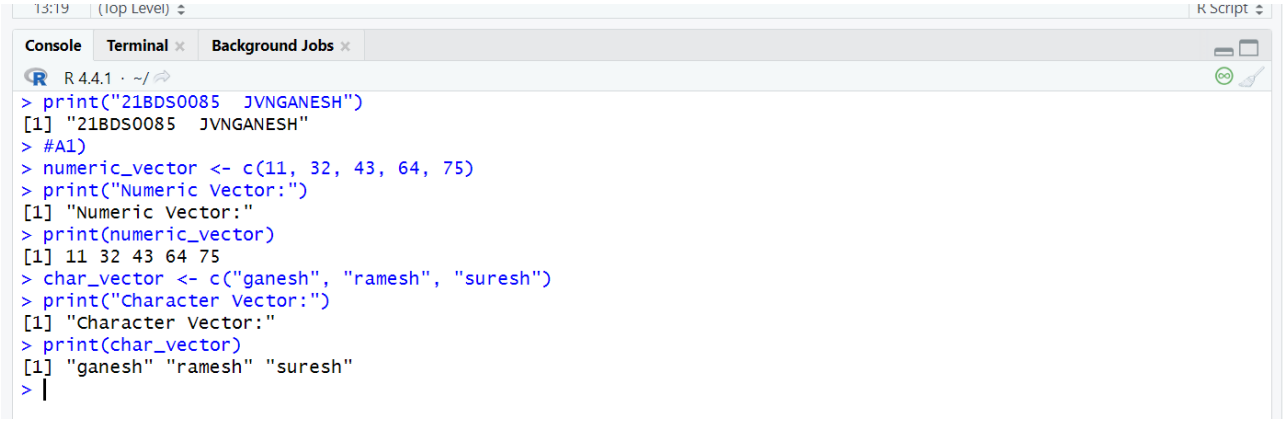
BASIC DATA STRUCTURES

1. Vectors

Code :

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

Output:



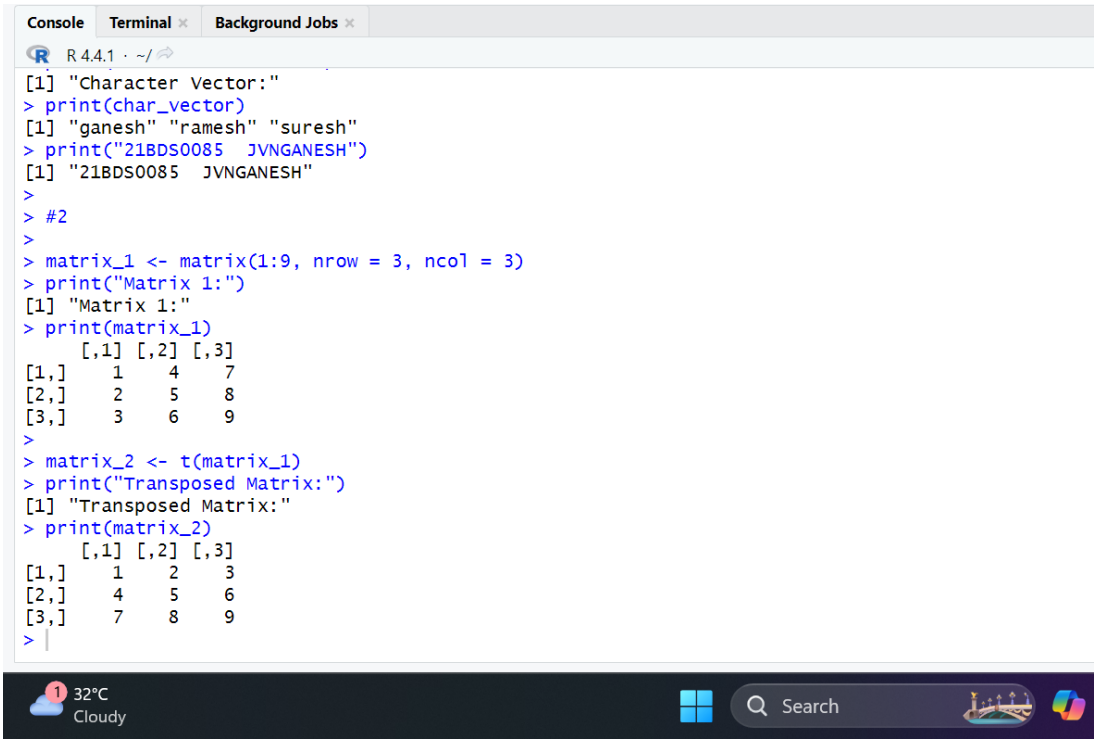
2. Matrices

Code:

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

OUTPUT:



3.ARRAYS

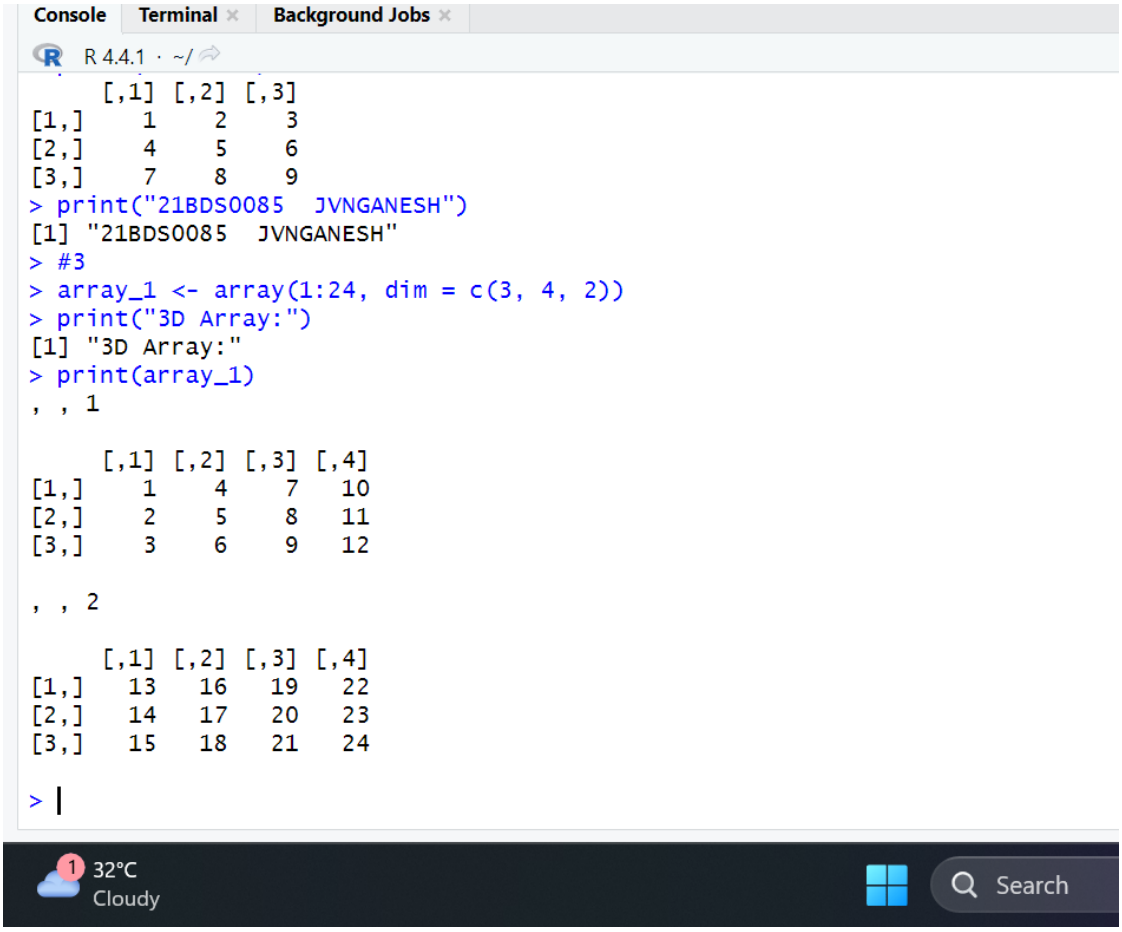
Code

```
print("21BDS0085 JVNGANESH")

#3

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

print("3D Array:")

print(array_1)

Output



4.Lists

Code

print("21BDS0085 JVNGANESH")

#4

list_1 <- list(name = "Ganesh", age = 22, scores = c(85, 90, 95))

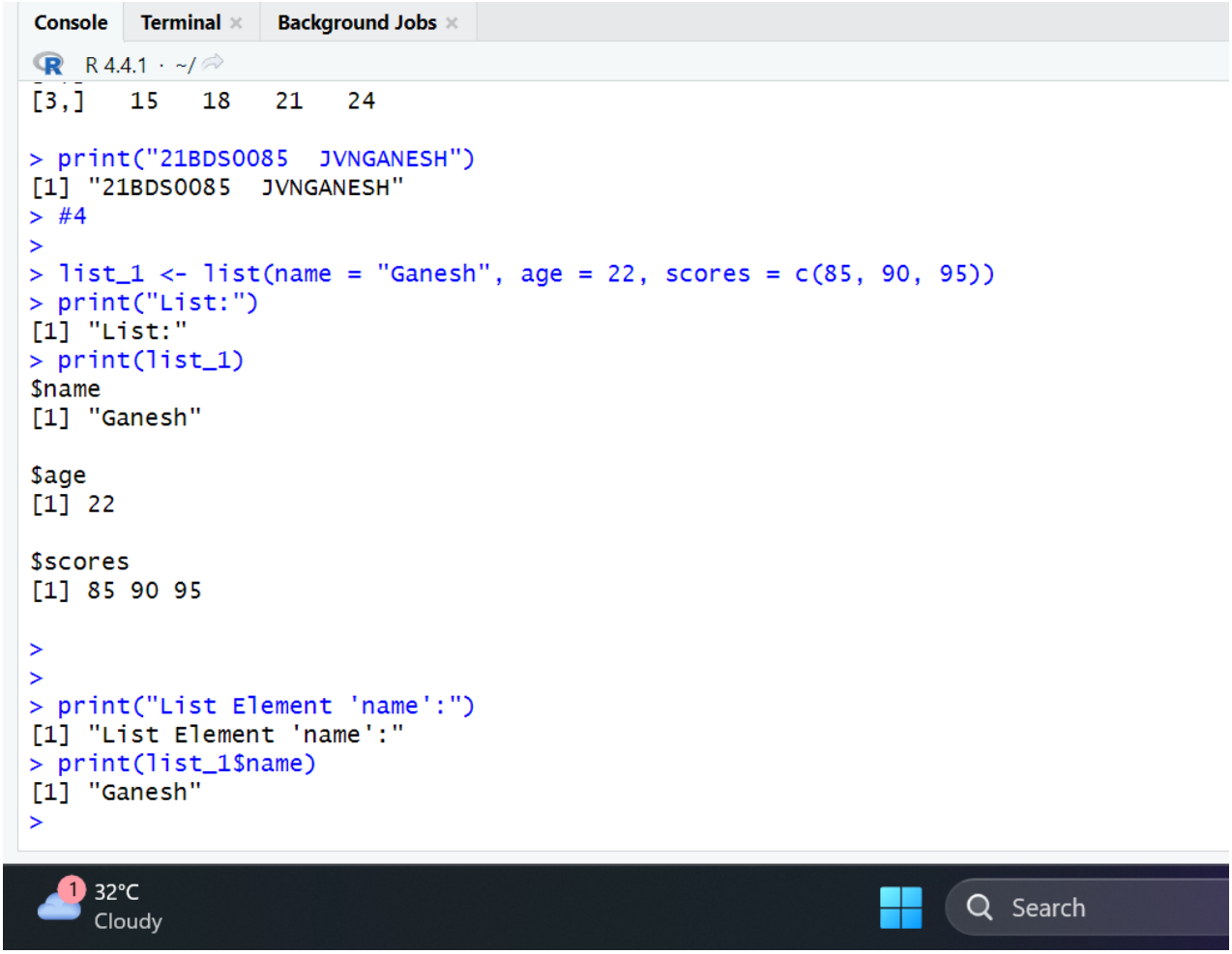
print("List:")

print(list_1)

print("List Element 'name':")

print(list_1$name)

Output



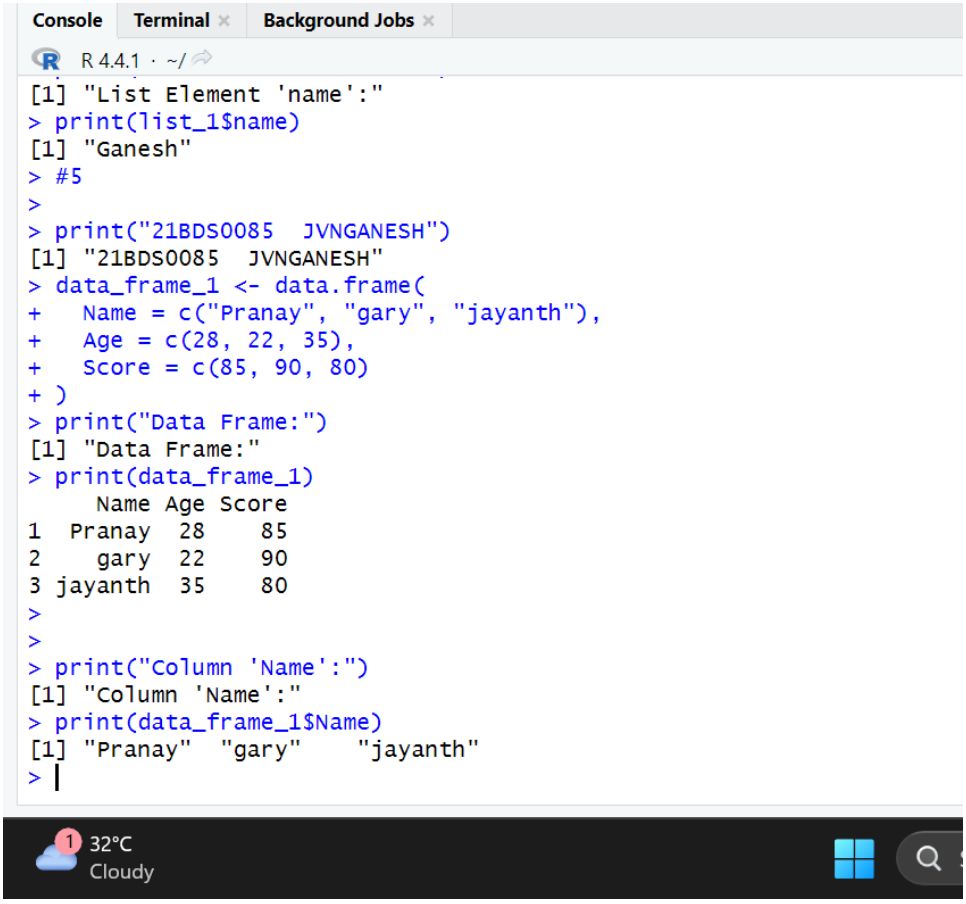
5. Data frames

Code

#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)
```

Output



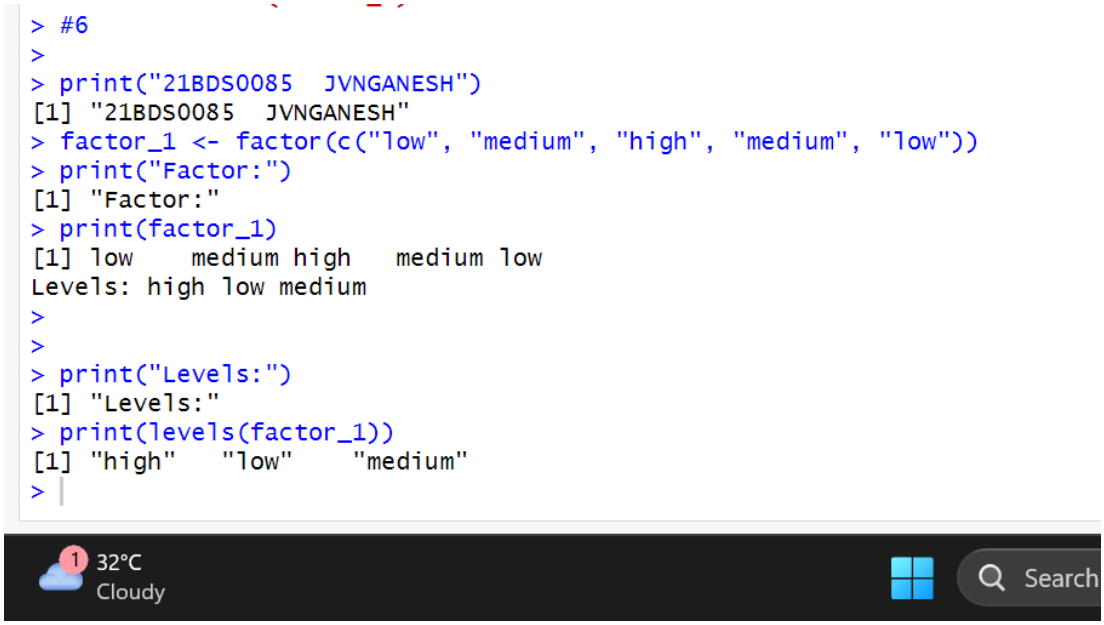
6.Factors

Code

#6

```
print("21BDS0085 JVNGANESH")

factor_1 <- factor(c("low", "medium", "high", "medium", "low"))

print("Factor:")

print(factor_1)



print("Levels:")

print(levels(factor_1))
```

Output:



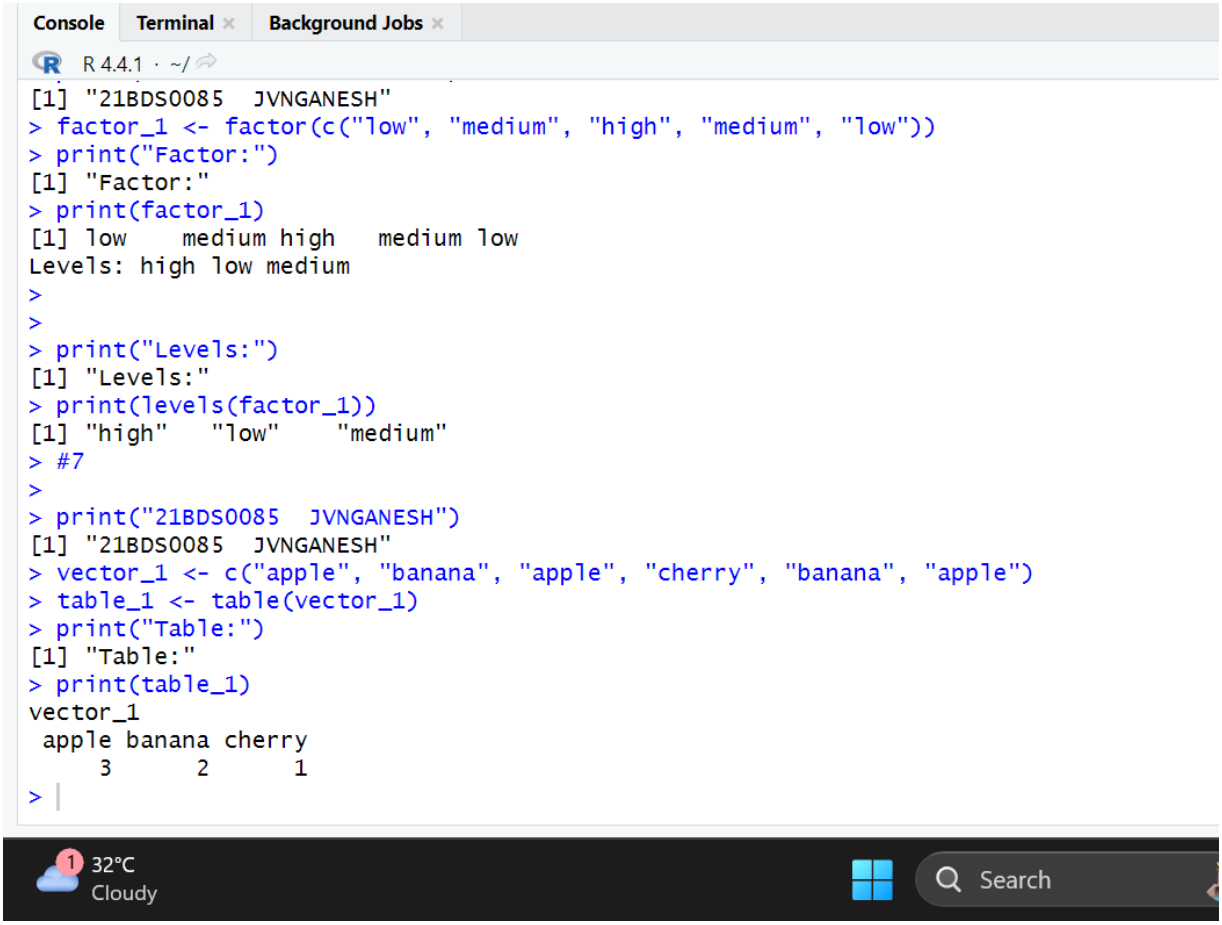
7.Tables

Code

#7

```
print("21BDS0085  JVNGANESH")

vector_1 <- c("apple", "banana", "apple", "cherry", "banana", "apple")

table_1 <- table(vector_1)

print("Table:")

print(table_1)
```

Output



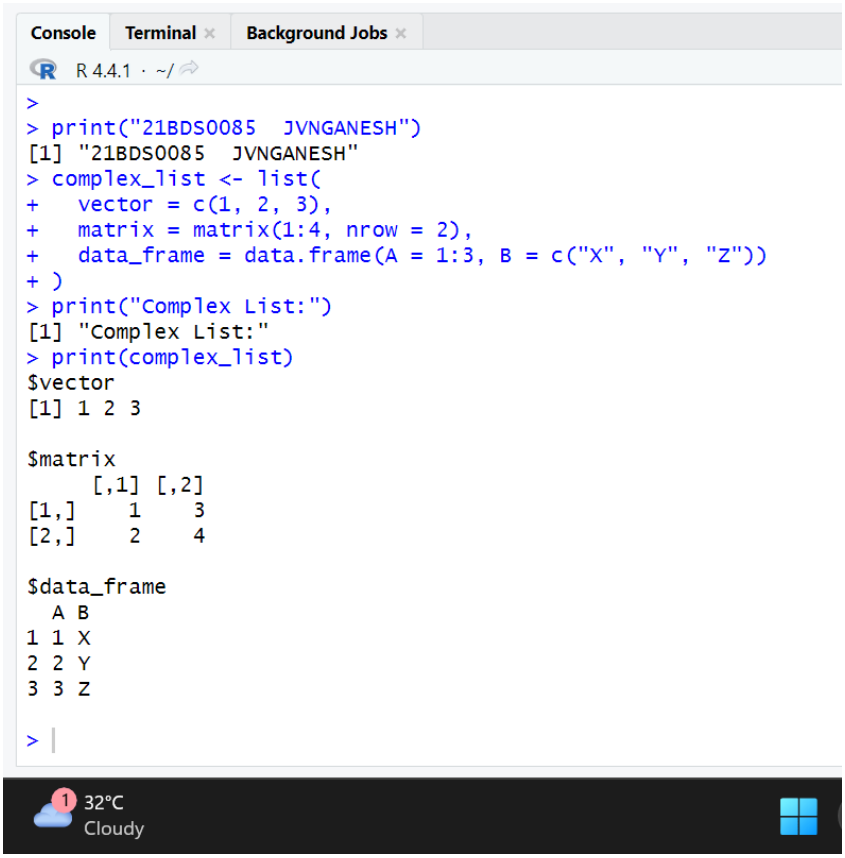
8. Lists with Different Types

Code

#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)
```

Output

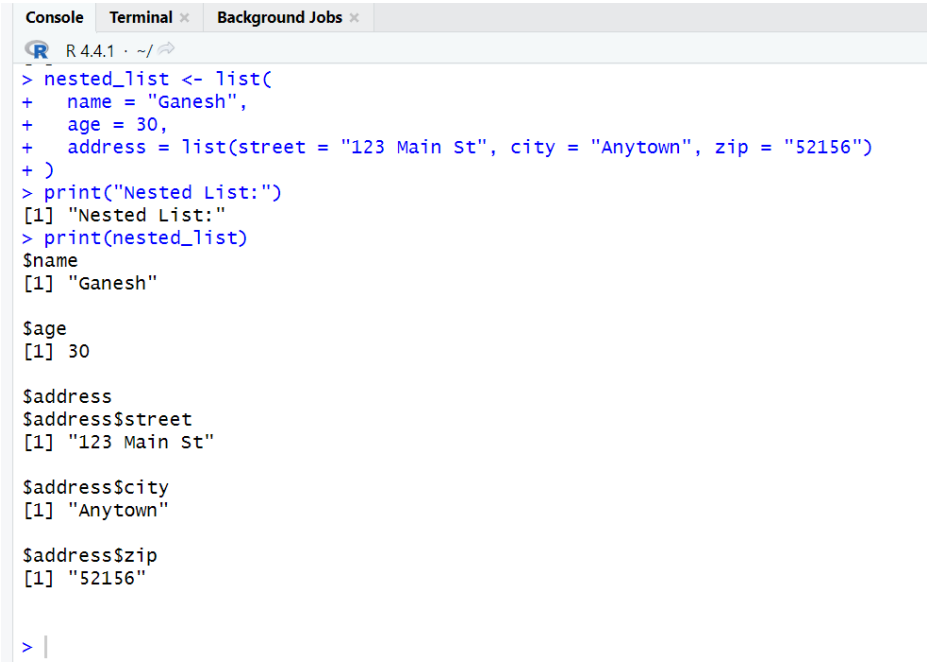


9. Nested Lists

Code

```
#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)
```

Output

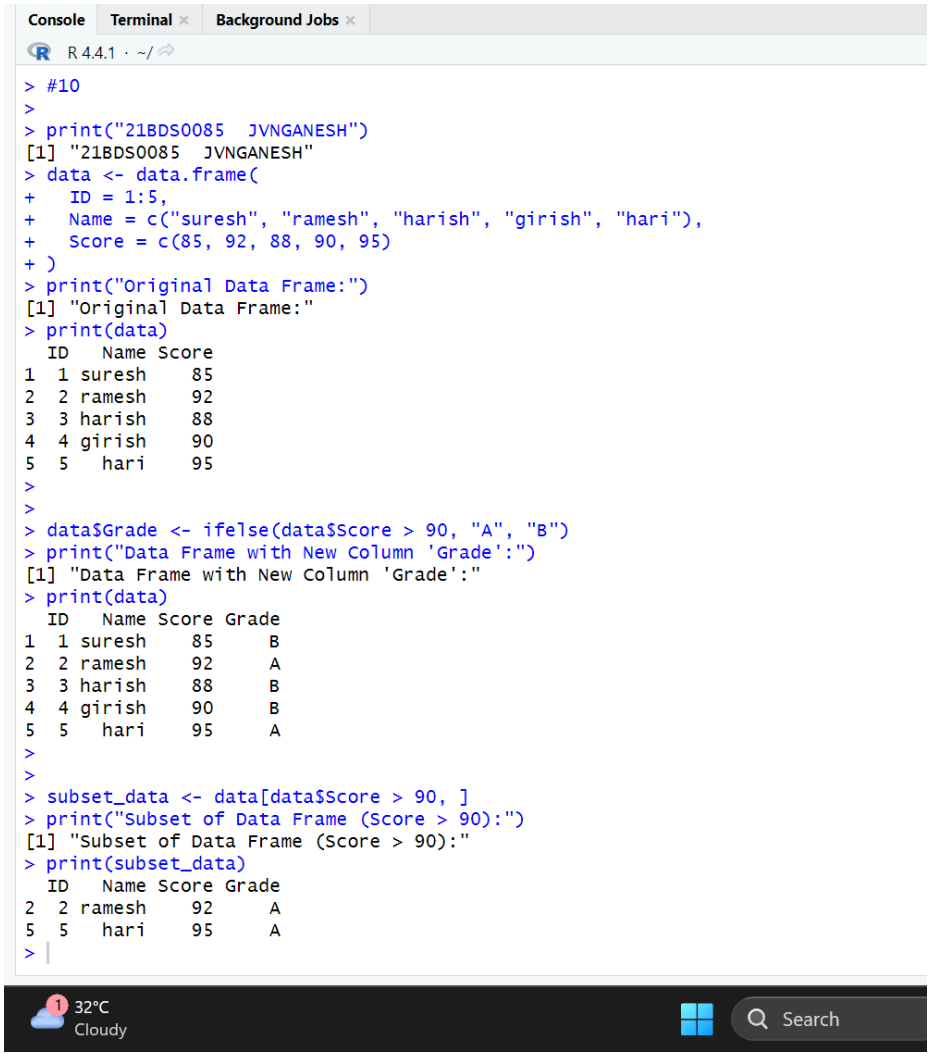


10. Creating and Manipulating a Data Frame

Code:

```
#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)
```

Output


Console
Terminal
Background Jobs
R 4.4.1 · ~/
> #10
>
> print("21BDS0085 JVNGANESH")
[1] "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:")
[1] "Original Data Frame:"
> print(data)
  ID   Name Score
1 1 suresh   85
2 2 ramesh   92
3 3 harish   88
4 4 girish   90
5 5   hari   95
>
>
> data$Grade <- ifelse(data$Score > 90, "A", "B")
> print("Data Frame with New Column 'Grade':")
[1] "Data Frame with New Column 'Grade':"
> print(data)
  ID   Name Score Grade
1 1 suresh   85   B
2 2 ramesh   92   A
3 3 harish   88   B
4 4 girish   90   B
5 5   hari   95   A
>
>
> subset_data <- data[data$Score > 90,]
> print("Subset of Data Frame (Score > 90):")
[1] "Subset of Data Frame (Score > 90):"
> print(subset_data)
  ID   Name Score Grade
2 2 ramesh   92   A
5 5   hari   95   A
>
32°C
Cloudy
Search

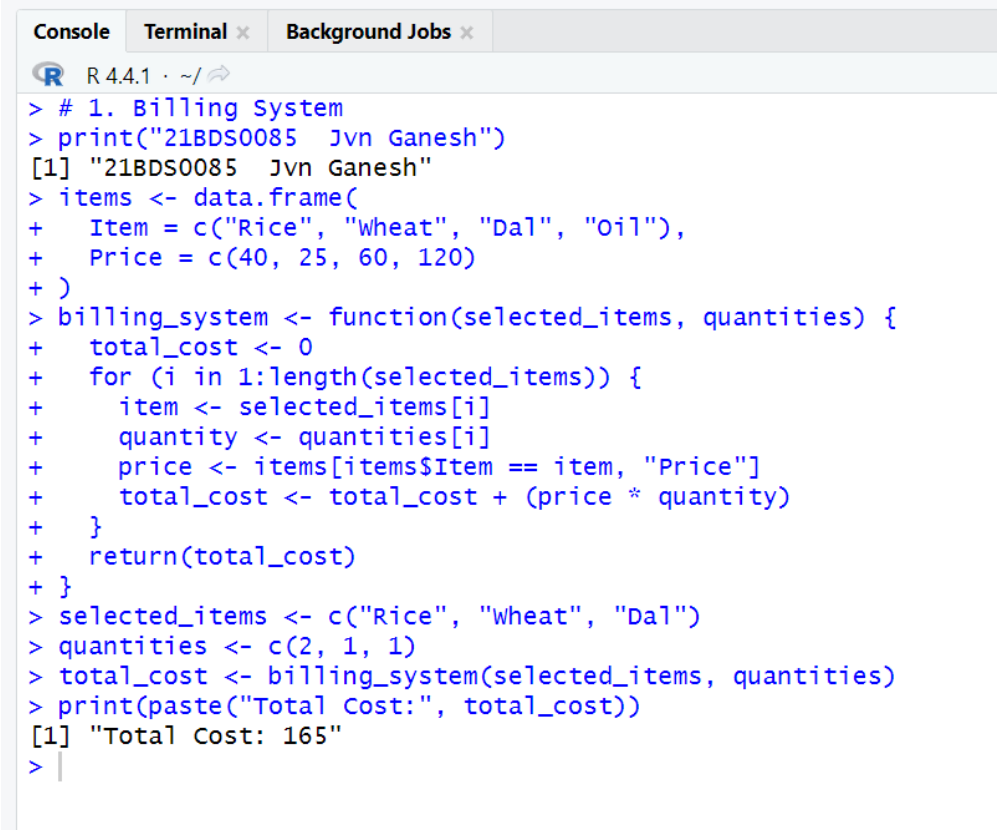
PROGRAMS USING DATA STRUCTURES

1. Billing System

Code

```
# 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))
```

Output



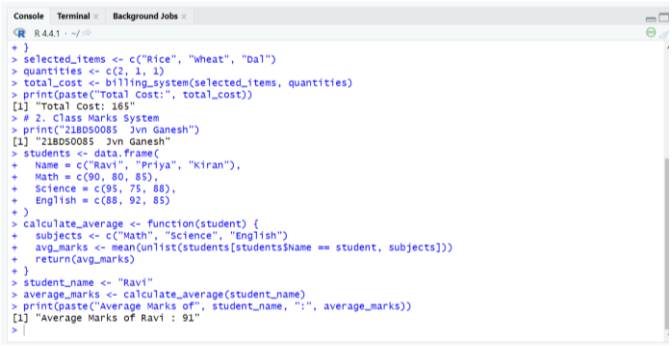
2. Class Marks System

Code

```
# 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))
```

Ouput



3. Employee records

Code

```
# 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)
```

Output

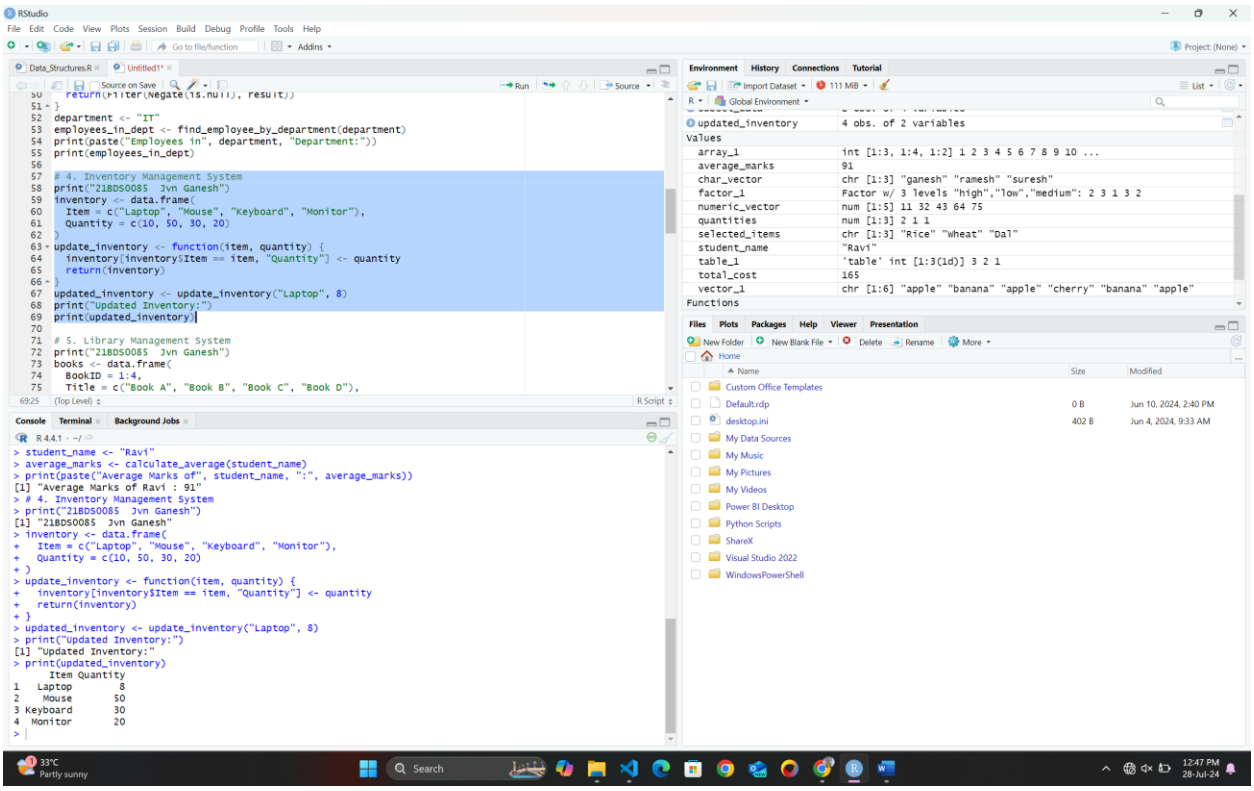
4. Inventory management system

Code

```
# 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)
```

Output



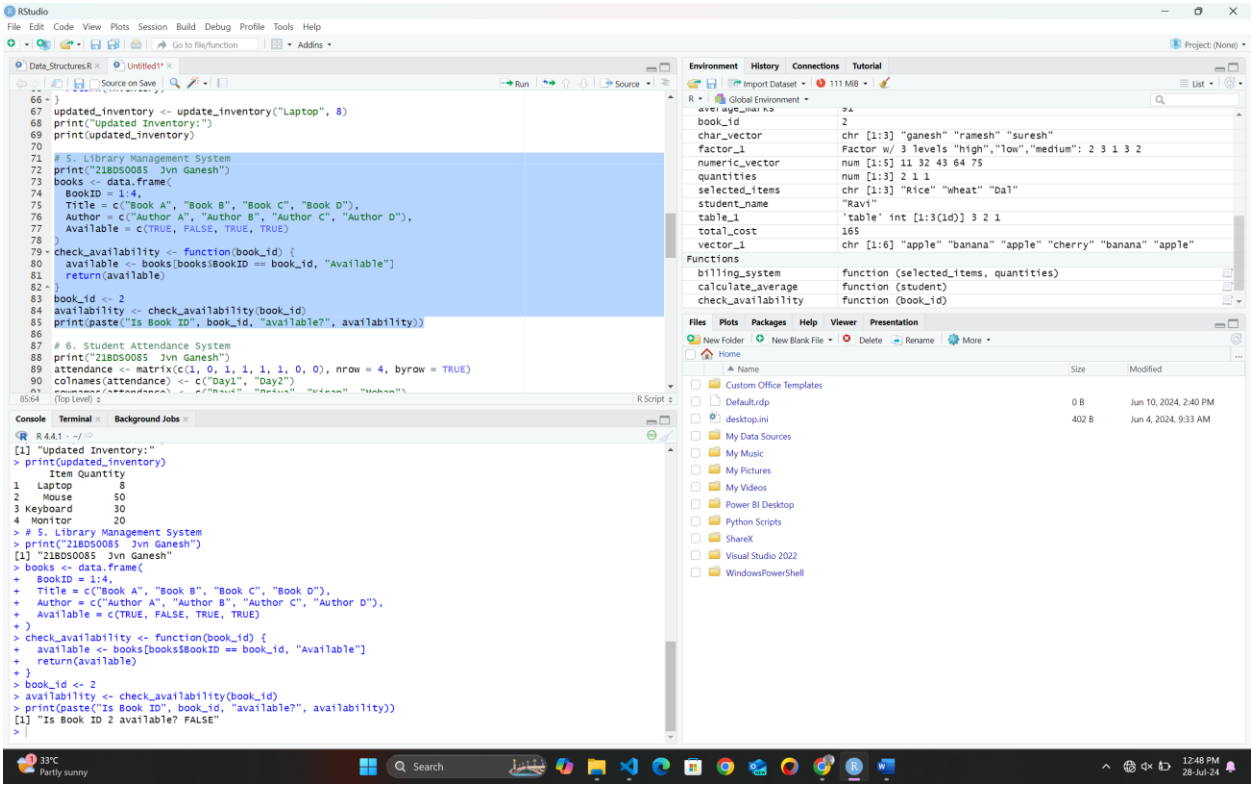
5.Library Management system

Code

```
# 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))
```

Output



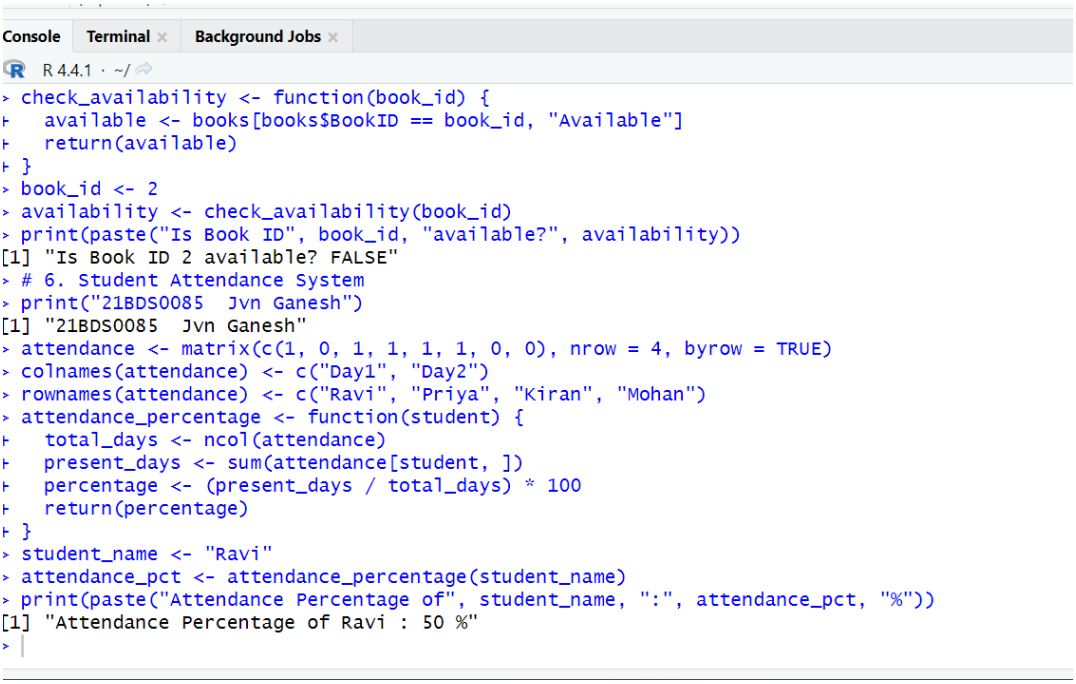
6.Student attendance

Code

```
# 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, "%"))
```

Output

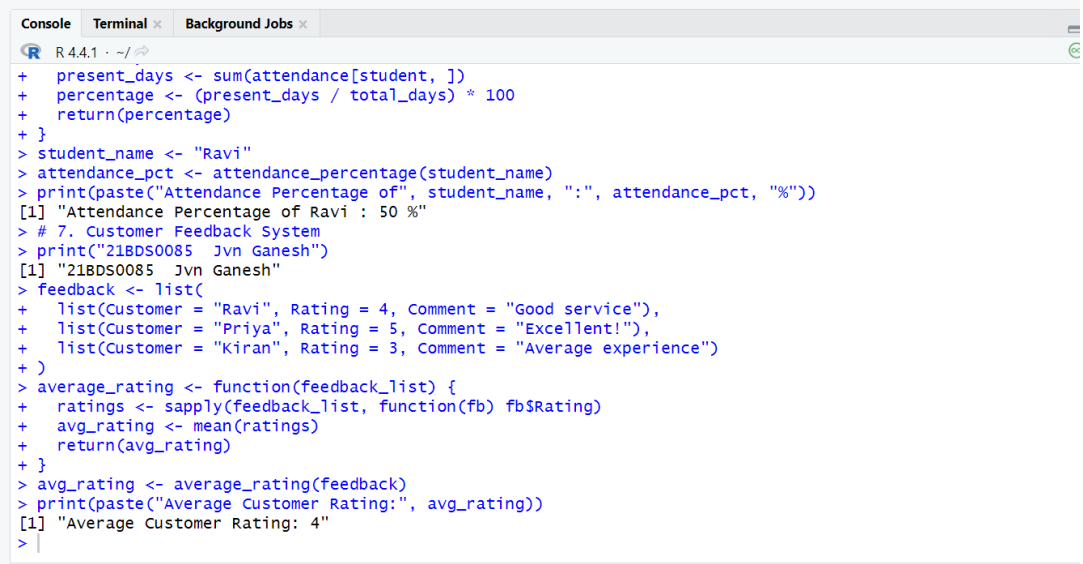


7. Customer feedback system

Code

```
# 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))
```

Output

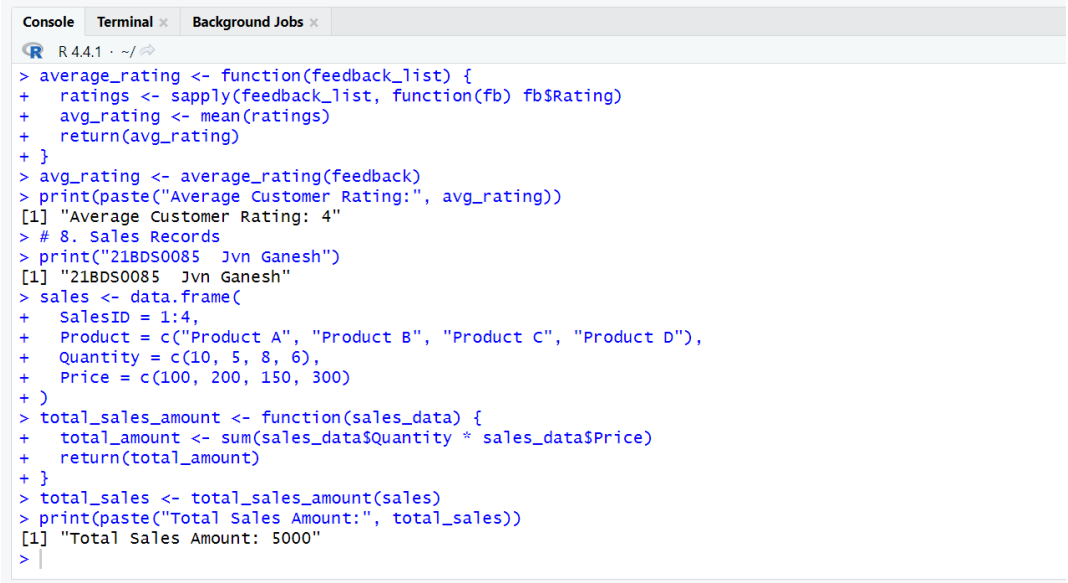


8. Sales record

Code

```
# 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))
```

Output

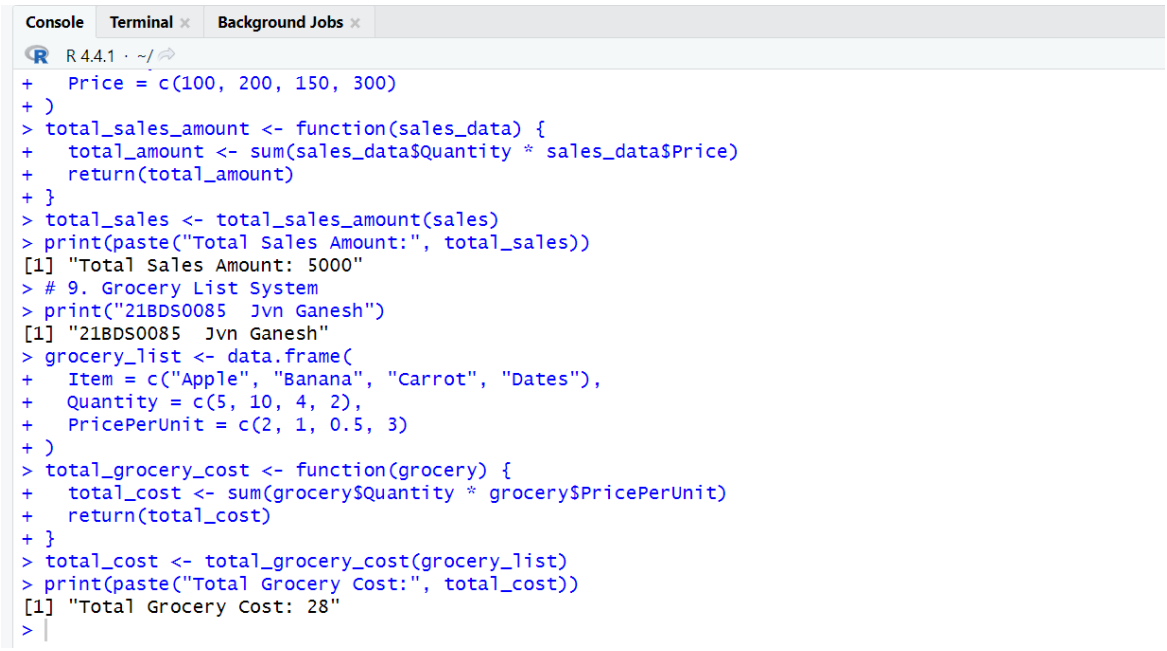


9.Grocery list system

Code

```
# 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))
```

Output



10.University enrollment system

Code

```
# 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)
```

Output

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")
```