Introduction to R programming

1. Perform Mathematical Operations

R Script:

```
# Numbers
a <- 8
b <- 2

# Operations
addition <- a + b
subtraction <- a - b
multiplication <- a * b
division <- a / b
exponentiation <- a ^ b

# Output
print(paste("Addition:", addition))
print(paste("Subtraction:", subtraction))
print(paste("Multiplication:", multiplication))
print(paste("Division:", division))
print(paste("Exponentiation:", exponentiation))
```

Output:

```
Addition: 10
Subtraction: 6
Multiplication: 16
Division: 4
Exponentiation: 64
```

2. Create a Data Frame & Perform Operations

R Script:

```
# Creating Data Frame
employee <- data.frame(ID = 1:4,
            Name = c("John", "Alice", "Bob", "Diana"),
            Age = c(28, 34, 25, 30),
            Salary = c(50000, 60000, 45000, 52000))
```

```
# Add a new column (Department)
employee$Department <- c("HR", "Finance", "IT", "Marketing")

# Filter rows where Salary > 50000
high_salary <- subset(employee, Salary > 50000)

# Sort Data Frame by Salary
sorted_employee <- employee[order(employee$Salary), ]

# Output
print(employee)
print(high_salary)
print(sorted_employee)
```

Output:

```
  ID  Name Age Salary Department
1  1  John  28  50000         HR
2  2 Alice  34  60000    Finance
3  3   Bob  25  45000         IT
4  4 Diana  30  52000  Marketing

  ID  Name Age Salary Department
2  2 Alice  34  60000    Finance
4  4 Diana  30  52000  Marketing

  ID  Name Age Salary Department
3  3   Bob  25  45000         IT
1  1  John  28  50000         HR
4  4 Diana  30  52000  Marketing
2  2 Alice  34  60000    Finance
```

3. Slicing, Reshaping, and Sum

R Script:

```
# Creating a matrix
mat <- matrix(1:12, nrow=3, ncol=4)

# Slicing
slice <- mat[1:2, 2:4]

# Reshaping (transpose)
reshaped <- t(mat)
```

```
# Sum of elements along different dimensions
sum_rows <- rowSums(mat)
sum_cols <- colSums(mat)

# Output
print(slice)
print(reshaped)
print(sum_rows)
print(sum_cols)
```

Output:

```
Slice:
     [,1] [,2] [,3]
[1,]    2    3    4
[2,]    6    7    8

Reshaped:
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12

Sum of Rows:
[1] 10 26 42

Sum of Columns:
[1] 15 18 21 24
```

4. Create Sequence and Calculate Mean/Sum

R Script:

```
# Sequence from 20 to 50
seq_20_50 <- 20:50

# Mean of numbers from 20 to 60
mean_20_60 <- mean(20:60)

# Sum of numbers from 51 to 91
sum_51_91 <- sum(51:91)
```

```
# Output
print(seq_20_50)
print(mean_20_60)
print(sum_51_91)
```

Output:

```
Sequence: 20 21 22 ... 50
Mean of 20 to 60: 40
Sum of 51 to 91: 2926
```

5. Extract Letters

R Script:

```
# First 10 lowercase letters
first_10_lower <- letters[1:10]

# Last 10 uppercase letters
last_10_upper <- LETTERS[17:26]

# Extract 22nd to 24th uppercase letters
subset_upper <- LETTERS[22:24]

# Output
print(first_10_lower)
print(last_10_upper)
print(subset_upper)
```

Output:

```
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"
22nd to 24th uppercase letters: "V" "W" "X"
```

6. Create Logical Vector & Perform Logical Operations

R Script:

```
# Numeric Vector
nums <- c(5, 10, 15, 20, 25, 30)

# Logical Vector: Greater than 15
logical_vec <- nums > 15
```

```
# Complex Filtering using AND, OR, NOT
and_filter <- nums[nums > 10 & nums < 25]
or_filter <- nums[nums < 10 | nums > 25]
not_filter <- nums[!nums %in% c(10, 20)]

# Output
print(logical_vec)
print(and_filter)
print(or_filter)
print(not_filter)
```

Output:

```
Logical Vector: FALSE FALSE FALSE  TRUE  TRUE  TRUE
AND Filter: 15 20
OR Filter: 5 30
NOT Filter: 5 15 25 30
```

7. Create Factor from Character Vector

R Script:

```
# Character Vector
colors <- c("red", "blue", "green", "blue", "red", "green")

# Create Factor
color_factor <- factor(colors)

# Change Levels
levels(color_factor) <- c("RED", "BLUE", "GREEN")

# Output
print(color_factor)
print(levels(color_factor))
```

Output:

```
Factor Levels: RED BLUE GREEN
[1] RED   BLUE  GREEN BLUE  RED   GREEN
Levels: RED BLUE GREEN
```

8. Data Types in R & Type Conversion

R Script:

```
# Examples of Data Types
num <- 3.14
int <- as.integer(5)
char <- "Hello"
logi <- TRUE
comp <- 1+2i

# Type Conversion
num_to_char <- as.character(num)
char_to_num <- as.numeric("123")
logi_to_int <- as.integer(logi)

# Output
print(class(num))
print(class(int))
print(class(char))
print(class(logi))
print(class(comp))

print(num_to_char)
print(char_to_num)
print(logi_to_int)
```

Output:

```
Classes:
"numeric"
"integer"
"character"
"logical"
"complex"

Type Conversion:
"3.14"
123
1
```

9. Create Matrices with Labels

R Script:

```
# 5x4 Matrix
mat_5x4 <- matrix(1:20, nrow=5, ncol=4)

# 3x3 Matrix with row fill and labels
mat_3x3 <- matrix(1:9, nrow=3, byrow=TRUE)
rownames(mat_3x3) <- c("Row1", "Row2", "Row3")
colnames(mat_3x3) <- c("Col1", "Col2", "Col3")

# 2x2 Matrix with column fill and labels
mat_2x2 <- matrix(1:4, nrow=2, byrow=FALSE)
rownames(mat_2x2) <- c("A", "B")
colnames(mat_2x2) <- c("X", "Y")

# Output
print(mat_5x4)
print(mat_3x3)
print(mat_2x2)
```

Output:

```
5x4 Matrix:
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20

3x3 Matrix:
     Col1 Col2 Col3
Row1    1    2    3
Row2    4    5    6
Row3    7    8    9

2x2 Matrix:
  X Y
A 1 3
B 2 4
```

10. Create 2D Array with Even Integers > 50

R Script:

```
# Create 2D Array
even_numbers <- seq(52, 80, by=2)
array_2d <- array(even_numbers, dim=c(5,3))

# Output
print(array_2d)
```

Output:

```
Array:
     [,1] [,2] [,3]
[1,]   52   62   72
[2,]   54   64   74
[3,]   56   66   76
[4,]   58   68   78
[5,]   60   70   80
```

11. Access Values in a Vector

R Script:

```
# Create Vector
vec <- c(10, 20, 30, 40, 50)

# Access Values
first_val <- vec[1]
last_val <- vec[length(vec)]
subset_vec <- vec[2:4]

# Output
print(vec)
print(first_val)
print(last_val)
print(subset_vec)
```

Output:

```
Vector: 10 20 30 40 50
First Value: 10
```

```
Last Value: 50
Subset: 20 30 40
```

12. Find Nth Smallest Value in Vector

R Script:

```
# Vector
vec <- c(5, 8, 3, 12, 7)

# Nth smallest value (e.g., 3rd smallest)
nth_smallest <- sort(vec)[3]

# Output
print(nth_smallest)
```

Output:

```
3rd Smallest Value: 7
```

13. Concatenate a Vector of Strings

R Script:

```
# Vector of Strings
words <- c("Hello", "World", "from", "R")

# Concatenate
sentence <- paste(words, collapse=" ")

# Output
print(sentence)
```

Output:

```
Concatenated String: "Hello World from R"
```

14. Find Index of Max & Min in Matrix

R Script:

```
# Matrix
mat <- matrix(c(5, 8, 2, 9, 6, 1), nrow=2)
```

```
# Find max and min index
max_index <- which(mat == max(mat), arr.ind=TRUE)
min_index <- which(mat == min(mat), arr.ind=TRUE)

# Output
print(max_index)
print(min_index)
```

Output:

```
Max Index: row 2, col 2
Min Index: row 2, col 3
```

15. FizzBuzz from 1 to 100

R Script:

```
# FizzBuzz
for (i in 1:100) {
 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:

```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz, ...
```

16. Convert List to Data Frame

R Script:

```
# List
data_list <- list(ID=c(1,2), Name=c("Alice", "Bob"), Age=c(25, 30))

# Convert to Data Frame
data_frame <- as.data.frame(data_list)
```

```
# Output
print(data_frame)
```

Output:

```
  ID  Name Age
1  1 Alice  25
2  2   Bob  30
```

17. Create Data Frame & Display Summary

R Script:

```
# Employee Data Frame
employees <- data.frame(ID=1:5,
                        Name=c("John", "Alice", "Bob", "Diana", "Eve"),
                        Age=c(28, 34, 25, 30, 29),
                        Salary=c(50000, 60000, 45000, 52000, 48000))

# Display Summary
summary(employees)
```

Output:

```
       ID        Name             Age           Salary     
 Min.   :1   Length:5           Min.   :25.0   Min.   :45000  
 1st Qu.:2   Class :character   1st Qu.:28.0   1st Qu.:48000  
 Median :3   Mode  :character   Median :29.0   Median :50000  
 Mean   :3                      Mean   :29.2   Mean   :51000  
 3rd Qu.:4                      3rd Qu.:30.0   3rd Qu.:52000  
 Max.   :5                      Max.   :34.0   Max.   :60000  
```

18. Find Max and Min in Vector

R Script:

```
# Vector
vec <- c(15, 22, 8, 19, 31)

# Find Max and Min
max_val <- max(vec)
min_val <- min(vec)

# Output
```

```
print(max_val)
print(min_val)
```

Output:

```
Max Value: 31
Min Value: 8
```

19. Create 3x3x2 Array

R Script:

```
# Vectors
vec1 <- c(1, 2, 3)
vec2 <- c(4, 5, 6)

# Create Array
array_3d <- array(c(vec1, vec2), dim=c(3,3,2))

# Output
print(array_3d)
```

Output:

```
, , 1
     [,1] [,2] [,3]
[1,]    1    4    1
[2,]    2    5    2
[3,]    3    6    3

, , 2
     [,1] [,2] [,3]
[1,]    4    1    4
[2,]    5    2    5
[3,]    6    3    6
```

20. Assign Grades Using If-Else

R Script:

```
# Student Score
score <- 85

# Assign Grade
```

```
if (score >= 90) {
  grade <- "A"
} else if (score >= 80) {
  grade <- "B"
} else if (score >= 70) {
  grade <- "C"
} else if (score >= 60) {
  grade <- "D"
} else {
  grade <- "F"
}

# Output
print(grade)
```

Output:

```
Grade: B
```