Slip 1

```
v1 <- c(20, 40, 44, 60)
v2 <- c(10, 20, 4, 12)
add_vec <- v1 + v2
multi_vec <- v1 * v2
divide_vec <- v1 / v2
print(add_vec)
print(multi_vec)
print(divide_vec)
```

Slip 2

```
multi_table <- function(n, length) {
for (i in 1:length) {
result <- n * i
cat(sprintf("%d x %d = %d\n", n, i, result))
}
}
multi_table(13, 10)
```

Slip 3

```
reverseAndSumDigits <- function(num) {
  reverseNumber <- function(n) {
    rev_num <- 0
    while (n > 0) {
      rev_num <- rev_num * 10 + (n %% 10)
      n <- n %/% 10
    }
    return(rev_num)
  }
  sumOfDigits <- function(n) {
    digit_sum <- 0
    while (n > 0) {
      digit_sum <- digit_sum + (n %% 10)
      n <- n %/% 10
    }
    return(digit_sum)
  }
  reversed <- reverseNumber(num)
  sum_digits <- sumOfDigits(reversed)
  cat("Original Number:", num, "\n")
  cat("Reversed Number:", reversed, "\n")
  cat("Sum of Digits of Reversed Number:", sum_digits, "\n")
}
number <- 76341
reverseAndSumDigits(number)
```

Slip 4

```
sum_of_matrices <- function(m1, m2) {
  if (all(dim(m1) == dim(m2))) {
    sum_matrix <- matrix(0, nrow = dim(m1)[1], ncol = dim(m1)[2])
    for (i in 1:dim(m1)[1]) {
      for (j in 1:dim(m1)[2]) {
```

```
        sum_matrix[i, j] <- m1[i, j] + m2[i, j]
      }
    }
    return(sum_matrix)
  } else {
    stop("Matrices must be of the same size.")
  }
}
m1 <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2)
m2 <- matrix(c(5, 6, 7, 8), nrow = 2, ncol = 2)
sum_matrix <- sum_of_matrices(m1, m2)
print(sum_matrix)
```

Slip 5

```
f1 <- factor(c("A", "B", "C"))
f2 <- factor(c("X", "Y", "Z"))
result <- c(f1, f2)
print(result)
```

Slip 6

```
v1 <- c(1, 2, 3, 4, 5, 2, 7)
v2 <- c('a', 'b', 'c', 'd', 'e', 'b', 'g')
my_data <- data.frame(Column1 = v1, Column2 = v2)
print("Original Data Frame:")
print(my_data)
duplicate_rows <- my_data[duplicated(my_data) |
duplicated(my_data, fromLast = TRUE), ]
print("Duplicate Elements:")
print(duplicate_rows)
```

Slip 7

```
s1 <- seq(20, 50, by = 1)
m1 <- mean(seq(20, 60, by = 1))
s2 <- sum(seq(51, 91, by = 1))
cat("Sequence from 20 to 50:", s1, "\n")
cat("Mean of numbers from 20 to 60:", m1, "\n")
cat("Sum of numbers from 51 to 91:", s2, "\n")
```

Slip 8

```
fibonacci <- function(n) {
  fib_seq <- numeric(n)
  if (n >= 1) {
    fib_seq[1] <- 0
  }
   if (n >= 2) {
    fib_seq[2] <- 1
  }
  for (i in 3:n) {
    fib_seq[i] <- fib_seq[i - 1] + fib_seq[i - 2]
  }
  return(fib_seq)
}
result <- fibonacci(10)
cat("First 10 Fibonacci numbers: ", result, "\n")
```

Slip 9

```
employee_data <- data.frame(
  employeeID = c(1, 2, 3, 4, 5),
  name = c("Rohan Joshi", "Manoj Jangid", "Avinash More", "Anil
Kamthe", "Shubham Kangle"),
  age = c(28, 35, 42, 30, 45),
  department = c("IT", "HR", "Finance", "Marketing",
"Operations"),
  salary = c(60000, 50000, 70000, 55000, 75000)
)
cat("Employee Data:\n")
print(employee_data)
cat("\nSummary of Employee Data:\n")
summary(employee_data)
```

Slip 10

```
vec <- c(5, 10, 3, 8, 15, 2, 7)
max_value <- max(vec)
min_value <- min(vec)
cat("Original Vector:", vec, "\n")
cat("Maximum Value:", max_value, "\n")
cat("Minimum Value:", min_value, "\n")
```

Slip 11

```
l1 <- list("x", "y", "z")
l2 <- list("X", "Y", "Z", "x", "y", "z")
result_setdiff <- setdiff(l2, l1)
cat("Using setdiff:")
print(result_setdiff)
```

Slip 12

```
employee_data <- data.frame(
  empno = c(101, 102, 103, 104, 105),
  empname = c("Rajesh", "Sneha", "Amit", "Priya", "Vikram"),
  gender = c("Male", "Female", "Male", "Female", "Male"),
  age = c(28, 24, 32, 29, 35),
  designation = c("Manager", "Software Engineer", "HR Manager",
"Data Analyst", "Senior Developer")
)
print("Employee Details:")
print(employee_data)
print("\nSummary:")
print(summary(employee_data))
```

Slip 13

```
digits <- c(1, 2, 3, 4, 5, 6)
frequency <- c(7, 2, 6, 3, 4, 8)
pie(frequency, labels = digits, main = "Distribution of Digits on
Dice")
```

Slip 14

```
employee_list <- list("Rahul", "Priya", "Amit", "Anita", "Vikram")
cat("Names of Employees:\n")
print(employee_list)
cat("\n")
new_employee <- "Deepika"
employee_list <- c(employee_list, new_employee)
cat("After adding", new_employee, "at the end:\n")
print(employee_list)
cat("\n")
index_to_remove <- 3
employee_list <- employee_list[-index_to_remove]
cat("After removing the third element:\n")
print(employee_list)
```

Slip 15

```
v1 <- c(20, 40, 44, 60)
v2 <- c(10, 20, 4, 12)
add_vec <- v1 + v2
multi_vec <- v1 * v2
divide_vec <- v1 / v2
Print(add_vec)
print(multi_vec)
print(divide_vec)
```

Slip 16

```
data <- data.frame(
  year = c(2001, 2002, 2003),
  export = c(26, 32, 35),
  import = c(35, 40, 50)
)
barplot(
  height = t(data[, c("export", "import")]),
  beside = TRUE,
  col = c("blue", "red"),
  names.arg = data$Year,
  ylab = "value",
  xlab = "year",
  main = "export and import Data",
  legend.text = c("export", "import"),
  args.legend = list(x = "topright"),
  Ylim = c(0, max(data$Export, data$Import) + 5)
)
```

Slip 17

```
fibonacci <- function(n) {
  fib_seq <- numeric(n)
  if (n >= 1) {
    fib_seq[1] <- 0
  }
  if (n >= 2) {
    fib_seq[2] <- 1
  }
  for (I in 3:n) {
```

```
    fib_seq[i] <- fib_seq[i - 1] + fib_seq[i - 2]
  }
  return(fib_seq)
}
result <- fibonacci(20)
cat("First 20 Fibonacci numbers: ", result, "\n")
```

Slip 18

```
vec <- c(5, 10, 3, 8, 15, 2, 7)
max_value <- max(vec)
min_value <- min(vec)
cat("Original Vector:", vec, "\n")
cat("Maximum Value:", max_value, "\n")
cat("Minimum Value:", min_value, "\n")
```

Slip 19

```
students_data <- data.frame(
  rollno = c(101, 102, 103, 104, 105),
  studname = c("Rajesh", "Sunita", "Amit", "Priya", "Vikas"),
  address = c("Delhi", "Mumbai", "Bangalore", "Chennai",
"Kolkata"),
  marks = c(85, 92, 78, 95, 88)
)
print(students_data)
```

Slip 20

```
name <- c("Rahul", "Priya", "Amit", "Sneha")
age <- c(28, 24, 30, 26)
city <- c("Mumbai", "Delhi", "Bangalore", "Kolkata")
salary <- c(60000, 55000, 70000, 50000)
df <- data.frame(name, age, city, salary)
print(df)
```