Part A

1. **Suppose that queue <- c ("Steve", "Russell", "Alison", "Liam") and that queue represents a supermarket queue with Steve first in line. Using R expressions update the supermarket queue as successively:**
   a. **Barry arrives**
   b. **Steve is served**
   c. **Pam takes her way to the front with one item**
   d. **Barry gets impatient and leaves**
   e. **Alison gets impatient and leaves**

   For the last case you should not assume that you know where in the queue Alison is standing.

   Finally, using the function which(x), find the position of Russell in the queue

   ```
   #set queue

   queue <- c("Steve", "Russell", "Alison", "Liam") #add
   barry to the end of the queue
   queue[length(queue)+1]="Barry"

   #remove steve from the queue queue
   <- queue[queue != "Steve"]

   #add pam to the beginning of the queue queue
   <- append(queue, "Pam", after = 0) #remove
   barry from queue

   queue <- queue[queue != "Barry"]
   #remove alison from queue

   queue <- queue[queue != "Alison"] queue

   #find russell's position in the queue
   which(queue=="Russell")
   ```

2. **Plot the graph of linear equation, Non-linear equation : square, cubic, square root, etc.,**

   ```
   x <- seq(-5, 10, by = 0.1)

   M <- 2

   C <- 4

   eq <- M * x + C
   ```

```
        neq1 <- x^2

neq2 <- x^3

neq3 <- sqrt(abs(x))

plot(x,eq, type = "l", col = "blue ", lty = 2,xlab = "X", ylab = "Y", main = "Graphs of Linear and
Non-linear Equations")

lines(x, neq1, col = "red")

lines(x, neq2, col = "green")

lines(x, neq3, col = "purple")

legend("topright", legend = c("Linear (Y = 2X + 5)", "Non-linear (Y = X^2)", "Non-linear (Y =
X^3)", "Non-linear (Y = √X)"),  col = c("blue", "red", "green", "purple"), lty = c(2, 1, 1, 1), cex =
0.8)
```

3. **The performance of a student in 3rd semester CSE as given below {SUB1, SUB2, SUB3, SUB4, SUB5, SUB6} with score {91, 73, 65, 45, 54, 32}. Draw the Bar chart to indicate the performance of the student Draw the pie chart to indicate the performance of the student Find the average marks and discuss qualitatively about the performance in each subject.**

```
subject<- c("MFC", "DAA", "MCES", "OS", "BFE", "UHV")

marks <- c(91, 73, 65, 45, 54, 32)

avg<- mean(marks)

barplot(marks, names.arg = subject, col = "skyblue",main = "Student's Performance i n 3rd
Semester CSE", xlab = "Subjects", ylab = "Marks ")

pie(marks, labels = subject, col = rainbow(length(marks)), main = "Student's Performance in
3 rd Semester CSE")

cat("Average Marks:", avg, "\n")

cat("Qualitative Performance:\n")

for (i in 1:length(marks)) {

  cat(subject[i], ":", ifelse(marks[i] >= avg, "Good", "Needs Improvement"), "\n")

}
```

Part B

4. **You have been tasked with creating a program that calculates and assigns grades for students enrolled in multiple courses. The program will take input for the marks obtained by 10 students in 5 different courses, compute the total and average marks for each student, and assign corresponding grades based on their average performance.**
   - **Declare constants for the number of students (num_students) and the number of courses (num_courses).**
   - **Initialize an empty list to store student information.**
   - **For each student:**
     - **Input the student's name.**
     - **Input marks for each of the 5 courses.**
     - **Calculate the total marks and average marks.**
     - **Determine the grade based on the average marks using a grading scale.**
     - **Display the student information, including their name, individual course marks, total marks, average marks, and the assigned grade.**

Program:

```
# Constants

num_students <- 5

num_courses <- 5

# Predefined student names

student_names <- c("Arun Rahul", "Bheem Kumar","Raj jumar","Jahal  A R","Suresh")

# Predefined course marks for each student

course_marks <- matrix(c(

  85, 92, 78, 88, 95,

  75, 80, 85, 70, 60,

  100,78,56,34,56,

  78,45,67,89,90,

  89,80,67,78,90

), nrow = num_students, byrow = TRUE)


# Initialize a list to store student information

student_records <- list()
```

```r
# Loop for each student
for (student_index in 1:num_students) {
  student_name <- student_names[student_index]

  # Initialize variables for calculations
  total_marks <- sum(course_marks[student_index, ])
  average_marks <- total_marks / num_courses

  # Determine grade based on average marks
  grade <- ifelse(average_marks >= 90, "A",
          ifelse(average_marks >= 80, "B",
              ifelse(average_marks >= 70, "C",
                  ifelse(average_marks >= 60, "D", "F"))))
  # Store student information in a record
  student_record <- list(name = student_name, marks = course_marks[student_index, ],
              total = total_marks, average = average_marks, grade = grade)

  student_records <- c(student_records, list(student_record))
}
# Display student information
cat("\nStudent Grade Report:\n")
for (student_record in student_records) {
  cat("\nName:", student_record$name, "\n")
  cat("Marks:", student_record$marks, "\n")
  cat("Total Marks:", student_record$total, "\n")
  cat("Average Marks:", student_record$average, "\n")
  cat("Grade:", student_record$grade, "\n")
}
```

2.   You have been tasked with creating a program that calculates and assigns grades for students enrolled in multiple courses. The program will take

input for the marks obtained by 10 students in 5 different courses, compute the total and average marks for each student, and assign corresponding grades based on their average performance.

Declare constants for the number of students (num_students) and the number of courses (num_courses).

Initialize an empty list to store student information.

For each student:

•   Define the student's name.

•   Define marks for each of the 5 courses.

•   Calculate the total marks and average marks.

•   Determine the grade based on the average marks using a grading scale.

•   Display the student information, including their name, individual course marks, total marks, average marks, and the assigned grade.

Program:

```
# Constants
num_students <- 5
num_courses <- 5

# Predefined student names
student_names <- c("Arun Rahul", "Bheem Kumar","Raj jumar","Jahal  A R","Suresh")

# Predefined course marks for each student
course_marks <- matrix(c(
  85, 92, 78, 88, 95,
  75, 80, 85, 70, 60,
  100,78,56,34,56,
  78,45,67,89,90,
  89,80,67,78,90
), nrow = num_students, byrow = TRUE)
# Initialize a list to store student information
```

```r
student_records <- list()

# Loop for each student
for (student_index in 1:num_students) {
  student_name <- student_names[student_index]
    # Initialize variables for calculations
  total_marks <- sum(course_marks[student_index, ])
  average_marks <- total_marks / num_courses

  # Determine grade based on average marks
  grade <- ifelse(average_marks >= 90, "A",
            ifelse(average_marks >= 80, "B",
              ifelse(average_marks >= 70, "C",
                ifelse(average_marks >= 60, "D", "F"))))

  # Store student information in a record
  student_record <- list(name = student_name, marks = course_marks[student_index, ],
                total = total_marks, average = average_marks, grade = grade)

  student_records <- c(student_records, list(student_record))
}
# Display student information
cat("\nStudent Grade Report:\n")
for (student_record in student_records) {
  cat("\nName:", student_record$name, "\n")
  cat("Marks:", student_record$marks, "\n")
  cat("Total Marks:", student_record$total, "\n")
  cat("Average Marks:", student_record$average, "\n")
  cat("Grade:", student_record$grade, "\n")
}
```