

## #Practise questions

1. Create a numeric vector `v <- c(12, 5, 8, 21, 17)` and answer the following:

- a) Print the second and fourth elements of the vector.
- b) Find the mean and sum of the vector.
- c) Add 5 to each element of the vector and print the result.
- d) Replace the third element with 10.
- e) Create a new vector containing only the elements greater than 10.  
`j = v[v > 10]`

2. Create a list `mylist` containing the following elements:

- Name: "Ravi"
- Age: 19
- Marks: `c(70, 85, 90)`

```
mylist <- list(Name = "Ravi", Age = 19, Marks = c(70, 85, 90))
```

Answer the following:

- a) Access the Name and Age elements from the list.
- b) Calculate the average of the Marks.
- c) Add a new element Gender = "Male" to the list.
- d) Modify the second mark to 88.
- e) Print the structure of the list using `str()`.

3. Create a matrix `m` with numbers 1 to 9 filled row-wise in 3 rows and 3 columns:

```
m <- matrix(1:9, nrow = 3, byrow = TRUE)
```

Answer the following:

- a) Print the entire second row of the matrix.
- b) Calculate the sum of each row using `rowSums()`.
- c) Multiply each element of the matrix by 2.
- d) Transpose the matrix using `t()`.
- e) Find the sum of the diagonal elements using `diag()`.

4. `x <- 15`

`y <- 4.`

`v1 <- c(10, 20, 30)`

`v2 <- c(2, 4, 6)`

Answer the following:

- a) Use arithmetic operators to calculate `x + y`, `x %% y`, and `x ^ y`.
- b) Use relational operators to check if `x` is greater than or equal to `y`.
- c) Perform element-wise addition of `v1` and `v2`.
- d) Use `%in%` to check if 20 is present in `v1`.

5.

`marks <- c(56, 78, 92, 45, 68)`

`names <- c("Anu", "Ravi", "Zoya", "Amit", "Neha")`

`students <- data.frame(Name = names, Marks = marks)`

Answer the following:

- a) Print the names of students who scored more than 60.
- b) Add a new column `Grade` using `ifelse()` (`>=75 = "A"`, else `"B"`).
- c) Convert the `Grade` column into a factor.
- d) Create a list named `student_info` containing the data frame and total number of students.
- e) Access the third student's name using the list.

6.

`v <- c(5, 10, 15, 20, 25)`    `v[-c(1,2)]`

- a) Print the length of the vector
- b) Access the last two elements
- c) Multiply all elements by 2
- d) Replace the first element with 100
- e) Sort the vector in descending order

7.

`mylist <- list(Name = "Priya", Age = 20, Scores = c(75, 80, 90))`

- a) Print the Name from the list
- b) Find the maximum of Scores
- c) Add a new field `City = "Mumbai"`
- d) Change Age to 21
- e) Display the structure of the list

8.

```
m <- matrix(1:6, nrow = 2)
```

- a) Print the number of rows and columns
- b) Access the element at row 2, column 1
- c) Calculate the row sums
- d) Transpose the matrix
- e) Multiply matrix by 3

9.

```
students <- data.frame(  
  Name = c("Asha", "Vijay", "Meena", "Sahil"),  
  Marks = c(78, 45, 88, 56)  
)
```

- a) Display names of students with marks > 60
- b) Add a column Result ("Pass" if marks ≥ 50)
- c) Convert Result to factor
- d) Calculate average marks
- e) Arrange data in descending order of Marks

10.

```
x <- 25
```

```
y <- 4
```

```
v1 <- c(5, 10, 15)
```

```
v2 <- c(3, 5, 7)
```

- a) Find  $x \%/\% y$  and  $x \% \% y$
- b) Check which elements of v1 are > v2
- c) Perform logical AND between  $v1 > 5$  and  $v2 < 6$
- d) Use `%in%` to check if 10 is in v1
- e) Add v1 and v2 element-wise

11.

```
data <- list(  
  Name = "Suresh",  
  Age = 22,  
  Scores = list(Math = 80, Sci = 85)  
)
```

- a) Access the Math score
- b) Change Sci score to 90
- c) Add new subject: English = 75
- d) Convert Scores to a data frame
- e) Calculate the average score

12.

```
m <- matrix(c(10, 20, 30, 40, 50, 60), nrow = 2, byrow = TRUE)
```

- a) Find column sums
- b) Add a new row: c(70, 80, 90)
- c) Replace the second column with c(100, 200)
- d) Extract the second row as a vector
- e) Calculate mean of entire matrix

13.

```
names <- c("Raj", "Simran", "Amit")
```

```
ages <- c(21, 22, 20)
```

```
marks <- c(85, 90, 78)
```

- a) Create a data frame with the above vectors
- b) Add a column Grade: "A" if marks  $\geq$  80 else "B"
- c) Extract only names and grades
- d) Filter students with age > 20
- e) Find number of students who got "A"

14.

```
math <- data.frame(Name = c("A", "B"), Score = c(78, 85))
```

```
science <- data.frame(Name = c("A", "B"), Score = c(80, 90))
```

```
subject_data <- list(Math = math, Science = science)
```

- a) Access Science score of student "B"
- b) Add a new student "C" with scores (Math: 90, Sci: 95)
- c) Calculate average score in each subject
- d) Combine both subject data into one data frame with columns: Name, Math, Science
- e) Find student with highest total score

15.

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

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

```
  Math = c(80, 55, 70, 90, 60),
```

```
  English = c(85, 45, 75, 88, 65)
```

```
)
```

- a) Use `rowMeans()` to calculate average of each student
- b) Add column Status: "Pass" if both subjects  $\geq 50$
- c) Extract IDs of students who failed any subject
- d) Use `apply()` to find max score for each student
- e) Create a barplot of Math scores with IDs on x-axis

16.

You are given student roll numbers from 101 to 106.

- a) Create a vector of these roll numbers
- b) Add 5 to each roll number
- c) Extract only odd-numbered roll numbers
- d) Find the position of roll number 105
- e) Reverse the vector

17.

Create a vector of 5 fruit names.

- a) Find the length of the vector
- b) Check which fruit names have more than 5 letters
- c) Convert all names to uppercase
- d) Test if "Mango" is in the vector
- e) Sort the vector alphabetically

18.

Create a 2x3 matrix with numbers from 1 to 6.

- a) Assign row names as "R1", "R2" and column names as "C1", "C2", "C3"
- b) Find sum of second column
- c) Access the full second row
- d) Multiply entire matrix by 10
- e) Change the value in row 1, column 3 to 99

19.

Create two vectors of length 4 with any numeric values.

- a) Use ^ operator to raise elements of vector1 to vector2
- b) Find modulo and integer division results
- c) Identify element-wise maximum using pmax()
- d) Use logical vector to filter elements > 50
- e) Replace all negative values (if any) with 0

20.

Create a list with the following: student name, a numeric vector of 3 subject marks, and a logical value indicating pass/fail.

- a) Access the second subject mark
- b) Change student name
- c) Append city name to the list
- d) Calculate the mean of the marks
- e) Check the class of each list component

21.

Create a data frame of 4 books with columns: Title, Price, and Pages.

- a) Increase all prices by 10%
- b) Add a new column Category of your choice
- c) Find books with more than 300 pages
- d) Remove the Pages column
- e) Rename column Price to Cost

22.

Create a vector of 6 numbers (mix of positive and negative values).

- a) Find which numbers are greater than 10 and less than 50
- b) Find positions of all negative numbers
- c) Replace all numbers < 0 with NA
- d) Check if any number is exactly 25
- e) Count how many numbers are greater than 30

23.

Create a 3x3 matrix with custom values.

- a) Calculate mean of each row using `apply()`
- b) Subtract 5 from all elements
- c) Replace diagonal elements with 100
- d) Convert matrix to a data frame
- e) Compute sum of all even elements

24.

Create two data frames: one for January sales and one for February sales. Each should have columns Product, Units, and Price.

- a) Store both data frames in a list
- b) Add a new product row to both months
- c) Calculate total sales (Units  $\times$  Price) for each product
- d) Combine both data frames into one with a new Month column
- e) Find the product with the highest combined sales

25.

Create a data frame of 6 students with marks in 3 subjects.

- a) Add a column Average of all three marks
- b) Create a column Result (Pass if all subjects  $\geq 35$ )
- c) Find students who failed in any one subject
- d) Calculate subject-wise average using `colMeans()`
- e) Create a frequency table of the Result column

26.

```
grades = c("A","B","A","C","B","A","C","B","A")
```

- Convert grades into a factor and display the result
- Find the levels of factor
- Change the order of levels to "C"<"B"<"A"
- Count the number of students in each grade using **table()**
- Convert the factor back to character vector.

27.

```
marks = c(45, 67, 89, 56, 72, 33, 90, 81, 60)
```

- Create a factor variable **result** with two levels "pass" (marks>=50) & **fail** students.
- Display the frequency of **pass** & **fail**.
- Convert the factor result into numeric (1 = fail, 2 = pass)
- Add result as a new column to a data frame containing marks.
- Find how many students scored **pass** using sum

28.

```
Months = c("mar","jan","feb","mar","jan","feb","mar")
```

- Convert months into a factor
- Display the unique levels of the factor
- Find frequency of every months

29.

You are given a vector of marks:

```
marks <- c(95, 67, 48, 82, 76)
```

**Sub-questions:**

- Write an if-else statement to check if the first student passed ( $\geq 40$ ) or failed.
- Use a for loop with if conditions to print "Pass" or "Fail" for each student.
- Assign grades (A for  $\geq 85$ , B for 70–84, C for 50–69, D for  $< 50$ ) using nested if-else.
- Solve (c) again using ifelse() instead of nested if-else.

30.

You are given a vector:

```
nums <- 1:10
```

**Sub-questions:**

- Write a for loop to check whether each number is odd or even.
- Store "Odd" or "Even" in a new vector using ifelse().
- Print only the odd numbers using an if condition inside the loop.
- Count how many even numbers are present using a loop + counter.



31.

Take a variable:

```
day <- 3
```

**Sub-questions:**

- a) Use switch() to print the name of the day (1 = Monday, 2 = Tuesday, ..., 7 = Sunday).
- b) If day = 8, print "Invalid Input" using if-else.
- c) Extend the problem: if day is 6 or 7, print "Weekend", else "Weekday".

32.

**Sub-questions:**

- a) Write a for loop to compute the factorial of n = 5.
- b) Repeat using a while loop.
- c) Modify your code to print the intermediate result after each multiplication step.
- d) Add an if condition to check if n < 0, print "Factorial not defined".

33.

You have a data frame:

```
students <- data.frame(  
  Name = c("Anu", "Ravi", "Zoya", "Amit", "Neha"),  
  Marks = c(56, 78, 92, 45, 68)  
)
```

**Sub-questions:**

- a) Use if-else to check if the first student has "Pass" or "Fail" (Pass = Marks >= 50).
- b) Use a for loop to print the result for all students.
- c) Add a new column "Result" in the data frame using ifelse().
- d) Use nested if-else to assign "Excellent" (>80), "Good" (60–80), "Average" (40–59), "Fail" (<40).
- e) Use switch() to print "Scholarship Eligible" only if grade is "Excellent".

34. Write an R program to check whether a number is positive or negative using if-else

35. Check whether a given number is even or odd using **ifelse()**.

36. Write a **for loop** to print numbers from 1 to 5.

37. Check if a student with **marks = 55** has passed (>= 40) or failed using **if-else**.

38. Use **switch** to print the days of the week (1 = Monday, 7 = Sunday)

39. Assign a grade based on marks

= 85 → A

= 70 - 84 → B

= 50 - 69 → C

= <50 → fail

40. Write an R programming using a loop to calculate the sum of first 10 natural numbers.

41. Find factorial of 5 using while loop.

42. Classify numbers in a vector into odd/even using ifelse().

43. Given the marks of students, count how many have failed (<40).

44. Print a multiplication table of 7 using loop.

45. Given a dataframe of students add a column "Result" showing "Pass" (>=40) or "Fail".

46. If marks >= 85 → "Grade A : Scholarship Eligible",

70 – 84 → "Grade B",

50 – 69 → "Grade C"

else fail

47. Write an R programming to print prime numbers from 1 to 20 using loops

48. Use switch() to perform an arithmetic operations

49. Write a program to reverse using loop (example 12345 → 54321)

50. Use a loop (not max()) to find the maximum value in vector c(12, 45, 67, 23, 89, 34)