Program 1 {a}- Write a R program to perform some arithmetic expressions.

SOURCE CODE:

```
# R Program for Expressions
# This program demonstrates basic arithmetic expressions in R.

# Define variables
a <- 10
b <- 5

# Calculate and print the results of different expressions
sum <- a + b
difference <- a - b
product <- a * b
quotient <- a / b

# Print the results
print(paste("Sum:",sum))
print(paste("Difference:", difference))
print(paste("Product:", product))
print(paste("Quotient:", quotient))
```

Sample OUTPUT:

```
# R Program for Expressions
> # This program demonstrates basic arithmetic expressions in R.
>
> # Define variables
> a <- 10
> b <- 5
>
> # Calculate and print the results of different expressions
> sum <- a + b
> difference <- a - b
> product <- a * b
> quotient <- a / b
>
> # Print the results
> print(paste("Sum:",sum))
[1] "Sum: 15"
> print(paste("Difference:", difference))
[1] "Difference: 5"
> print(paste("Product:", product))
```

```
[1] "Product: 50"
> print(paste("Quotient:", quotient))
[1] "Quotient: 2"
```

Program 1 {b}- Write a R program to create various data structures and print them

SOURCE CODE:

```
# R Program for Data Structures
# This program demonstrates the use of different data structures in R.

# 1. Vectors
numeric_vector <- c(1, 2, 3, 4, 5)
character_vector <- c("apple", "orange", "banana")

# 2. Matrices
my_matrix <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow = 3, ncol = 3)


# 3. Lists
my_list <- list(name = "Ram", age = 30, hobbies = c("reading", "cricket"))

# 4. Data Frame
my_data_frame <- data.frame(Name = c("Ram", "Rani", "Raja"),
                            Rollno = c(20, 21, 22),
                            Marks = c(95, 88, 90))

# Print the elements of different data structures
print("Numeric Vector:")
print(numeric_vector)

print("Character Vector:")
print(character_vector)

print("Matrix:")
print(my_matrix)

print("List:")
print(my_list)

print("Data Frame:")
print(my_data_frame)
```

SAMPLE OUTPUT

```
> print("Numeric Vector:")
[1] "Numeric Vector:"
> print(numeric_vector)
[1] 1 2 3 4 5   >
> print("Character Vector:")
[1] "Character Vector:"
> print(character_vector)
[1] "apple"  "orange" "banana"
>
> print("Matrix:")
[1] "Matrix:"
> print(my_matrix)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> print("List:")
[1] "List:"
> print(my_list)
$name
[1] "Ram"

$age
[1] 30

$hobbies
[1] "reading" "cricket"

>
> print("Data Frame:")
[1] "Data Frame:"
> print(my_data_frame)
  Name Rollno Marks
1  Ram     20    95
2 Rani     21    88
3 Raja     22    90
```

Program 2 Manipulation of vectors and matrix

Program 2 {a}- Write a R program to perform manipulation in vectors

SOURCE CODE:

```
# Manipulation of Vectors
# Create  two numeric vectors
v1 <- c(1, 2, 3, 4, 5)
v2 <- c(1, 1, 1, 1, 1)

# Print the original vector
print("Original Numeric Vectors:")
print(v1)
print(v2)

print("Manipulation of Vectors")
print("Adding 1 to each element of the 1st vector")
m <- v1 + 1
print(m)

print("Adding 1st and 2 nd vectors")
print(v1+v2)

print("Subtract 2nd vector from 1st vector")
print(v1-v2)

print(" Multiplying two vectors")
print(v1*v2)
```

OUTPUT

```
> # Manipulation of Vectors
> # Create  two numeric vectors
> v1 <- c(1, 2, 3, 4, 5)
> v2 <- c(1, 1, 1, 1, 1)
>
> # Print the original vector
> print("Original Numeric Vectors:")
[1] "Original Numeric Vectors:"
> print(v1)
[1] 1 2 3 4 5
```

```
> print(v2)
[1] 1 1 1 1 1
>
> print("Manipulation of Vectors")
[1] "Manipulation of Vectors"
> print("Adding 1 to each element of the 1st vector")
[1] "Adding 1 to each element of the 1st vector"
> m <- v1 + 1
> print(m)
[1] 2 3 4 5 6
>
> print("Adding 1st and 2 nd vectors")
[1] "Adding 1st and 2 nd vectors"
> print(v1+v2)
[1] 2 3 4 5 6
>
> print("Subtract 2nd vector from 1st vector")
[1] "Subtract 2nd vector from 1st vector"
> print(v1-v2)
[1] 0 1 2 3 4
>
> print(" Multiplying two vectors")
[1] " Multiplying two vectors"
> print(v1*v2)
[1] 1 2 3 4 5
```

Program 2{b}- Write a R Program to Perform Manipulation in Matrix

Source Code:

```
# Manipulation of Matrix

# Create two matrices
m1 <- matrix(1:9, nrow = 3, ncol = 3)
m2 <- matrix(1:9, nrow = 3, ncol = 3)

# Print the original matrix
print("Original Matrices:")
print("Matrix 1:")
print(m1)
print("Matrix 2:")
print(m2)


print("Addition of two matrices")
print(m1+m2)

print("Subtraction of two matrices")
print(m1-m2)

print("Multiplication of two matrices")
print(m1*m2)

print("Division of two matrices")
print(m1/m2)
```

Output

```
> # Manipulation of Matrix
>
> # Create two matrices
> m1 <- matrix(1:9, nrow = 3, ncol = 3)
> m2 <- matrix(1:9, nrow = 3, ncol = 3)
>
> # Print the original matrix
> print("Original Matrices:")
[1] "Original Matrices:"
> print("Matrix 1:")
[1] "Matrix 1:"
> print(m1)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> print("Matrix 2:")
[1] "Matrix 2:"
> print(m2)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
>
> print("Addition of two matrices")
[1] "Addition of two matrices"
> print(m1+m2)
     [,1] [,2] [,3]
[1,]    2    8   14
[2,]    4   10   16
```

```
[3,]    6   12   18
>
> print("Subtraction of two matrices")
[1] "Subtraction of two matrices"
> print(m1-m2)
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
>
> print("Multiplication of two matrices")
[1] "Multiplication of two matrices"
> print(m1*m2)
     [,1] [,2] [,3]
[1,]    1   16   49
[2,]    4   25   64
[3,]    9   36   81
>
> print("Division of two matrices")
[1] "Division of two matrices"
> print(m1/m2)
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    1    1
[3,]   1     1    1
```

Program 3.   Operators on Factors

```
# Creating a factor

ug <- factor(c("Bsc","BCom","BCA","Bsc","BCA","Bsc"))

# Print the factors

print("Factor:")

print(ug)

# Modifying a level in factor

levels(ug)[2] <- "BBA"

# Print the factors

print("Factor after modifying level -BCOM to BBA:")

print(ug)

# Changing order of levels in factor

ug <- factor(ug, levels = c("BBA","BCA","Bsc"))

# Print the modified factors

print("Modified levels in Factor :")

print(ug)

# Print elements from index 1 to 3 in factor

print("Elements from Index 1 to 3 in Factor :")

print(ug[1:3])  # Accessing elements from index 1 to 3 in factor

# Creating another factor

pg <- factor(c("MCom","MCA","MSC"))

# Concatenation of two factors
```

```
college <- c(ug, pg)

print("Concatenated Factor:")

print(college)

# Unique values in a factor

unique_values <- unique(ug)

print("Unique Values in Factor:")

print(unique_values)

# Length of factors

l <- length(ug)

print("Length of Factor :")

print(l)

#Check whether it is a factor

print("Is concatenated factor a factor?")

print(is.factor(college))
```

OUTPUT

```
          > # Creating a factor
> ug <- factor(c("Bsc","BCom","BCA","Bsc","BCA","Bsc"))
 > # Print the factors
> print("Factor:")
[1] "Factor:"
> print(ug)
[1] Bsc  BCom BCA  Bsc  BCA  Bsc
Levels: BCA BCom Bsc
> # Modifying a level in factor
> levels(ug)[2] <- "BBA"
> # Print the factors
> print("Factor after modifying level -BCOM to BBA:")
[1] "Factor after modifying level -BCOM to BBA:"
> print(ug)
```

```
[1] Bsc BBA BCA Bsc BCA Bsc
Levels: BCA BBA Bsc
>
> # Changing order of levels in factor
> ug <- factor(ug, levels = c("BBA","BCA","Bsc"))
>
> # Print the modified factors
> print("Modified levels in Factor :")
[1] "Modified levels in Factor :"
> print(ug)
[1] Bsc BBA BCA Bsc BCA Bsc
Levels: BBA BCA Bsc
>
>
> # Print elements from index 1 to 3 in factor
> print("Elements from Index 1 to 3 in Factor :")
[1] "Elements from Index 1 to 3 in Factor :"
> print(ug[1:3])  # Accessing elements from index 1 to 3 in factor
[1] Bsc BBA BCA
Levels: BBA BCA Bsc
>
> # Creating another factor
> pg <- factor(c("MCom","MCA","MSC"))
>
> # Concatenation of two factors
> college <- c(ug, pg)
> print("Concatenated Factor:")
[1] "Concatenated Factor:"
> print(college)
[1] Bsc  BBA  BCA  Bsc  BCA  Bsc  MCom MCA  MSC
Levels: BBA BCA Bsc MCA MCom MSC
>
> # Unique values in a factor
> unique_values <- unique(ug)
> print("Unique Values in Factor:")
[1] "Unique Values in Factor:"
> print(unique_values)
[1] Bsc BBA BCA
Levels: BBA BCA Bsc
>
> # Length of factors
> length_factor <- length(ug)
> print("Length of Factor :")
[1] "Length of Factor :"
```

```
> print(length_factor)
[1] 6
>
> #Check whether it is a factor
> print("Is concatenated factor a factor?")
[1] "Is concatenated factor a factor?"
> print(is.factor(cfactor))
[1] TRUE
>
```

Program 4   Data Frames in R

Program 4- Write a R program to perform various operations in data frame
SOURCE CODE:

```
# R program for dataframe
# A vector which is a character vector
name = c("Ram", "Raja", "Rani")
# A vector which is a character vector
subject = c("R", "Python", "Java")
# A vector which is a numeric vector
mark = c(64, 72, 80)
df = data.frame(name, subject, mark)
print(df)
# Accessing first and second row
print("Accessing first and second row\n")
print(df[1:2, ])
# Accessing first and second column
print("Accessing first and second column\n")
print(df[, 1:2])
print("Before adding row\n")
print(df)
# Add a new row using rbind()
newDf = rbind(df, data.frame(name = "deepa",
                  subject = "C",
                   mark = 75
))
print("After Added a row\n")
print(newDf)
print("Before adding column\n")
print(df)
# Add a new column using cbind()
newDf = cbind(df, rank=c(3, 5, 1))
print("After Added a column\n")
print(newDf)
print("Before deleting the 3rd row and 2nd column\n")
print(df)
# delete the third row and the second column
newDF = df[-3, -2]
print("After Deleted the 3rd row and 2nd column\n")
print(newDF)
```

OUTPUT:

```
  name subject mark
```

```
1  Ram       R   64
2 Raja  Python   72
3 Rani    Java   80

[1] "Accessing first and second row\n"
  name subject mark
1  Ram       R   64
2 Raja  Python   72

[1] "Accessing first and second column\n"
  name subject
1  Ram       R
2 Raja  Python
3 Rani    Java

[1] "Before adding row\n"
  name subject mark
1  Ram       R   64
2 Raja  Python   72
3 Rani    Java   80

[1] "After Added a row\n"
   name subject mark
1   Ram       R   64
2  Raja  Python   72
3  Rani    Java   80
4 deepa       C   75

[1] "Before adding column\n"
  name subject mark
1  Ram       R   64
2 Raja  Python   72
3 Rani    Java   80


[1] "After Added a column\n"
  name subject mark rank
1  Ram       R   64    3
2 Raja  Python   72    5
3 Rani    Java   80    1

[1] "Before deleting the 3rd row and 2nd column\n"
  name subject mark
1  Ram       R   64
```

```
2 Raja  Python   72
3 Rani    Java   80

[1] "After Deleted the 3rd row and 2nd column\n"
  name mark
1  Ram   64
2 Raja   72
```

Program 5 List and Operators

Program 5- Write a R program to perform various operations on lists
SOURCE CODE:

```
        # R program for lists

        # Create a list

rollno <- c(10,11,12)

name <- c("ram","rocky","raj")
```

```
marks <- c(98,99,80)

rank <- c('B','A','C')

student <- list(rollno, name, marks, rank)

        # Print the original list

print("Original List:")

print(student)

        # naming the original list

print("Original List with names:")

names(student) <- c("Roll number", "Name", "Marks", "Rank")

print(student)

        # Accessing elements in the list

print("Accessing third element:")

print(student[3])

        # Accessing fourth element with name in the list

print("Accessing fourth element with name:")

print(student$Rank)
        # Adding an element to the list

print("Adding an element:")

student[5] <- "Bsc IT"

print(student)
        # updating an element to the list

print("updating last element:")

student[5] <- "Bsc CS"
```

```
print(student)
        # Calculating the length of the list

print("Length of the list:")

student_length <- length(student)

print(student_length)

        # Removing an element from the list

print("Removing an element:")

student <- student[-5]  # Remove the fifth element

print(student)
```

Output:

```
        > # Create a list
> rollno <- c(10,11,12)
> name <- c("ram","rocky","raj")
> marks <- c(98,99,80)
> rank <- c('B','A','C')
> student <- list(rollno, name, marks, rank)
>
        > # Print the original list
> print("Original List:")
[1] "Original List:"
> print(student)
[[1]]
[1] 10 11 12
[[2]]
[1] "ram"   "rocky" "raj"
[[3]]
[1] 98 99 80
[[4]]
[1] "B" "A" "C"
>
        > # naming the original list
> print("Original List with names:")
[1] "Original List with names:"
> names(student) <- c("Roll number", "Name", "Marks", "Rank")
> print(student)
$`Roll number`
[1] 10 11 12
$Name
[1] "ram"   "rocky" "raj"
$Marks
[1] 98 99 80
$Rank
[1] "B" "A" "C"
>
>
          > # Accessing elements in the list
> print("Accessing third element:")
[1] "Accessing third element:"
> print(student[3])
$Marks
[1] 98 99 80
>
          > # Accessing fourth element with name in the list
> print("Accessing fourth element with name:")
[1] "Accessing fourth element with name:"
> print(student$Rank)
[1] "B" "A" "C"
>
>
          > # Adding an element to the list
> print("Adding an element:")
[1] "Adding an element:"
> student[5] <- "Bsc IT"
> print(student)
$`Roll number`
[1] 10 11 12
$Name
[1] "ram"   "rocky" "raj"
$Marks
[1] 98 99 80
$Rank
[1] "B" "A" "C"
[[5]]
[1] "Bsc IT"
>
          > # updating an element to the list
> print("updating last element:")
[1] "updating last element:"
```

```
> student[5] <- "Bsc CS"
> print(student)
$`Roll number`
[1] 10 11 12
$Name
[1] "ram"   "rocky" "raj"
$Marks
[1] 98 99 80
$Rank
[1] "B" "A" "C"
[[5]]
[1] "Bsc CS"
        > # Calculating the length of the list
> print("Length of the list:")
[1] "Length of the list:"
> student_length <- length(student)
> print(student_length)
[1] 5
        > # Removing an element from the list
> print("Removing an element:")
[1] "Removing an element:"
> student <- student[-5]  # Remove the fifth element
> print(student)
$`Roll number`
[1] 10 11 12
$Name
[1] "ram"   "rocky" "raj"
$Marks
[1] 98 99 80
$Rank
[1] "B" "A" "C"
>
```

Program 6   Loops in R

Program 6- Write a R program to demonstrate various types of loops

SOURCE CODE:

```
        > # R program to demonstrate loops
        > # application of for loop
> week <- c('Sunday',
       'Monday',
       'Tuesday',
       'Wednesday',
       'Thursday',
       'Friday',
           'Saturday')
>
>
> for (day in week)
 {
   print(day)
 }
[1] "Sunday"
[1] "Monday"
[1] "Tuesday"
[1] "Wednesday"
[1] "Thursday"
[1] "Friday"
[1] "Saturday"
>
            > # R program to demonstrate the use of while loop
>
> x = 1
> while (x <= 5)
 {

   cat("2 x ", x,"=", x*2,"\n")
   x = x + 1
 }
2 x  1 = 2
2 x  2 = 4
2 x  3 = 6
2 x  4 = 8
2 x  5 = 10
>
> # R program to illustrate
> # the application of repeat loop
>
> i <- 0
> repeat
 {
   print("Basic to pro tamil!")
   i = i + 1
   if (i == 5)
   {
     break
   }
 }
```

Output:

```
            > # R program to demonstrate loops
            > # application of for loop
> week <- c('Sunday',
           'Monday',
           'Tuesday',
           'Wednesday',
           'Thursday',
           'Friday',
           'Saturday')
>
>
> for (day in week)
 {
   print(day)
 }
[1] "Sunday"
[1] "Monday"
[1] "Tuesday"
[1] "Wednesday"
[1] "Thursday"
[1] "Friday"
[1] "Saturday"
>
            > # R program to demonstrate the use of while loop
>
> x = 1
> while (x <= 5)
 {

   cat("2 x ", x,"=", x*2,"\n")
   x = x + 1
 }
2 x  1 = 2
2 x  2 = 4
2 x  3 = 6
2 x  4 = 8
2 x  5 = 10
>
> # R program to illustrate
> # the application of repeat loop
>
> i <- 0
> repeat
```

```
{
  print("Basic to pro tamil!")
  i = i + 1
  if (i == 5)
  {
    break
  }
}
```

```
[1] "Basic to pro tamil!"
[1] "Basic to pro tamil!"
[1] "Basic to pro tamil!"
[1] "Basic to pro tamil!"
[1] "Basic to pro tamil!"
```

Program 7   Graphs in R

Write a R program to demonstrate various types of graphs

SOURCE CODE:

```
# Create sample data
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)
par(mfrow = c(2, 2))
# Plot a basic line graph
pts <- c(2,3,4,6,8)
 plot(pts, type = "o", col = "blue", lwd = 2, main = "Simple Line Graph", xlab = "X-axis", ylab = "Y-axis")
# Add points to the graph
```

```
points(pts, col = "red", pch = 16)
barplot(c(3, 5, 2, 7, 4), names.arg = c("A", "B", "C", "D", "E"), col = "orange", main = "Bar Plot",
xlab = "Categories", ylab = "Values")
plot(x, y, col = "green", pch = 18, main = "Scatter Plot", xlab = "X-axis", ylab = "Y-axis")


boxplot(x, col = "blue", names = "X", main = "Box Plot", xlab = "Variables", ylab = "Values")
```