**R Programming \_Basic\_Task 1**

**1.Vector Recycling:**

We can add two vectors together using the + operator.

One thing to keep in mind while adding (or other arithmetic operations) two vectors together is the recycling rule.

If the two vectors are of equal length then there is no issue. But if the lengths are different, the shorter one is recycled (repeated) until its length is equal to that of the longer one.

This recycling process will give a warning if the longer vector is not an integral multiple of the shorter one.

a <- c (1, 2, 3)

b <- c (5, 6, 7, 8, 9, 10)

a + b

#output: c (6, 8, 10, 9, 10, 11, 13)

a – b

#output: c (-4, -4, -4, -7, -7, -7)

**2.Inner multiplication:**

If matrix is multiplied with vector then vector will be promoted to either row or column matrix to make two arguments conformable.

#use %\*%

m <- matrix(1:8, nrow=2)

n <- matrix(8:15, nrow=4)

m %\*% n

output:

[162 226]

[200 280]

**3.Outer multiplication:**

Rules:

1. the operate that being used is %o%

2. the number of column in first matrix must equal number of row in second matrix

#use %o%

u <- 1:4

u %o% u

output:

[1 2 3 4]

[2 4 6 8]

[3 6 9 12]

[4 8 12 16]

**4.Function:**

**1. sample() :** it takes a random sample of elements

Sample(x = 1:5)

#output: [1 2 3 4 5]

Sample(x = 1:5, size = 3)

#output: [1 2 3]

**2. seq()**: it generate a regular sequence of output base on from, to arguments(number of output specify by length.out)

X <- seq(-1,1)

#output: [-1, 0, 1]

**3. rep() :** it generates a repeatedly sequence of output as mentioned

Req(10,5)

#output: [10,10,10]

**4. round()** round a provided double/ float in to n-digits after decimal base on mentioned

round(0.1009, 1)

` #output: 0.1

**5. factorial()** it return factorial value of the input

factorial(2)

#output: 2

**6. Is()** it returns true/ false if an object belong to a class

is(False, “logical”)

#False

**7. mean()** :it return average value of an input

mean(1:5)

#output: 3

**8. set.seed()** set the seed for the generate random number

set.seed(3)

rnorm(3)

#output: [2 1 0]

**5.Subset:** Subsetting is a useful indexing feature for accessing object elements.

a< -c(1,2,3,4,5,6,7,8)

subset(a, a<=5)

#output: [1,2,3,4,5]

**6.Write a program to calculate the BMI rate.Get the user input & Result should be in integer.**

weight <- as.numeric(readline(prompt = "enter your weight: "))

height <- as.numeric(readline(prompt = "enter your height: "))

bmi <- round(weight/(height\*height))

print(bmi)

**7. Create a function to calculate the BMI Rating? Result should be in integer**

calcBMI <- function(){

weight <- as.numeric(readline(prompt = "enter your weight: "))

height <- as.numeric(readline(prompt = "enter your height: "))

bmi <- round(weight/(height\*height))

print(bmi)

}

calcBMI()