Assignment 5 (Manjeet Singh)

Q -5.

Q1) Write a R program using control operators to test whether following values are prime numbers or not by providing a PRIME or NOT PRIME message as output

Ans :-
```
x = c(103, 82, 179)
for(z in x){
 if((z %% 2 != 0) && (z %% 3 != 0)){
 cat(z,": PRIME NUMBER \n")
}else{
 cat(z, ": Not PRIME NUMBER \n")
}
}
```

Q2) Write a R program using control operators to identify letter u and a both occur in the following words:

Ans :
```
library(stringr)
library(dplyr)
words = c("above", "unit", "Under")
for(alphabet in words){
 if((str_detect(alphabet, "a"))==TRUE && ((str_detect(alphabet, "u"))==TRUE)){
cat(alphabet, ": both u and a appear in each words \n")
  }else{
 cat(alphabet, ": both u and a doesn't' appear in each words \n")
}
}
```

Q3) Write a function that to calculate BMI (Body Mass Index):

Ans:
```
BMI <- function(weight, height){
BMI = (weight/(height^2))*703
return(BMI)
}
```

Q4: Write a function called sum_of_cubes, that calculates the sum of cubes of the first n natural numbers :

Ans :
```
sum_of_cubes <- function(x){
cubes <- x^3
sum <- sum(cubes)
return(sum)
}
sum_of_cubes(6)
```

Q5: Write a function to calculate the mode (highest frequency) of the following vector:

Ans :
```
get <- function(x){
Mode <- unique(x)
Mode[which.max(tabulate(match(x, Mode)))]
```

```
}
x = c(2,3,3,4,4,5,6,7,9,10)
get(x)
```

Q6: Write a function to calculate the no. of prime numbers of the following vector :

Ans :
```
install.packages("numbers")
library(numbers)
x <- c(2,2,3,3,4,5,7,11,15,19,24,29)
calculate_no_primes <- function(x){
for(i in x){
 if(isPrime(i)){
cat(i, ": Prime \n")
}else{
cat(i, ": Not Prime \n")
}
}
}
calculate_no_primes(x)
```

Q 7. Count the number of prime numbers

Ans :
```
prime count  <- function(n){
x= 0
for (a in n){
 if(is.prime(a)){
x =x+1
}
return(x)
}
```

Q8. Perform below operations using Data.frame and Data.table

Ans :

(a)
```
boys <- read.csv("boystop100s.csv", header = TRUE)
head(boys)
girls <- read.csv("girlstop100s.csv", header = TRUE)
head(girls)
stu <- data.frame(roll_no = c(3,1,2,5,4), names = c('peter', 'jack', 'david', 'james', 'john'))
stu
library(data.table)
marks <- data.table(roll_no = c(4,2,3,6,1), maths = c(89,92,76,67,90), science =
c(98,92,88,91,92))
marks
```

(b)
```
Join <- merge(stu, marks)
Join
```

(c)
```
Join1 <- merge(stu, marks, by="roll_no", all.x=TRUE)
Join1
```

(d)
```
Join2 <- merge(stu, marks, by="roll_no", all.y=TRUE)
Join2
```

(e)
```
Join3 <- merge(stu, marks, by='roll_no', all=TRUE)
```

```
        Join3
(f)     fill <- Loblolly
        fill
        fill$height
        library(dplyr)
        filter(fill, !is.null(fill$height))[,2]
        library(dplyr)
(g)     group_by(Loblolly, age) %>% summarise(sum = sum(age),average = mean(age))
```

Q9: Create R functions for the following operations.

Ans :
```
gas <- function(){
library(dplyr)
data("infert")
color <- infert
color
distinct(color, age)
}
gas()
```

Q10: Create R functions for the following operations

(a): Find out if there are any nulls in a dataset or in some specific number of columns

```
 mydata <- function(){
 mydata <- BOD
mydata
if(is.null(mydata$demand) || (!complete.cases(mydata))){
print(TRUE)
}else{
print(FALSE)
}
}
mydata()
```

(b) HDFS is out of context.

Q11: Create R functions for the following operations

Ans:

(a)
```
poise <- function(x){
viz <- x[duplicated(x)]
print(viz)
return(x)
}
x <- c(8,9,9,7,5,4,4,3,2,6,6,2,1)
```

(b)
```
poise(x)
Uni <- function(x){
sun <- unique(x)
return(sun)
}
```

```
x <- c(8,9,9,7,5,4,4,3,2,6,6,2,1)
Uni(x)
```

(c)
```
clip <- function(){
 j <- "Planet"
w <- "World"
paste(j,w, sep="::")
}
clip()
```

(d)
```
Melt <- function(){
mat <- matrix(c(1:10), nrow=5,ncol=2)
cat("Sum column wise :", apply(mat, 2, sum), "\n")
cat("Sum row wise :", apply(mat, 1, sum))
}
Melt()
```

(e) and (f) and out of context - HDFS is out of context

Q12: Create R functions for the following operations

Ans :

(a)
```
r.file <- hdfs.file(hdfsFilePath,"r")
from.dfs(
mapreduce(
input = as.matrix(hdfs.read.text.file(r.file)),
input.format = "csv",
map = ...
))
```

(b)
```
seat <- function(){
seat <- Seatbelt
seat
colnames(seat) <- c("driverkilled", "Drivers", "Front",
        "Rear", "KM/S", "PetrolPrice",
        "vankill", "LAW")
print(seat)
seat <- seat[,(names(seat)) %in% c("law", "vankill")]
print(seat)
```

(c)
```
d <- data.frame(alpha =1:3 , beta = 4:6, gamma = 7:9)
library(plyr)
names(d)
rename(d, c("beta" = "bb" , "gamma" ="c")
```

(e)
```
 x <- c(NA, NaN)
print(is.na(x))
print(is.nan(x))
print(class(NA))
print(class(NaN))
```

```
print(class(NULL))
```

(f)
```
vec <- c(1,2,3,4,5)
if(is.numeric(vec)){
print(TRUE)
}
```

(g)
```
library(dplyr)
Orange
distinct(Orange, Tree)
}
seat()
```